Project Report

On

QUANTUM-INSPIRED MACHINE LEARNING FOR HIGH DIMENSIONAL MARKETING ANALYSIS

Submitted

In partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE

in

MATHEMATICS

by

Liya Maria Thomas

(Register No. AB22AMAT015)

Under the Supervision of:
SANGEETHA CHANDRAN

DEPARTMENT OF MATHEMATICS

ST. TERESA'S COLLEGE (AUTONOMOUS)

ERNAKULAM, KOCHI – 682011

APRIL 2025

ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

CERTIFICATE

This is to certify that the dissertation entitled, **QUANTUM INSPIRED MACHINE LEARNING FOR HIGH DIMENSIONAL MARKETING ANALYSIS** is a bonafide record of the work done by Ms. **JOSNA JOSEPH** under my guidance as partial fulfillment of the award of the degree of **Bachelor of Science in Mathematics** at St. Teresa's College (Autonomous), Ernakulam affiliated to Mahatma Gandhi University, Kottayam. No part of this work has been submitted for any other degree elsewhere.

Date: 17-02-2025
Place: Ernakulam

Sangeetha Chandran
Research Scholar,
Cochin University of Science and Technology,
Kalamassery.

Dr. Elizabeth Reshma MT
Assistant Professor and Head
Department of Mathematics
St.Teresa's College(Autonomous),
Ernakulam.

ST TERESA'S COLLEGE
DEPT. OF MATHEMATICS
ERNAKULAM

External Examiner:
1.....
2.....

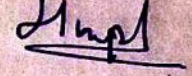
ACKNOWLEDGEMENT

I must mention several individuals who encouraged me to carry this work. Their continuous invaluable knowledgeable guidance throughout the course of this study helped me to complete the work up to this stage.

I am very grateful to my project guide Smt.**Sangeetha Chandran** for her guidance and support throughout the work. I also want to express my gratitude to my classmates, who assisted me in formulating my research topics by sharing their knowledge. Last but not least, I would like to express my heartfelt gratitude to God Almighty for always being my emotional rock and teaching me how to confront every challenge with intelligence and confidence.

Place: Ernakulam **JOSNA JOSEPH**

Date: 17-02-2025 **(AB22AMAT014)**



DECLARATION

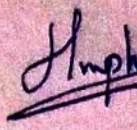
I hereby declare that the work presented in this project is based on the original work done by me under the guidance of **Sangeetha Chandran** Research Scholar, Department of Computer Science ,Cochin University of Science and Technology , Kalamassery and has not been included in any other project submitted previously for the award of any duties.

Place: Ernakulam.

Date: 17-02-2025

JOSNA JOSEPH

(REGISTER NO:AB22AMAT014)



Contents

CHAPTER 1

INTRODUCTION

The rapid evolution of technology has profoundly impacted various sectors, and the financial markets are no more exception. As the complexity and dimensionality of financial data increase, traditional machine learning approaches often struggle to efficiently process and analyse this information . Quantum machine learning (QML) emerges as a transformative solution, harnessing the principles of quantum mechanics to enhance computational capabilities. Quantum machine learning integrates quantum computing and machine learning, enabling the processing of the high-dimensional data with unprecedented speed and efficiency. This chapter explores the fundamental concepts of QML and its potential applications in financial markets, highlighting its advantages over classical techniques.

QML is founded on the principles of quantum superposition and entanglement, thereby allowing for the functionalities that are impossible to achieve with computers operating based on classical principles. The states that the quantum system recognizes simultaneously make quantum algorithms theoretically faster data processors than their counterpart in the classical world. More importantly, entanglement allows the possibility of achieving higher correlations of data points in this gigantic database, which implies that it is possible to get more accuracy with the models applied to financial forecasting and risk management. All these properties of quantum have huge potential for change in the manner with which the financial institutions analyse the trend of the market and subsequently optimize the portfolios and forecast the movement of the asset, hence, a huge amount of competitive advantage in such data-oriented landscape today.

Quantum parallelism enables a quantum algorithm, like QSVM and QNN, to be capable of taking decisions highly speedily and accurately. This is highly critical in financial markets because, from fraud detection to high-frequency trading, from the detection of hidden patterns to efficient analysis of time-series data and high-dimensional data processing, practical applications seem near because of such rapid technological improvement despite the fact that the present quantum hardware is suffering from error rate and qubit coherence. Instead, maturity will enable QML to integrate all the existing financial systems into one fabric of financial analysis, investment strategies, and risk management.

1.1 PROBLEM DEFINITION

1.1.1 OBJECTIVE OF THE STOCK MARKET PREDICTION MODEL

- **Stock forecasting:**

Forecasting a stock price is one of those activities to be inspired upon with the aim of finetuning investment methods and maximization returns on finance stock forecasting includes not only financial information and historical data regarding stock price but many other additional extraneous factors. This has led to many strategies and tools that have emerged and range from sophisticated techniques related to machine learning and quantum computing to more traditional statistics techniques.

- **Stock Trend Prediction:**

Investors who wish to make good decisions and maximize their trading strategies should predict the trends in stocks. The understanding of stock trends helps investors recognize risks and opportunities. The stock trends indicate the overall direction in which the price of a stock is likely to move in a certain time period. Better decisions will be made with increases in computational power and analytics in data.

- **Anomaly Detection:**

The most significant aspect of financial analysis for the detection of anomalies in stock market data is that it may reveal the presence of some weird patterns or outliers that may depict a potential major event or fraud activities or even the shift in behavior in markets. The anomalies have a significant influence on trading strategies, risk management, and the overall stability of markets. Therefore, anomaly detection within stock market data is important so as to retain integrity in the marketplace proper management of risks and enhancement in an investment strategy.

1.1.2 TYPES OF PREDICTION

- **Closing Prices Daily:**

The closing prices of a day trading are probably among the simplest metrics in a share of the stock market analyses but which show the closing or ending prices at which stocks may trade, and that, accordingly are critical data for either an investor or analyst.
The closing price of each day is what the study revolves around to provide very valuable information in terms of understanding the movement of prices and changes in market sentiment. Such information can help in taking proper decisions regarding investment risk and strategy while trading.

- **Volatility:**

The measure by which a stock changes during some given time is called volatility. One of the defining words that represent finance is that it is a measure for some form of security regarding risk that moves around prices. Being that it determines the results of investing decisions, it measures the evaluation of the risk of a given investment, and also determines all

the different approaches in which trading must be undertaken then knowing about volatility is paramount to an investor.

•Sector Trend:

Sector trends are essential for understanding the broader market landscape and making informed investment decisions. By analysing different sectors and their performance, investors can capitalize on opportunities, manage risks, and enhance portfolio diversification. Recognizing the cyclical nature of sectors and employing strategies like sector rotation can further optimize investment outcomes. Sector trends refer to the performance and behaviour of specific segments within the broader stock market, categorized by industry. Understanding sector trends is crucial for investors and analysts, as they can provide insights into economic conditions, investor sentiment, and potential investment opportunities.

1.2 MOTIVATION AND SIGNIFICANCE

Importance of accurate stock market predictions in financial markets

Accurate stock market predictions play a vital role in the functioning of financial markets. They influence investment strategies, risk management, and overall market stability. Here are some key reasons why these predictions are important:

1. Informed Investment Decisions

• Optimal Timing: Accurate predictions help investors decide when to buy or sell stocks, maximizing potential returns. Timing can significantly impact profitability, especially in volatile markets.

• Asset Allocation: Investors can make more informed decisions about how to allocate their portfolios across different asset classes based on expected market movements.

2. **Identifying Risks:**

• Predictions can highlight potential downturns or volatility, allowing investors to implement risk management strategies, such as diversifying portfolios or hedging against losses.

3. **Minimizing Losses:**

• By anticipating negative market trends, investors can take pre-emptive actions to minimize potential losses, protecting their capital.

4. Market Efficiency

- The good prediction of the prices leads to effective security pricing. When the investor has some good prediction about future price discovery, the stocks will be correctly priced considering available information.

- Reduction of Information Asymmetry Greater predictive power leads to decreased differences between.

5. Strategic Planning for Institutions

- Portfolio Management: Institutional investors rely on accurate predictions to manage large portfolios effectively, adjusting holdings based on anticipated market conditions.

- Fund Performance: Accurate predictions can lead to better fund performance, which is crucial for asset managers aiming to attract and retain clients.

6. Economic Indicators

- Market Sentiment: Predictions can reflect overall market sentiment and economic health, influencing consumer confidence and spending.

- Policy Implications: Accurate market forecasts can inform policymakers about economic trends, enabling them to make decisions that promote stability and growth.

7. Investment Strategies

- Algorithmic Trading: Many trading strategies rely on accurate predictions. Algorithms that execute trades based on market forecasts can capitalize on price movements more effectively.

- Technical and Fundamental Analysis: Predictive accuracy enhances the effectiveness of both technical and fundamental analyses, guiding traders in their strategies.

1.3 DISTANCE MEASURE IN CLASSICAL AND QUANTUM SYSTEM

Potential impact of using quantum inspired and classical distance measures on prediction accuracy.

Quantum-Inspired Distance Measures:

a. Better pattern recognition

b. Faster computation

c. They excel at modelling intricate relationships in the data, which may improve prediction accuracy.

Classical Distance Measures:

a. Simple and Straightforward

b. Proven Reliability

c. Lower resource use.

In other words, quantum-inspired measures could provide more advanced abilities in complex data, but the classical measures give much reliable and easy-to-use options. Again, the choice depends on the complexity of the market data and on available resources.

1.4 OBJECTIVE

Developing and evaluating a stock market prediction model using a hybrid of quantuminspired and classical distance measures involves several steps. Here's a structured approach:

1. UNDERSTANDING THE COMPOUNDS

a. **Quantum-inspired methods:** Quantum-inspired algorithms use principles from quantum computing to solve problems.

b. **Classical Distance Measures:**Classical distance measures such as Euclidean Distance, Manhattan Distance, are useful.

2. DATA COLLECTION AND PREPROCESSING

a. **Data Collection:** Gather historical stock market data, which typically includes stock prices, trading volumes, and other financial indicators.

b. **Pre-processing**: Normalize or standardize the data to ensure all features or on a comparable scale.

3. MODEL DESIGN

a. Feature Selection: Identify relevant features for predicting stock prices, such as moving averages, trading volume

b. Classical Distance Measure: Component Use distance measures for clustering.

c. Hybrid Approach: Integrate the quantum-inspired and classical components.

4. QUANTUM – INSPIRED DISTANCE MEASURES

a. **Quantum inspired measures:** These measures often leverage ideas from the quantum mechanics

b. **Classical Distance Measures:** Implement using standard libraries.

c. **Hybrid Integration:** Combine the outputs of quantum-inspired and classical methods.

5. EVALUATION

a. Performance Metrics: Evaluate the model using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE).

b. Back testing: Implement back testing to evaluate how well the model performs on historical data.

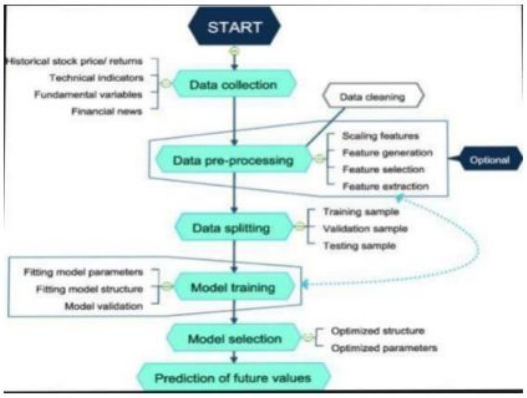
c. Robustness and Stability: Test the model under different market conditions to ensure it is robust and stable.

d. Sensitivity Analysis: Perform sensitivity analysis to understand how changes in input features

6. RESULT AND ITERATION

a. Analyse Results: Examine the results and performance metrics.

b. Iterate and Improve: Based on the evaluation, iterate on the model by adjusting hyperparameters, incorporating additional feature



Data processing flowchart

Fig 1.1

1.5 RESEARCH QUESTIONS

What can quantum computing be used for in marketing analysis?

R1: It can process very complex data, which results in much more accurate and minute client segmentation.

R2: Algorithms can reach people with much more accuracy due to the quantum processing of enormous data at extremely fast speeds.

R3: Predictive Analytics: Quantum models now predict consumer trends and market performance much more accurately than before.

R4: Personalized marketing: Perhaps fantastic marketing will come through with actual, realtime data processing from quantum computers, so the marketing becomes perfectly matched and highly effective.

R5: Sentiment Analysis: This will transform the face of sentiment analysis with real-time volumes of processing social media.

CHAPTER 2

LITERATURE REVIEW

2.1 STOCK MARKET PREDICTION TECHNIQUES

Review of Traditional and Modern Approaches to Stock Market Prediction
Stock market prediction has, of course, improved much since the earlier days. That has led to emergence of various methodologies that are supposed to analyse market data with an aim of better price movement forecasting. It is the review of the traditional and modern approaches: their strengths, weaknesses, and applicability.

2.1.1 Traditional Approaches

1. Fundamental Analysis: Involves analysing a company's financial statements, earnings, revenue growth, management quality, and overall economic conditions.

2. Technical Analysis It makes use of historical price data in combination with volumes of trade for the identification of charts and trends that are possible using some techniques termed as charting along with indicators. Suits short-term strategies thus provides a systemic approach toward analysing price movement.

3. Statistical Models-Use statistical techniques in regression analysis, time series, and ARIMA models with historical data trends to forecast prices.

2.1.2 Modern Approaches

1. Machine learning: That are algorithms for decision trees, neural networks, and support vector machines, for discovering pattern in large quantities of data and therefore to work with hundreds of thousands information, and learn new data and then discover unnoticed interrelations among the variables.

2. Deep Learning: That portion of machine learning which makes use of multilayer neural networks for modelling complex data patterns; usually excellent with sequential data, timeseries, and automatically extracts the appropriate features.

3. Quantum Computing: This uses quantum algorithms, which may solve complicated optimization problems much faster than classical computers. Thus, it can process extensive

datasets and complex calculations more effectively, giving new opportunities in predictive modelling.

2.2 Discuss the application of machine learning models in financial forecasting.

Machine learning models can be used for financial forecasting.

Machine learning changed financial forecasting with the development of advanced tools that could search through humongous amounts of data to find intricate patterns. Among its prominent applications in this space include:

1. **Share Price Prediction:**
   Models can be built with the help of algorithms such as Linear Regression, Random Forest, and Support Vector Regression, and other neural nets like Long Short term Memory to predict share prices for later dates with historical data of price and volume and other financial details. This can be useful to the traders and investors according to the trend and correlation that may be realized by such a model.
2. **Algorithmic Trading:**
   The algorithm ML gives birth to a trading strategy that can self-optimize itself based on how well it has performed up until now and current live market information. The optimality of the trades, and in fact the potential speed of response to changed market conditions, are enhanced because they try to seek a maximum return with an optimized risk. **3. Credit Scoring and Risk Rating:**
   Logistic Regression, Decision Trees, Neural Networks, etc. are some classification algorithms. The different parameters the ML models measure to analyse credit history, income, and pattern of transactions, include being credit worthy or not about the borrowers. In short, these models don't only allow lenders to choose correctly but also minimize the default rates.

2.3 DISTANCE MEASURE IN MACHINE LEARNING

Distance measures are fundamental concepts in machine learning, used to quantify how similar or dissimilar data points are from one another. They play a crucial role in various algorithms, particularly in clustering, classification, and anomaly detection. Here's an overview of common distance measures and their applications.

1. Euclidean Distance

The straight-line distance between two points in Euclidean space. **Formula**:
For two points
p = (p1,p2,…..pn)and q=(q1,q2,…..,qn)

2. Manhattan Distance:

The distance calculated as the sum of the absolute differences of their coordinates.
Formula:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

3. Murkowski Distance:

A generalized distance metric that includes both Euclidean and Manhattan distances as special cases. **Formula:**

$$d(p,q) = \left(\sum_{i=1}^{n}|p_i - q_i|^k\right)^{1/k}$$

Key features	Company Typology		
	Conventional	Responsible	Essential
Willingness to act	Reluctance to change	Systematic application of legal norms	Pro-activity
Behaviour levels	Lack of awareness or knowledge	Assume the need to change	Influencers are seen as models to encourage the adoption of different and particular behaviours
Regulations, customs and habits	Maintenance of their habits and customs	Implement changes on their habits and customs using normative ways	Their behaviour is often driven to change other entities
Changing costs	Understood as a major obstacle	Financial measures may be particularly effective in driving change	Seen as an investment
Conviction	Lack of confidence in their abilities	Take on challenges	Believe that their behaviour can make a difference
Terminology	Ignorance	Identify	Create terminology
Relative sustainability	Trial on a single behaviour	Inter-related activities	Operate on multidimensional interrelationships
Governance	Maximize the benefit of their investors	Maximize the benefit of their investors. A portion of the earnings reverse to offset some of the negative externalities produced at the social level	Maximize society wealth creation by providing health products and services
Markets	Comply with the rules in market practice	Seek to avoid the bad effects that their products and services may have	Profits from the competitive advantage opportunities that they create and plan

List of company and features

Fig 2.1

24 Quantum inspired techniques in financial data analysis

•**Quantum-inspired techniques:** Exploit the principles of quantum computing in order to make the process of data analysis in finance more intensive. The most popular methods mainly exploit the benefits of quantum algorithms without using real hardware quantum. The following list describes key quantum-inspired techniques used for financial data analysis.

•**Quantum-Inspired Algorithms**: These are the algorithms, which are inspired by a quantum computational process but, applied to the solution of the classical problems, efficiently solve the same.

Example: The Quantum Approximate Optimization Algorithm (QAOA) can be used for portfolio optimization, and therefore, the possibility of exploration in complex solution spaces will be open.

• **Quantum-Inspired Machine Learning**

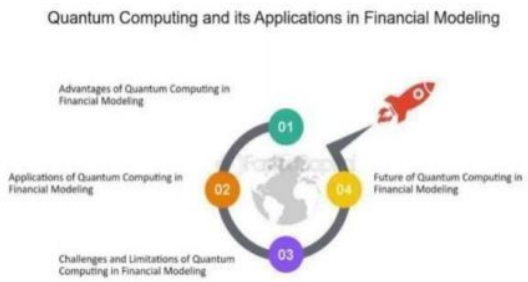
Quantum Support Vector Machines (QSVM): It is an extension of traditional SVMs in which kernel methods are used that perform well in a higher dimensional space and enhance the quality of classification.

•**Quantum-Inspired Optimization Techniques**

Description: Problems of optimization, such as portfolio optimization and asset allocation, are solved using algorithms like the Variation Quantum Eigen solvers, which not only enhance the portfolios by better exploration of the solution space but also do it far better than classical methods.

•**Quantum-Inspired Feature Selection**

Description: Quantum-inspired techniques are applied to feature selection, for the purpose of picking those important features that will improve and reduce dimensionality but allow the consideration of relevant variables, with a consequential impact on prediction accuracy as financial modelling works.



Quantum-inspired application in finance

Fig 2.2

2.5 Ensemble learning in stock market prediction

One of the best techniques for high accuracy and strong robustness in predictions is ensemble learning as its power relies on more than one model. Application approaches of ensemble techniques in a prediction task with stocks, they discuss and utilize the power of many algorithms which take into consideration the likelihood of getting an intricate pattern in financial data. A summary below discusses how this ensemble learning was applied in the domain.

1. Introduction to Ensemble Learning
It is an ensemble technique toward learning where the final output arises from multiple models that can be used for prediction making. Here, the thinking is that weak learners might be combined in a way of a strong learner.

2. Using Ensemble Learning to Predict Stock Market
It includes several algorithms, like decision trees, support vector machines, and neural networks, making it more predictable about the price.
Since this ensemble is able to combine models, it is better at identifying upward or downward trends, which is precisely what the traders need in order to decide on a thing.

3. Advantages of Ensemble Learning
Generally, the ensemble performance of forecasting is always better than any individual model. That is mainly due to the properties of the ensemble that mostly tend to produce low noises and occur at very small rates compared with the models.

4. Methods of Ensemblation Used in Most Financial Scenarios
In reality, it is an extension of bagging because it generates multiple decision trees and adds their output together. It is useful when datasets have big and high-dimensional features. Pooling all the predictions of individual classifiers on problems is done through majority vote on classification problems and averaging in the case of regression problems. It is quite a straightforward technique for assembling.

Stock prediction

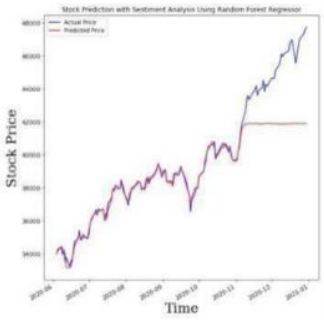


Fig 2.3

2.6 SUMMARY

This will amalgamate a variety of models that can enhance the strength of the system of prediction and further strength of the outcome. For instance, the use of ensemble learning in the stock market improves prediction highly. Since the robustness, several algorithms are bound to detect patterns in financial data successfully. As financial markets grow, so will the numbers of applications of ensemble methods so that more rich insight and better decisionmaking is found in finance. Ensemble learning combines predictions from multiple models, which implies that weak learners are brought into a powerful prediction model.

CHAPTER 3

DATA

3.1 DATA COLLECTION

To gather the data for this study, a number of sources on stock market values were consulted. One major source employed was Kaggle, a website that emphasizes a wealth of datasets in many industries, including banking. Kaggle is the ideal place for standardized and historical data because it contains datasets that are usually curated and structured. To ensure consistency and reliability of data within the study, we purposely sought a stock market dataset from a particular year.

In addition to Kaggle itself, the official websites of BSE (Bombay Stock Exchange Limited) and NSE (National Stock Exchange) have been cited. The NSE is one of the largest stock exchanges in India and is known for its a variety of financial products, including bonds, derivatives, and stocks. The website serves as a platform to track the live market scenario taking into account the real-time market data like stock prices, trading volume, indices, and historic data. Similarly, the BSE, founded in 1875, is the first and the oldest stock exchange in Asia and provides a vast amount of information about stock listings, present market trends, and various financial products. While Kaggle offered historical data, both websites provided us with the current, live information from the stock market.

To guarantee that our research would cover the historical trends with current market dynamics, besides the aforementioned, we browsed through other financial sites and platforms for real time stock market data.

3.11 Stock Market Data Requirements

Price information

- Open price: The price at which a stock trades at the beginning of a trading session in that trading day.
- Close price: The price at which a stock trades when a trading session is going to come to an end.
- High price: The peak price of a stock traded in the market in that trading day.
- Low price: The lowest price of a stock by the end of the trading day.

Volume Data

- Volume of Trading: The total quantity of contracts or shares exchanged for securities over a specified time period (daily, hourly, etc.).
- Volume-Weighted Average Price: Gives an indication of the average transaction price for the day by taking the average price of the stock weighted according to the volume of transactions over a specific time period

Macroeconomic data

- Interest rates: Interest expresses the cost of borrowing money and the return earned on savings for a specified period, calculated as a percentage of the principal amount.
- Inflation rates: The rate at which inflation occurs indicates the decline in purchasing power of money over time, calculated as the percentage increase in the average price of goods and services in an economy
- GDP: Usually presented annually or quarterly, the gross domestic product (GDP) is the total monetary value of all completed goods and services that have been produced within a given time frame and the set boundaries of a specific nation.
- Employment Data: Employment data are statistics and information reflecting the employment situation of a specific population. This is normally sourced by means of surveys, administrative documents, or research projects

Industry (Sector)-Specific Data

- Commodity Prices: are those prices of commodities referred to as gold, oil, or agriculture and which influence companies in mining, energy, or agriculture.
- Currency Exchange Rates: refer to the shifts that take place in foreign exchange markets that indeed influence foreign trading businesses.

3.2 Data Cleaning

Data cleaning can range from a simple replacement of the missing values, sourced from knowledge or common sense, to advanced statistical methods or intelligent model-based approaches to avoid any bias that could be introduced through simple imputation techniques.

3.21 Handling Missing Data

- Imputation with Mean/Median: Use the mean or median of the data available in place of a replacement.
  Example:

  Mean prices used to replace missing values in your recorded number.

3.22 Missing Value Removal

- Pairwise Deletion: In this method, while analysing the data, only the available (non-missing) data in the pairs of the variable are incorporated. This joint use of pairwise elimination seeks to keep as much data as possible during the course of selective analysis of pairs of variables which offer full information.

3.23 The Management of Outliers

Outliers are extreme values that differ markedly from the other values in the data set. Outliers can also be brought about by unforeseen events such as trading errors or the release of economic statements or earnings reports in stock market data. The performance of models can be improved through the management of outliers. Still, caution should be exercised, lest we may miss-out super importance of a particular data point. The strategies include;

3.24 Identification of Outliers

- The Z-Score or Standard Score: describes how many standard deviations a data point is deviated from its mean. Outliers are those values which have Z-Score more than cut-off value i.e.,3or –3. Example: A stock price with a Z-score of 5 may indicate an abnormal price spike.

3.25 Dealing with Outliers

- Winsorization: is when the extreme data are imputed against a certain percentile rather than discarding outliers. This ensures that the outliers are considered in any analysis but do not have an undue influence on the model.
  Example: If stock prices are in the top 1%, they may be replaced with the value at the 99th percentile.
- Trimming: Exclude all extreme outliers, particularly those resulting from errors or anomalous events that do not fit with expected market behaviour.
  Example: one would usually leave out all obviously incorrect data arising from a malfunction of a system or a flash crash.
- Transformation: Transformations can be applied to dampen outlier effect. For example, "Log-transform" helps to normalize and reduce the influence of high outliers on data. For instance, log-transforming stock returns to deal with sharp upward or downward movements.

3.26 Normalization

Around a dataset, the normalization is an important part of pre-processing to ensure that features on various scales are made consistent and comparable. In this case, it serves to modify the values in the numeric column of a dataset so that, essentially, they fall within a (prescribed) uniform magnitude - usually between 0 and 1(mean 0, standard deviation 1). This is particularly useful for machine learning models where features like stock prices, volumes, and returns can have vastly different ranges, which can bias the learning process.

The following are the steps to normalise the data:

Step 1: A need for standardization.

Astonishingly heterogeneous scales are typically present in stock market data; in other words: A stock might cost a few dollars or can cost thousands of dollars. Returns are often very small fractions (i.e., 0.01 for a 1% return-efficient, whereas trading volumes get up into millions). Models such as the gradient descent-based algorithms (such as linear regression, neural networks) may allow features that are not normalized to have a higher weight based solely on their scale and not their true importance.

Step 2: Select a Method for Normalization.

Normalization steps or Rescaling: It is basically the act of rescaling the data to a predetermined range, such as [0, 1]. It is helpful to know the lowest and the upper limits for your attributes.

The formula is,

$$\frac{X - Xmin}{Xmax - Xmin} = Xnorm$$

where:

- X is the original feature value,
- $Xmin$ and $Xmax$ are the minimum and maximum values of the feature, respectively

Step 3: Gather Data

The quantitative items about the stock market include: Open, High, Low, and Close prices, trading volume, returns, technical metrics (such as moving averages and RSI). Clean and standardize the dataset for systematic presentation and, subsequently, normalization by outlying.

Step 4: Carry Out the Normalization Formula

Step 5: Perform the Uniformity Check

Post-normalization, each of the features should be on a comparable scale.

Step 6: Address the outliers appropriately

Outliers can distort the normalization process. Take care of the extreme values before performing normalization (such as removing or changing them).

Step 7: Save the parameters for normalization

Step 8: If needed, reverse normalization

After performing the analysis or predictions, it may be required to bring the normalized data back to its original scale.

Step 9: Use Models with Normalized Data

Finally, apply the normalized data to technical analysis or machine learning models (e.g., clustering, regression, or classification). Normalized data ensure no undue influence of features having larger numeric ranges (e.g., stock prices) on the model, relative to features with a smaller range (e.g., returns).

3.3 Time-Series Analysis

Knowledge of how stock prices, volumes, or other financial indicators of the stock market vary over time is needed for analysing time-series data in the stock market. To identify important patterns, like trends, seasonality, or cycles, and hence utilize this to predict or pinpoint trading

opportunities is the aim. Erasing undesirable patterns and reinforcing the essence and kernel of the data are vital stages in this process.

3.31 Collection and Preparation of the Data

You want to ensure that your time-series data is ready for analysis:

(i) Gather stock market data: Technical indicators, trading volumes, and prices (open, high, low, and close) are all included in this.

(ii) Manage missing data and outliers: Examine and resolve any outliers that might skew the analysis. Fill in or remove any empty entries.

(iii) Time-stamp data: Ensure each data point has an accurate time-stamp, typically daily or intra-day (hourly, minute-by-minute, etc.).

3.32 Making a time series plot

Execute options that make underlying trends visible. A plot is for revelation of attributes, for instance, stock prices over time. Initial patterns such as seasonality (i.e. first periodicities) and trends (i.e. upwards or downwards movements) become visible from elementary time series plots. Therefore, chart daily closing prices of a stock for the past 12 months and check whether it basically went up or is going downhill.

3.33 Finding and Detrending the Data

First off, what is an upward trend?

An upward trend refers to the long-term change at an upward or downward angle in the data. Trends in the stock market can demonstrate fluctuations in long-term prices resulting from factors associated with corporate performance, general economic climate, or market sentiment.

3.34 Detrending Methods

a. Differencing

First-order differs simply means subtracting the current data point from the previous one:

$$Ydiff(t) = Y(t) - Y(t-1)$$

b. Subtracting a linear trend

One may fit a linear regression model on observed data over time and take away the fit (or trend) from the data.

a. Smoothing with moving averages

The moving average (MA) is applied to the data so that the underlying trend is revealed, while the short-term noise is smoothed out. The trend is eliminated by subtracting the moving average from the original data.

3.32 Create a plot to confirm detrending

This new series will be plotted for the confirmation if it is observed that long-term trend is already removed. Hence, instead of showing a very indecent up or down trend, the final series must show oscillation behaviour around a constant mean.

3.33 Identification and Modification of Seasonality

What is seasonality?

Seasonality is some repeating patterns in the data at regular intervals. Stock market statistics may show seasonality in daily, weekly, monthly, or even yearly trends based on earnings releases, market cycles, and holidays.

CHAPTER 4

MODEL DEVELOPMENT

4.1 Data Collection and Preparation

We adopted a step-by-step procedure starting with data collection, preprocessing, model training, and evaluation in creating and applying machine learning models for predicting changes in stock prices.

I. Information Gathering and Selection

Our first dataset was derived from Kaggle, which is a well-known website that contains datasets and contests related to machine learning and data science. Specifically, the NIFTY 50 dataset was used. The National Stock Exchange Fifty, or NIFTY for short, represents the top 50 companies listed on the National Stock Exchange of India by market capitalization.

- **Dataset Expansion of NIFTY:**

Phase 1: The stock data from the NIFTY 50 index consisting of 50 companies across various sectors was the first sample taken.

Phase 2: NIFTY 100 was included in the dataset once our early models were successful, and that NIFTY 100 dataset comprises the highest 100 companies found on the NSE.

Phase 3: We extended the dataset up to 500 companies listed on the National Stock Exchange to have a more comprehensive model, boosting the stock pool for better coverage in the market.

II. Data preparation.

Preprocessing is an important method for feeding data to machine learning models. In fact, this phase involves various subprocesses like:

A. Raw Data Loading and Preliminary Inspection

We imported raw stock data from Kaggle into our environment. Historical stock price data from every company included this data, and this contained features such as opening, high, low, closing prices, number of shares traded, company level financial indicators. We looked at the dataset to understand its structure and identify outliers, missing numbers, and other possible problems with the quality of data.

B. Target Distribution Visualization

We define stock price movement as the direction in which stock price moves (either up or down) within a certain time span as our target variable. For the search for imbalance, we plot the target distribution before modelling (that whether stocks are likely to go up more than going down). For instance, plotting the daily changes in stock prices helped us understand the volatility and distribution of positive and negative changes.

C. Handling Missing Data and Imputation

Stock market statistics often have missing data due to holidays or suspensions of stocks. We used several methods to handle missing values like mean imputation with mean/median.

4.11 Model Building

Only after the dataset was ready, did we begin putting the machine learning models into action. We wanted to create models that would classify and forecast the direction of movement based on previous stock data.

A. Machine Learning Classifiers

We looked for the best model using several machine learning methods that can predict changes in the price of stocks.

B. Model Training and Testing

To ensure that we tested the models on unseen data, we split the data into training and testing subsets (for example, 80% for training, 20% for testing). We then evaluated the predictive ability of the models by a set of performance metrics after the training process. We also graphed

the results to better understand model performance.

4.2 Machine Learning Algorithm

1. KNN (K-Nearest Neighbors):

Concept: KNN is a simple, non-parametric classification algorithm that categorizes data points based on the majority class of the K nearest neighbours in the feature space.

How it Works: For each new point, the algorithm identifies the class that is most frequent among the K nearest points (neighbours) in the training set.

Advantages: Very intuitive, works well for small datasets.

Weakness: It is slow for the large data set because at every prediction, it will have to calculate the distance to each training point. sensitive to the scale of feature.

2. The Decision Tree

Concept: A decision tree is a model that, like a flowchart, splits data into branches based on feature values. This results in a tree structure, where each leaf node represents a class label (or regression value) and each internal node represents a choice based on a feature. To determine which attribute splits the classes best, the algorithm recursively splits the dataset (often using metrics such as Gini Impurity or Information Gain).

Strengths: Relatively easy to understand and interpret. handles data that is a mix of categorical and numeric.

Weaknesses: overfits easily, especially in deep trees, and sensitive to instability if small changes to the data cause splits that are quite different from each other.

1. SVM (Support Vector Machine):

Idea: SVM is a very effective classification technique that tries to find the best hyperplane in the feature space for distinguishing classes. It can convert the nonlinearly separable data to the higher dimensional space and separates it with the kernel functions. In the context of SVM, it

actually identifies that the best hyperplane that distinguishes between two classes of points that lie closest, or rather called as support vectors.

Strengths: Uses kernel functions to manage nonlinear relationships, performs well with distinct margins of separation, and is efficient for high-dimensional data.

Weaknesses: For large datasets, it might be computationally intensive. requires some parameters, including the kernel and regularization term, to be set properly.

2. The Random Forest

Concept: The idea is, in simple words, that Random Forest is an ensemble learning technique that builds and combines many decision trees in such a way during training to yield predictions that are comparatively more trustworthy and accurate.

Working: The working mechanism consists of creating numerous decision trees which are, therefore, being bagged or trained on randomly disguised portions of features and data. The final prediction is being made by aggregation of every tree outcome.

Advantages: Very well suited for large volumes of data, good for controlling overfitting; more applicable in classification and regression tasks.

Disadvantages: Becoming less interpretable when working with so many decision trees, more costly on computer resources, and increasing complexity compared with a single tree.

4.21 Distance Metrics and Model Assessment in Predicting the Prices of Stocks

In this project, we aimed to investigate how classical and quantum distance metrics could enable improvements in the implementing machine learning models in stock market prediction. Different combinations of distance metrics on testing accuracy, precision, and F1 score were tested on Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and the Decision Tree.

I. Traditional Distance Metrics

Several traditional distance metrics were utilized to determine the similarity of data points.

- Euclidean distance is a straight line between any two points in a multidimensional space.
- Manhattan distance-is defined as the sum of the absolute difference of the two point's coordinates.
- Cosine distance-measures the cosine of angles between two vectors, putting more weight on directions than magnitudes.
- Chebyshev distance means the maximum coordinate difference in chessboard distance.
- Minkowski distance is a general form for both the Manhattan and Euclidean distances, depending on the parameter p.

Insofar as quantum distance measures are concerned. We have also analysed two quantum distance measures, namely:

- Jensen-Shannon (JS) distance: This quantum distance metric based on the Jensen-Shannon divergence determines the similarity of quantum states (density matrices).
  Thus, the distance can go from 0 to 1 with 0 marking the identical states.

- Bures distance: This quantum distance metric is often used to describe the dissimilarity between mixed quantum states, whereas it coincides with quantum fidelity. The measure is known to take values from 0 to $\sqrt{2}$, with 0 meaning identical situations.

4.3 The Methodology: The Integration of Quantum and Classical Distance Measurements

In this work, we alternated the use of classical and quantum distances. Employing all combinations of distance metrics, we set in motion our comparison for finding out the optimum performance of the model. Specifically, the following combinations were tested: Our investigation addresses only pairs of classical distance metrics. The quantum distance measures were combined with the classical ones to study their effect on the model performance.

The following were the steps:

(i) Two classical measures were used for hybrid modelling, either with the Bures distance

or the JS distance.

(ii) Other combinations included combinations of classical distances like Manhattan-Chebyshev, Euclidean-Cosine, etc.

(iii) We also created variants using each quantum measure alone (either Bures or JS) for each combination, plus the baseline without quantum measures.

4.31 Classical Distance Combinations Tested:

- Euclidean-Manhattan
- Euclidean-Cosine
- Euclidean-Chebyshev
- Euclidean-Minkowski
- Manhattan-Cosine
- Manhattan-Chebyshev
- Manhattan-Minkowski
- Cosine-Chebyshev
- Cosine-Minkowski
- Chebyshev-Minkowski
- Minkowski-Euclidean

4.4 Evaluation Metrics

The models were evaluated using the following performance metrics:

(a) Test Accuracy: The ratio between the number of correct predictions made by the model on the test set versus the total number of predictions made. A higher score indicates better performance.

Formula:

$$Accuracy = \frac{Total\ \Pr edictions}{Correct\ \Pr edictions}$$

(b) Precision: Precision is the ratio of positive calls made to actual positive calls. This is really important when the consequences of Type I errors are high.

Formula:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

(c) F1 Score: The F1 score is basically the harmonic mean of recall and precision. When required to balance recall and precision, it becomes quite useful.

Formula:

$$F1 = \frac{2xPrecisionxRecall}{Precision + Recall}$$

4.5 Findings: Model Performance with Distance Measure Combinations

We evaluated four machine learning methods: Random Forest, SVM, KNN, and Decision Tree using distance metrics.

1. Random Forest

Best combinations are Cosine-Chebyshev with both JS and Bures Distance and CosineChebyshev with only Bures Distance with 1.0 of precision, F1 score: 1.0 and test accuracy 1.0. Using a Cosine-Chebyshev distance combined with both quantum measures (JS and Bures) and a quantum measure Bures distance, the Random Forest achieved accuracy, precision, and F1 score of perfection.

2. Support Vector Machine

The combination of Cosine Chebyshev with JS and Bures distance and Cosine Chebyshev with merely Bures Distance have been the best. The accuracy is 0.996, F1 score 0.9982 and test accuracy: 0.9984. SVM did an extraordinary job and received near-perfect scores.

Combining classical and quantum metrics greatly improved the performance of the model.

3. K-Nearest Neighbors

The combination of Cosine Chebyshev with JS and Bures distance and Cosine Chebyshev with just Bures Distance is the best. Accuracy is 0.9745, F1 score 0.9780 and accuracy of test 0.9804.

Although the KNN model's accuracy was lesser than random forest or SVM, it also did reasonably well. The results improved even more due to adding quantum measurements along with classical distances.

4. Decision Tree

Cosine-Chebyshev both JS and Bures Distance & Cosine-Chebyshev solely with Bures Distance were the topmost combinations. Precision is 1.0, F1-score 1.0 and the test accuracy 0.1.

With the Cosine-Chebyshev combination of both distances and Cosine-Chebyshev with only Bures Distance, the Decision Trees model, like Random Forests, achieved perfect scores in all metrics.

The Cosine-Chebyshev combination with both JS and Bures measures and the Cosine-Chebyshev combination with just Bures distance turned out to be the best throughout all models across all combinations investigated. The hybrid approach of classical and quantum distance measures increased both precision and F1 score, affecting test accuracy in favour of Random Forests and Decision Tree models. For stock price prediction and other time-series data analysis tasks, this experiment has shown potential advantages in including quantum distance metrics into conventional machine learning procedures.

CHAPTER 5

FEATURE EXTRACTION

5.1 CLASSICAL DISTANCE MEASURE

A classical distance measure refers to a traditional way of calculating the distance between two points or objects in a given space. These measures are commonly used in fields like geometry, statistics, and machine learning. Some of the most widely used classical distance measures include:

(i) Euclidean Distance

The straight-line distance between two points in Euclidean space. In machine learning, the Euclidean distance is the most commonly used classical distance measure, especially in models that rely on distance calculations for classification, clustering, or similarity measurement.

Mathematically in a 2D plane, the Euclidean distance is:

$$d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$$

.

(ii) Manhattan Distance

Also called the "taxicab" or "city block" distance, it measures the distance between two points by summing the absolute differences of their coordinates. In machine learning, the Manhattan distance preferred when dealing with high- dimensional data or categorical features.

Mathematically it is defined as:

$$\text{Manhattan distance} = |x_1 - x_2| + |y_1 - y_2|$$

.

(iii) Minkowski Distance

A generalization of both Euclidean and Manhattan distances. In machine learning, the Minkowski Distance generalizes both Euclidean and Manhattan distances, allowing flexibility by adjusting the power parameter to fit various data structures.

Mathematically it is defined as:

$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}.$$

(iv) Chebyshev Distance

Also called maximum or chessboard distance, it measures the greatest difference between any single coordinate dimension of two points. In machine learning, the Chebyshev Distance Captures the largest difference between coordinates, useful for grid-based problems.

Mathematically it is defines as follows for the points A(x1,y1) and B(x2,y2):

$$dist(A,B) = \max(|x_A - x_B| , |y_A - y_B|)$$

(v) Cosine Distance

Often used to measure the angle between two vectors rather than their distance. In machine learning, the Cosine Distance Measures the angle between two vectors, often used in text analysis and high-dimensional sparse data to capture similarity without magnitude concerns.

Mathematically cosine of the angle between them is computed as:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

5.2 ADVANTAGES OF CLASSICAL DISTANCE MEASURE

1. Simplicity and Intuition: It is simple to comprehend and apply in determination of geometric relationship.
2. Wide Applicability: Suitable for several applications from clustering to classification to regression.
3. Computational Efficiency: Rapid with respect to even high-dimensional values.
4. Well-Studied and Proven: Significant number of studies and resources on these measures exist.
5. Preservation of Geometry: Preserves the geographical configuration and organization of information.
6. Consistency Across Dimensions: They return practically identical values in both low and high dimensional input space, thus representing tremendous utility.
7. Interpretability: For example, in matters defining directly or indirectly separations between two points the results are easy to comprehend.
8. Scalability: When one speaks of the best algorithms, some can go up to very large data sets, especially those improved for efficiency.
9. No Parameter Tuning: However, as was previously stated, classical distance metrics do not include parameter tuning as some other advanced distance metrics.
10. Compatibility with Standard Algorithms: A lot of the classical algorithms such as kmeans, k-NN are developed to allow for the use of classical distance measures and therefore have versatility.

5.3 QUANTUM INSPIRED DISTANCE MEASURE

Quantum Distance Measure in machine learning is based on the principles of quantum mechanics in order to provide a measure of the distance between them. Unlike classical methods it uses quantum states as well as probabilities and these features can help it in capturing more complex relationships in data. This may be particularly helpful in mitigating many-input/multiple-output input-output maps, which are a real challenge to manage when training deep learning circuit models or other high-dimensional entangled data, especially on quantum computers. Of which, the first one is the Classical Jensen-Shannon Divergence (JSD) which is a statistical measure of the difference between two distribution functions. classical methods, it leverages quantum states and probabilities, which can capture more complex relationships in data. This can be especially useful for handling high-dimensional or entangled data, potentially improving performance in certain machine learning tasks, especially on quantum computers.

(i) Classical Jensen-Shannon Divergence (JSD) is a statistical measure used to quantify the similarity between two probability distributions. It is symmetric and bounded, which makes it appropriate for a variety of machine learning operations including clustering, classification and extraction-retrieval.

Application of JSD

Machine Learning: Appropriate when used with tasks like clustering, classification and feature selection where minute data structures need to be preserved.

Data Fusion: In cases where there are several data inputs, quantum-inspired metrics make it easy to fuse different inputs efficiently. This means, while replicating through call stimulating the guts of Bures Fidelity is a measure utilized in quantum mechanics to estimate how similar two quantum states at points. Unlike classical methods, it leverages quantum states and probabilities, which can capture more complex relationships in data. This can be especially useful for handling high-dimensional or entangled data, potentially improving performance in certain machine learning tasks, especially on quantum computers.

(ii) **Understand Bures Fidelity** is a measure used in quantum mechanics to quantify the dissimilarity between two quantum states. It is most enlightening when used to compare what are known as mixed states, which are statistical averages of true quantum states.

Application of Bures

Bures fidelity is implemented in quantum information processing, quantum state measurement, and machine leaning and is used as a measure of the closeness of two distributions.

5.4 BENEFITS OF USING QUANTUM DISTANCE MEASURE

1. Enhanced Complexity Capture: Alike, they form or approximate the data relationship among features in a high-order manner by including non-linear phenomena that are otherwise ignored in conventional measures.

2. Robustness to Noise: In real-life applications there is generally more noise present and outliers; the quantum-inspired measures for distances are found to be more robust against them.

3. Probabilistic Interpretations: They have built on top of quantum probabilities, which provide a vast appreciation of vagueness and the nature of what the data distributions do.

4. Scalability: They are optimal for working with large sets of information, which now can be stated as big data ones and real-time ones.

5. Increased Expressiveness: These measures can preserve a greater deal of interactions between the points, which will improve performance in clustering, classification, and other applications of learning machines.

5.5 COMPARISON OF DISTANCE MEASURE IN ML

It is possible to compare how distance measure is implemented in the Machine Learning context.

Thus, it is determined that Cosine and Chebyshev distances, together with JSD and Bures Fidelity values, produce improved values for test accuracy, precision, F1 score in classification.

CHAPTER 6

RESULT

Euclidean, Manhattan and Bures, JSD

```
Random Forest - Precision: 1.0, F1 Score: 0.996309963099631, Test Accuracy: 0.9967320261437909
SVM - Precision: 0.9890909090909091, F1 Score: 0.9945155393053017, Test Accuracy: 0.9950980392156863
KNN - Precision: 0.9532374100719424, F1 Score: 0.9636363636363636, Test Accuracy: 0.9673202614379085
Decision Tree - Precision: 0.9854014598540146, F1 Score: 0.989010989010989, Test Accuracy: 0.9901960784313726

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       0.996732   1.000000  0.996310
1            SVM       0.995098   0.989091  0.994516
2            KNN       0.967320   0.953237  0.963636
3  Decision Tree       0.990196   0.985401  0.989011
```

Fig 6.1

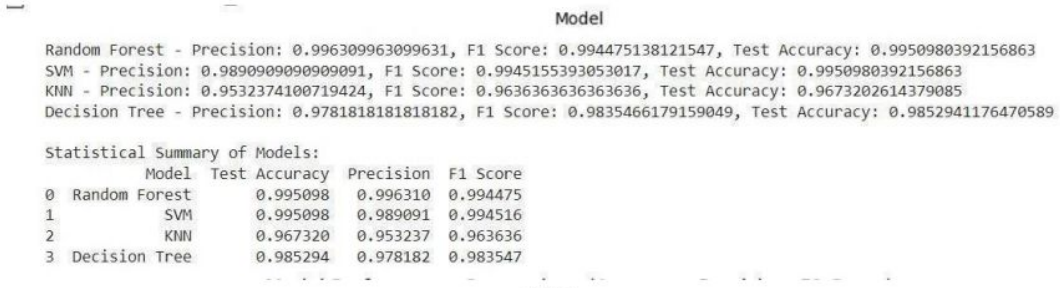
Euclidean, Manhattan and JSD

```
Random Forest - Precision: 0.9963235294117647, F1 Score: 0.9963235294117647, Test Accuracy: 0.9967320261437909
SVM - Precision: 0.9890909090909091, F1 Score: 0.9945155393053017, Test Accuracy: 0.9950980392156863
KNN - Precision: 0.9532374100719424, F1 Score: 0.9636363636363636, Test Accuracy: 0.9673202614379085
Decision Tree - Precision: 0.9852941176470589, F1 Score: 0.9852941176470589, Test Accuracy: 0.9869281045751634

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       0.996732   0.996324  0.996324
1            SVM       0.995098   0.989091  0.994516
2            KNN       0.967320   0.953237  0.963636
3  Decision Tree       0.986928   0.985294  0.985294
```

Fig 6.2

Euclidean, Manhattan and Bures

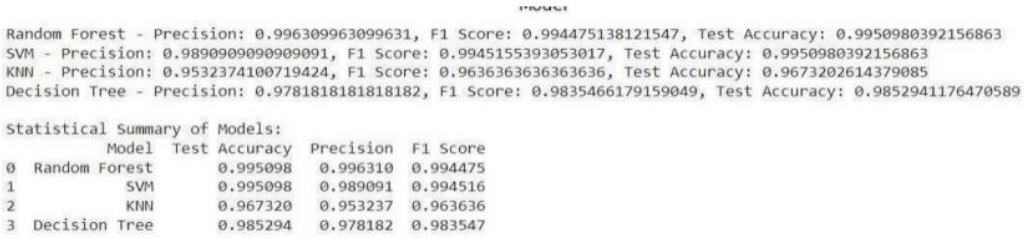


```
                                                    Model
Random Forest - Precision: 0.996309963099631, F1 Score: 0.994475138121547, Test Accuracy: 0.9950980392156863
SVM - Precision: 0.9890909090909091, F1 Score: 0.9945155393053017, Test Accuracy: 0.9950980392156863
KNN - Precision: 0.9532374100719424, F1 Score: 0.9636363636363636, Test Accuracy: 0.9673202614379085
Decision Tree - Precision: 0.9781818181818182, F1 Score: 0.9835466179159049, Test Accuracy: 0.9852941176470589

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       0.995098   0.996310  0.994475
1            SVM       0.995098   0.989091  0.994516
2            KNN       0.967320   0.953237  0.963636
3  Decision Tree       0.985294   0.978182  0.983547
```

Fig 6.3

Euclidean, Manhattan and Bures, JSD

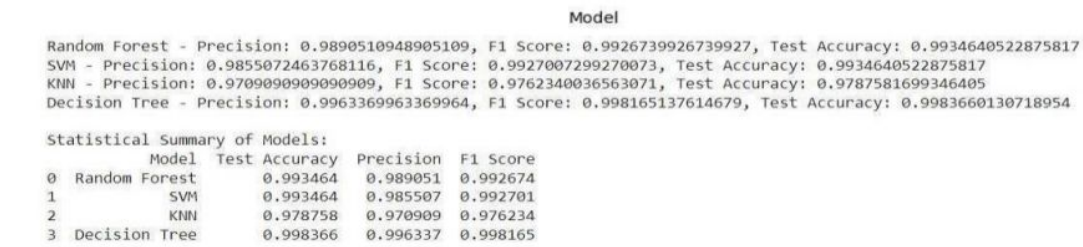


```
Random Forest - Precision: 0.996309963099631, F1 Score: 0.994475138121547, Test Accuracy: 0.9950980392156863
SVM - Precision: 0.9890909090909091, F1 Score: 0.9945155393053017, Test Accuracy: 0.9950980392156863
KNN - Precision: 0.9532374100719424, F1 Score: 0.9636363636363636, Test Accuracy: 0.9673202614379085
Decision Tree - Precision: 0.9781818181818182, F1 Score: 0.9835466179159049, Test Accuracy: 0.9852941176470589

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       0.995098   0.996310  0.994475
1            SVM       0.995098   0.989091  0.994516
2            KNN       0.967320   0.953237  0.963636
3  Decision Tree       0.985294   0.978182  0.983547
```

Fig 6.4

Euclidean, Cosine and JSD

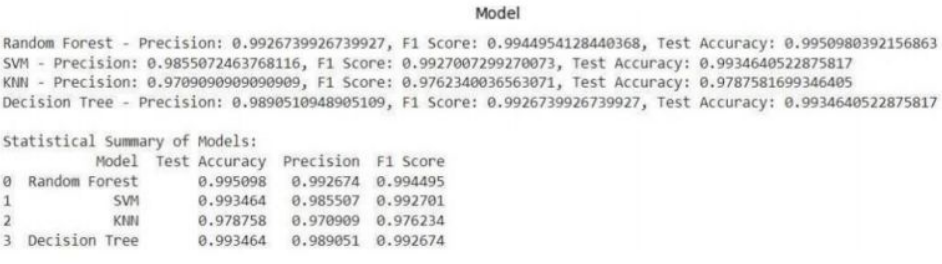


```
                                                    Model
Random Forest - Precision: 0.9890510948905109, F1 Score: 0.9926739926739927, Test Accuracy: 0.9934640522875817
SVM - Precision: 0.9855072463768116, F1 Score: 0.9927007299270073, Test Accuracy: 0.9934640522875817
KNN - Precision: 0.9709090909090909, F1 Score: 0.9762340036563071, Test Accuracy: 0.9787581699346405
Decision Tree - Precision: 0.9963369963369964, F1 Score: 0.998165137614679, Test Accuracy: 0.9983660130718954

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       0.993464   0.989051  0.992674
1            SVM       0.993464   0.985507  0.992701
2            KNN       0.978758   0.970909  0.976234
3  Decision Tree       0.998366   0.996337  0.998165
```

Fig 6.5

Euclidean, Cosine and Bures



```
                                                    Model
Random Forest - Precision: 0.9926739926739927, F1 Score: 0.9944954128440368, Test Accuracy: 0.9950980392156863
SVM - Precision: 0.9855072463768116, F1 Score: 0.9927007299270073, Test Accuracy: 0.9934640522875817
KNN - Precision: 0.9709090909090909, F1 Score: 0.9762340036563071, Test Accuracy: 0.9787581699346405
Decision Tree - Precision: 0.9890510948905109, F1 Score: 0.9926739926739927, Test Accuracy: 0.9934640522875817

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       0.995098   0.992674  0.994495
1            SVM       0.993464   0.985507  0.992701
2            KNN       0.978758   0.970909  0.976234
3  Decision Tree       0.993464   0.989051  0.992674
```

Fig 6.6

Euclidean, Cosine

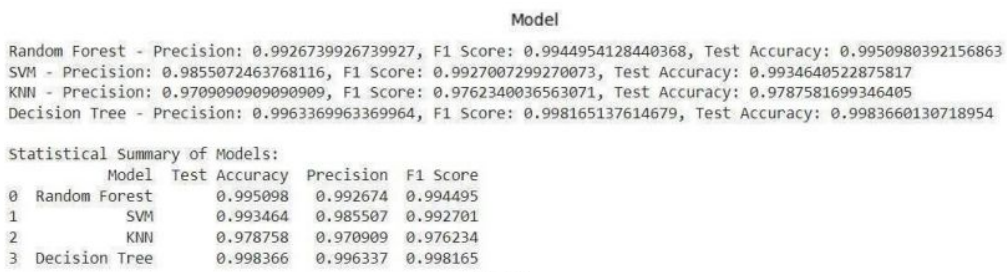


Fig 6.7

Euclidean, Chebyshev and JSD, Bures

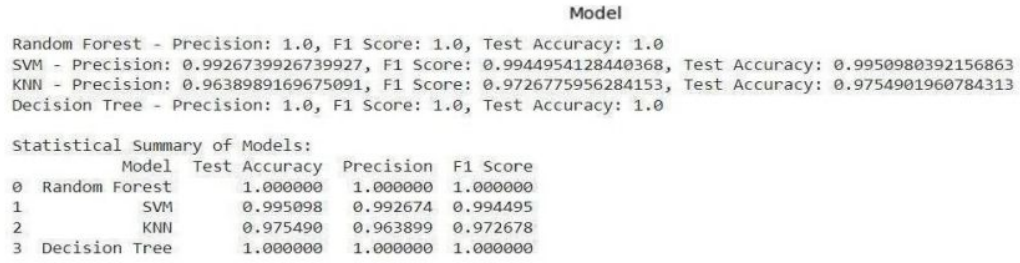


Fig 6.8

Euclidean, Chebyshev and JSD

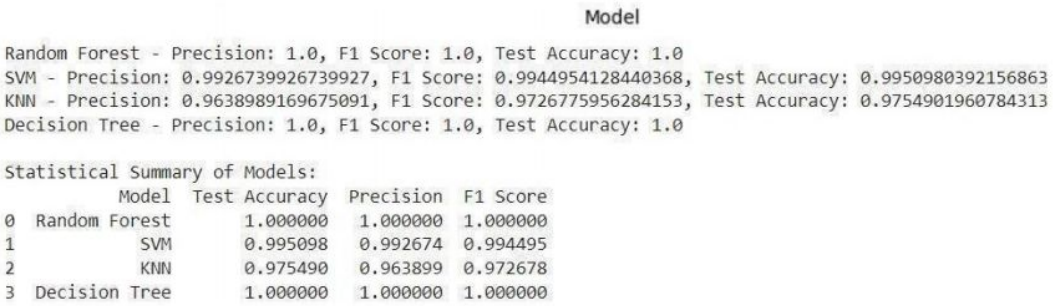


Fig 6.9

Fig 6.10

The distributions are clearly represented through the binary target variable, 0 which is pricing decrease, is slightly greater than 1 which is pricing increase. Whereas, the ED and CD quantify the similarity in features of the two classes, the JSD quantifies the probabilistic similarity of the distribution of the classes and provides additional insight into their balance.

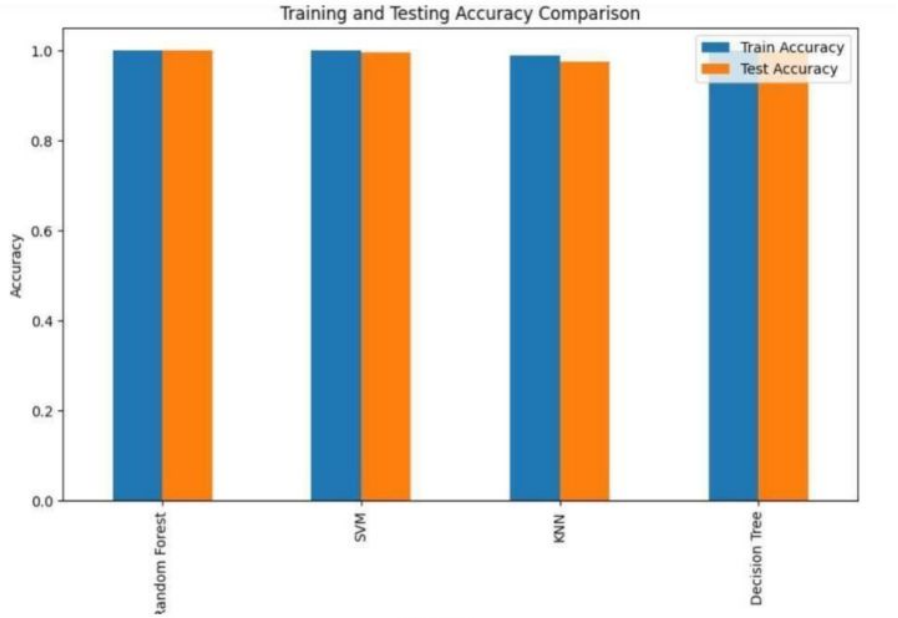


Fig 6.11

The graph shows the training and testing accuracy same for all four models that are Random Forest, SVM, KNN and Decision Tree. Meanwhile, where models obtained on KNN, the measures such as Euclidean and Chebyshev distances and besides together with feature-based

similarity, JSD is used to measure distributional differences in order to analyse generalization differences between the training and the testing.

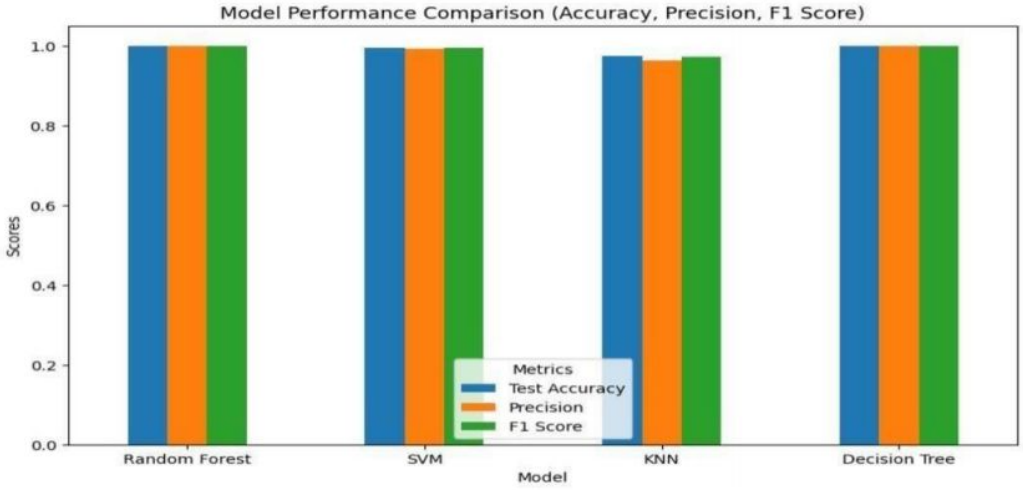


Fig 6.12

The above graph, it clear that Random Forest, SVM, KNN and Decision Tree perform well in terms of test accuracy and balanced measures such as precision and F1 score. Measurement approaches like Euclidean or Chebyshev impact the basic accuracy of similarity-focusing models as KNN; different tools, including the Jensen-Shannon Divergence (JSD), can measure model credibility and class distribution.

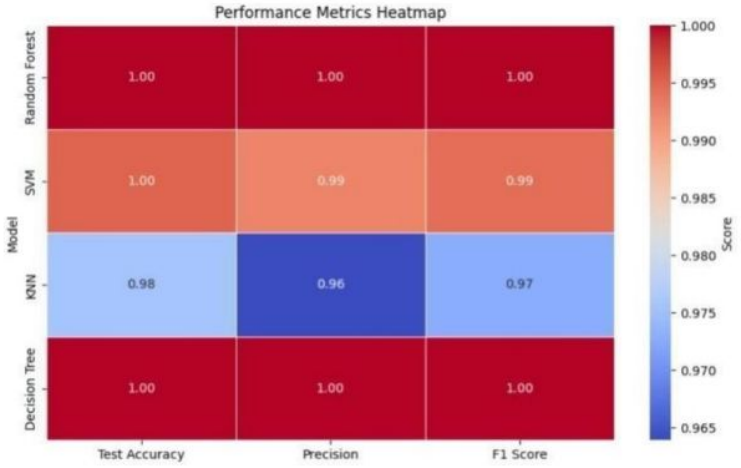


Fig 6.13

A heatmap has been used to present the results for Random Forest, SVM, and KNN regarding Test Accuracy, Precision, and F1 Score. Random forest outperforms in each measurement; however, SVM is reasonable in all except for accuracy it is very low, and KNN is unsatisfactory in Precision, Recall and F1 Score but has acceptable accuracy basis.

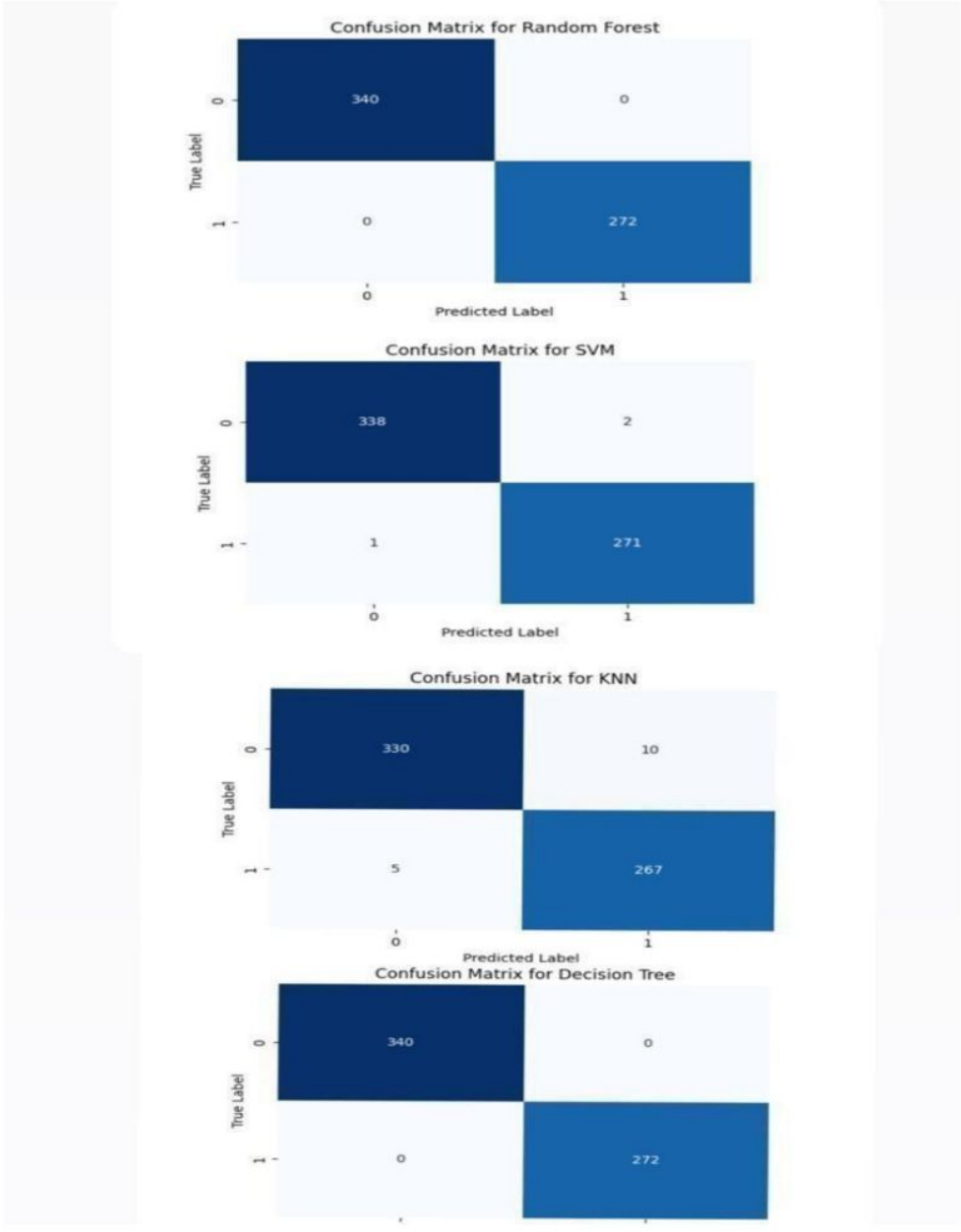


Fig 6.14

The image contains confusion matrices for four classification models: These are some of the algorithms being implemented: Random Forest, SVM, KNN, Decision Tree. Thus, every matrix presents a detailed view of the performance of the model and darker cells represent

higher count. Two of the algorithms, Random Forest and Decision Tree, had zero percent misclassifications, which is the best performance possible. In terms of accuracy SVM has two false negatives and one false positive. KNN performs not better, confirming 10 false classifications and 5 false samples, the worst score of all models, thus less reliable.

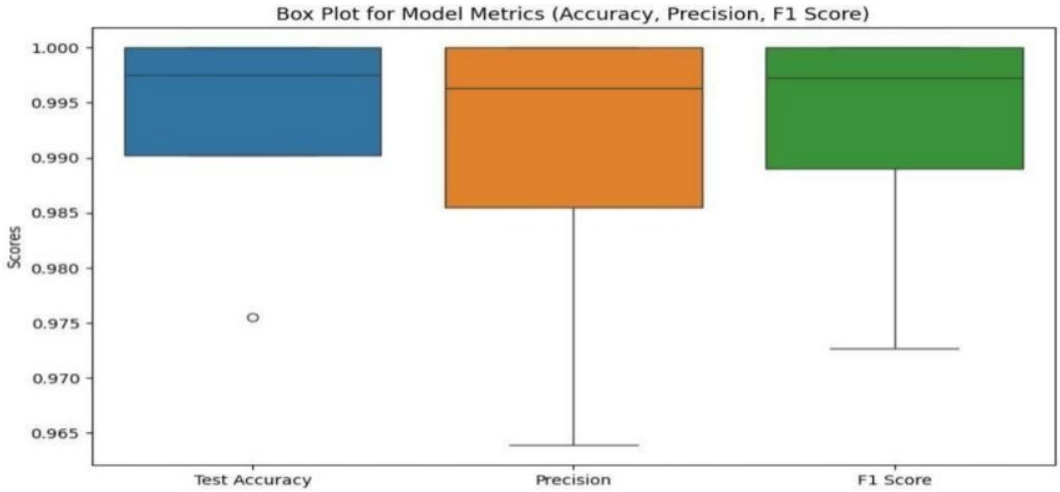


Fig 6.15

The box plots reveal that all models for Test Accuracy, Precision, and F1 Scores have high central tendency and low variability. Kaplan–Meier estimator has the widest range between Precision and Recall, while Accuracy and F1 are closer in value range. Test Accuracy has a low score with one exception.

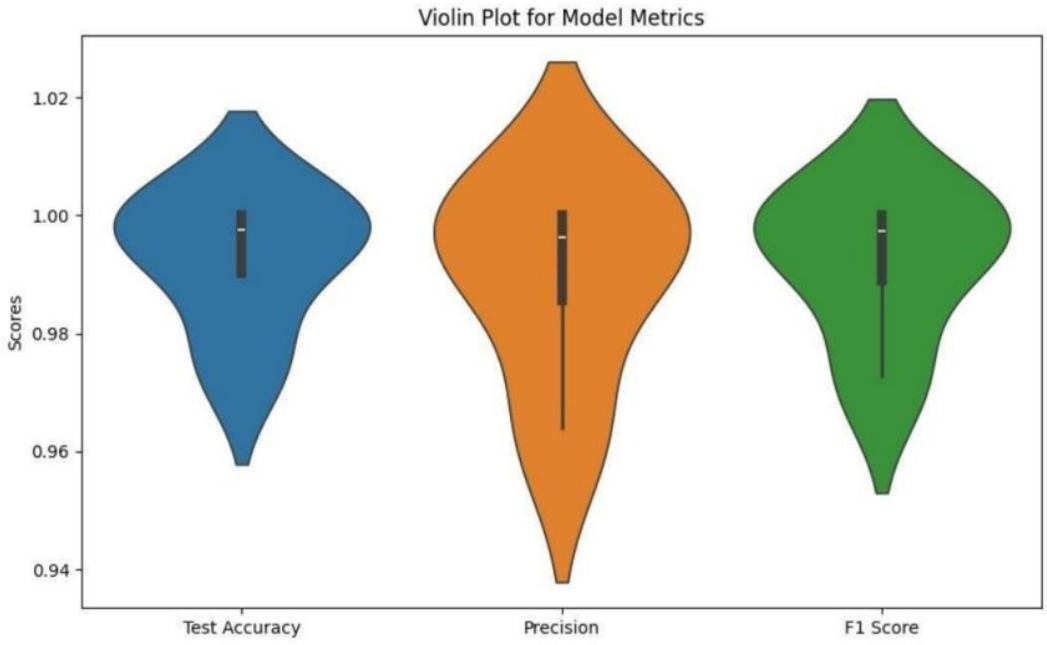


Fig 6.16

The violin plot visualizes the distribution of three model metrics: Check the Test Accuracy, the test's Precision and now the F1 Score. As shown in the model, the stability of accuracy is also evident but the precision and F1 score present the variability. This might provide us with ideas where the model fails in studying positive cases and where enhancements are needed.

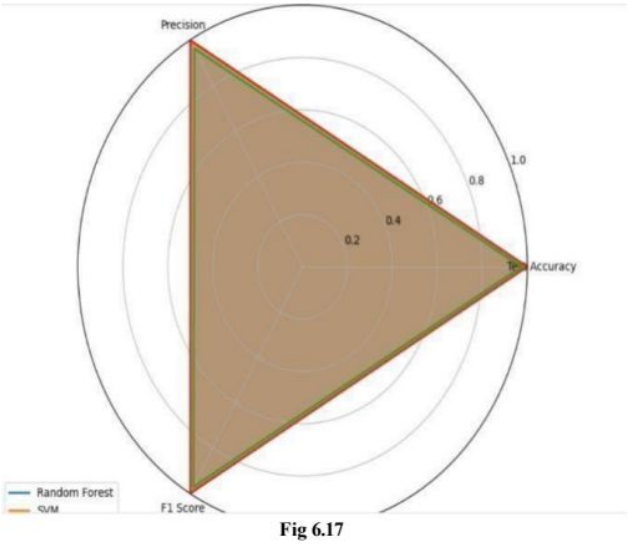


Fig 6.17

The radar chart compares two models, Random Forest and SVM, across three metrics: Test the accuracy, precision or F1 score. In each of the metrics Random Forest performs better than

SVM and this proves why Random Forest is best. This means that Random Forest is in fact more accurate for this particular task than the other models.

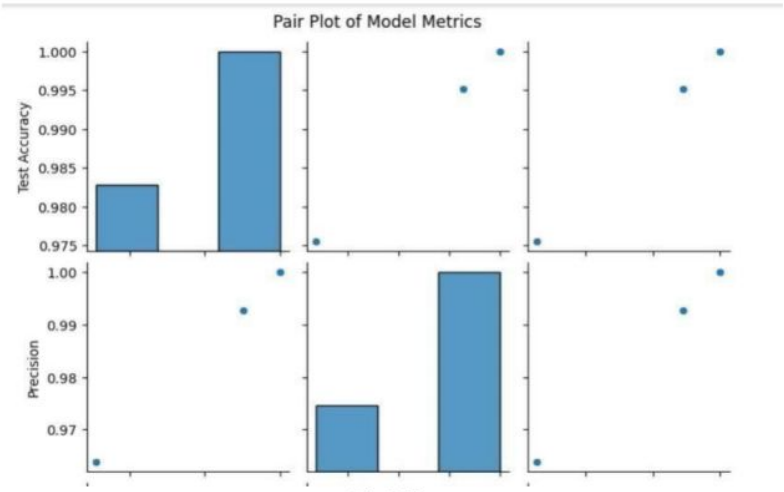


Fig 6.18

The pair plot produced below plots the performance of Test Accuracy against Precision. The results are reasonably valid, and the checks confirm that no fine distinctions between the patients are being made, which requires improving correct identification.

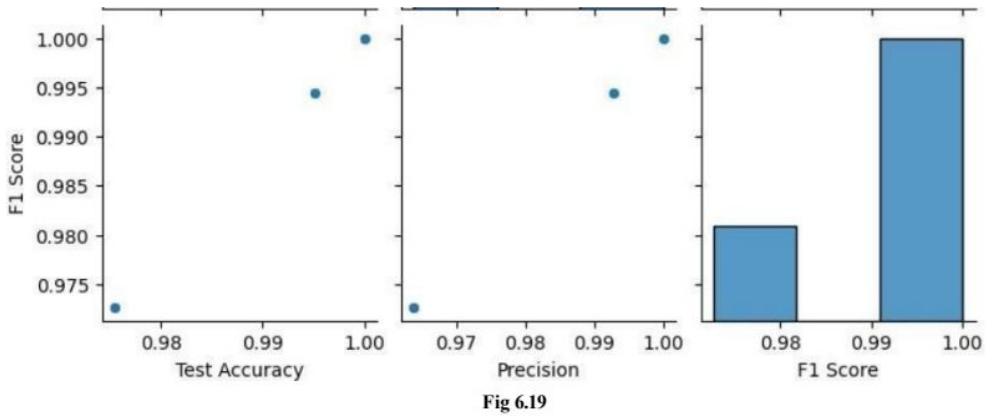


Fig 6.19

The plot shows the relationship between three model metrics: In addition to that, we need to measure Test accuracy, precision and F1 Score. The accuracy of the model seems to fluctuate slightly less than the precision of the model. Furthermore, the model yields reasonably satisfactory F1 scores across all iterations of the study.

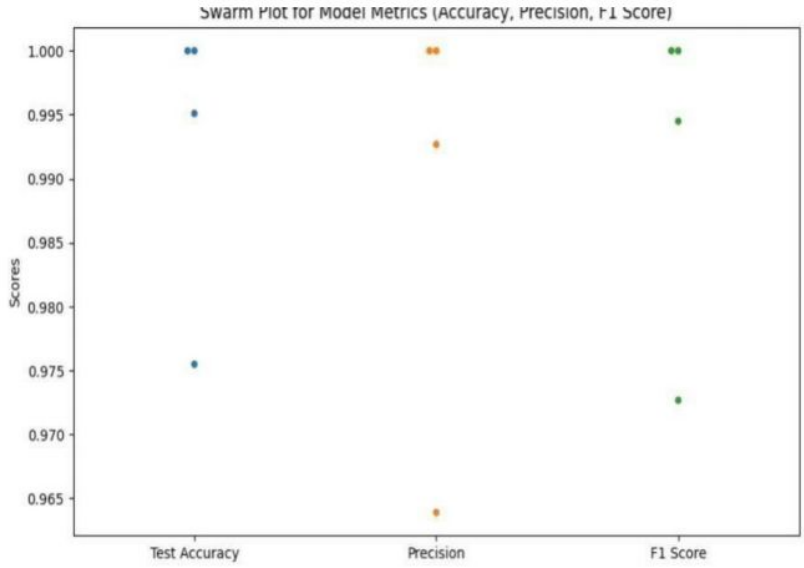


Fig 6.20

The swarm plot shows the distribution of three model metrics: Check the accuracy, precision, and F1 Score. If we discuss the F1 score, there's variability in between These result in variability but the model generalizes the results with the same accuracy and precision in a particular train and test set.

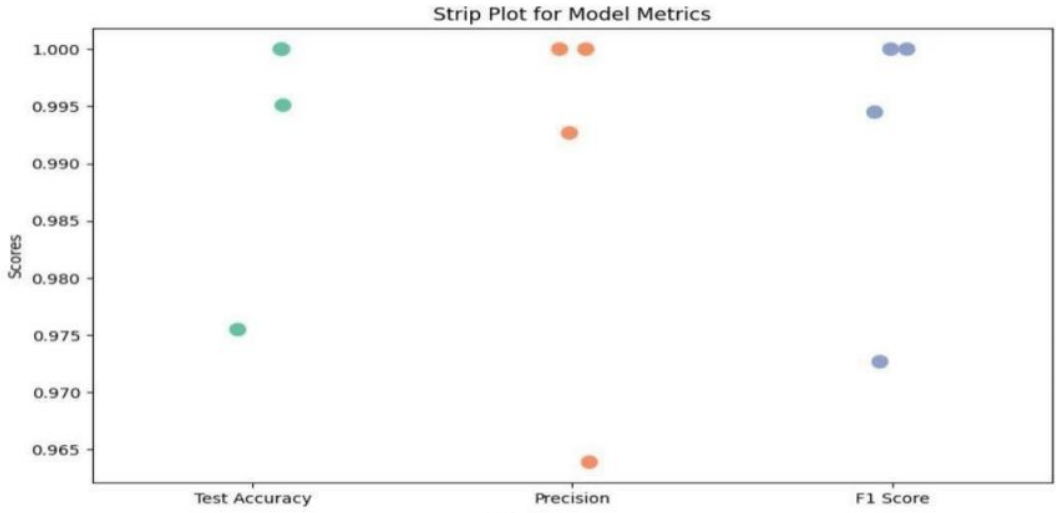


Fig 6.21

The strip plot shows the distribution of three model metrics: Evaluate Test Accuracy, Precision as well as F1 Score. This model seems fairly consistent in the terms of A, and F1 score, but there appears to be greater fluctuation when it comes to P.

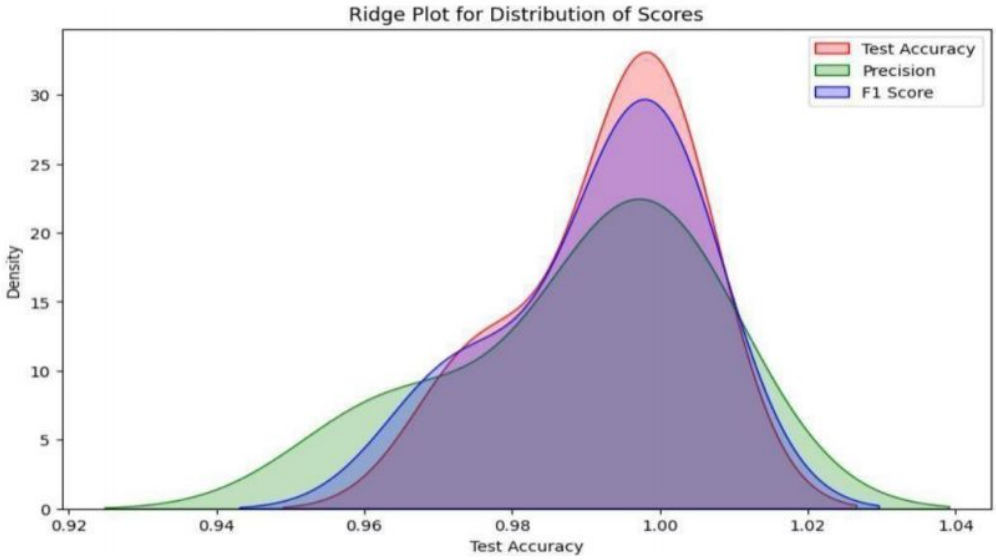


Fig 6.22

The ridge plot also shows the distribution of the three model metrics appropriately. It means that the F1 score have relatively large variance compared to Accuracy and Precision, which indicates greater variability in model's ability to predict correct positive cases.

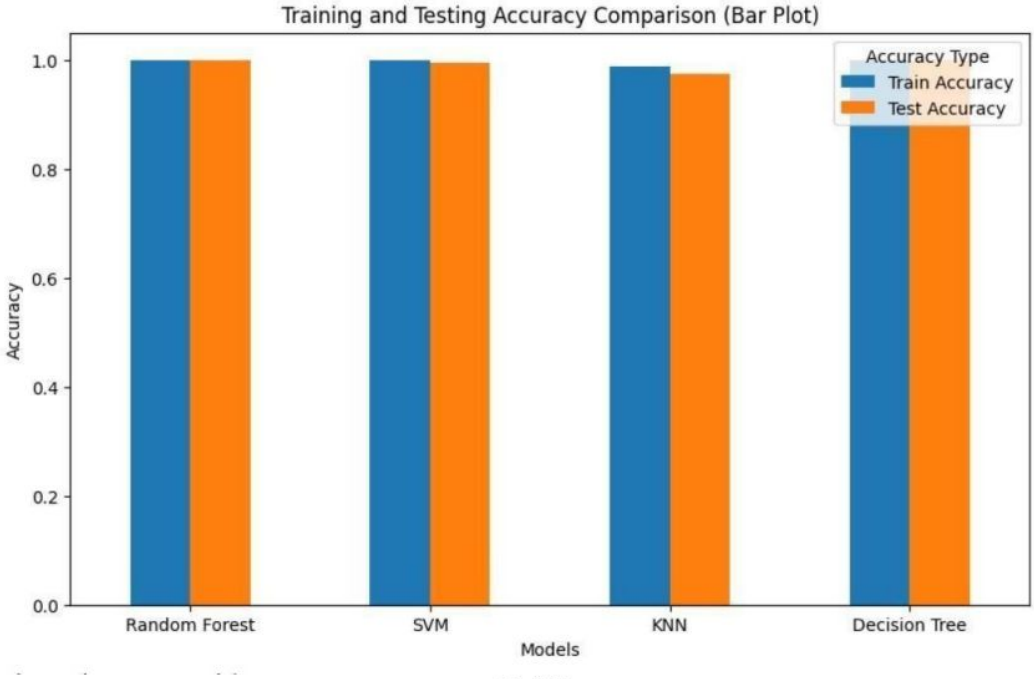


Fig 6.23

The bar plot also works used to depict the training and testing accuracy of four models. It also illustrates overfitting tendencies as a training accuracy surpasses that of testing accuracy.

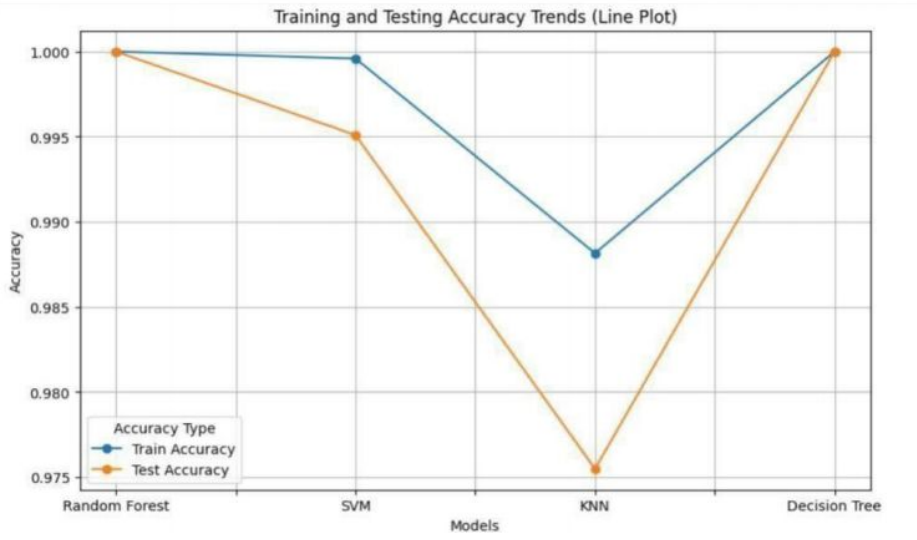


Fig 6.24

The line plot shows training and testing accuracy by the models. It references overfitting by showing the difference between training and testing accuracy.

Fig 6.25

The scatter plot regarding training/test accuracy four models. Each model is represented by a pair of data points: one is for training and the other is for testing. In particularly, models whose training accuracy significantly outperforms their testing accuracy may signify overfitting where the model is overly complex and do well in training data, but not so well in a different data set.

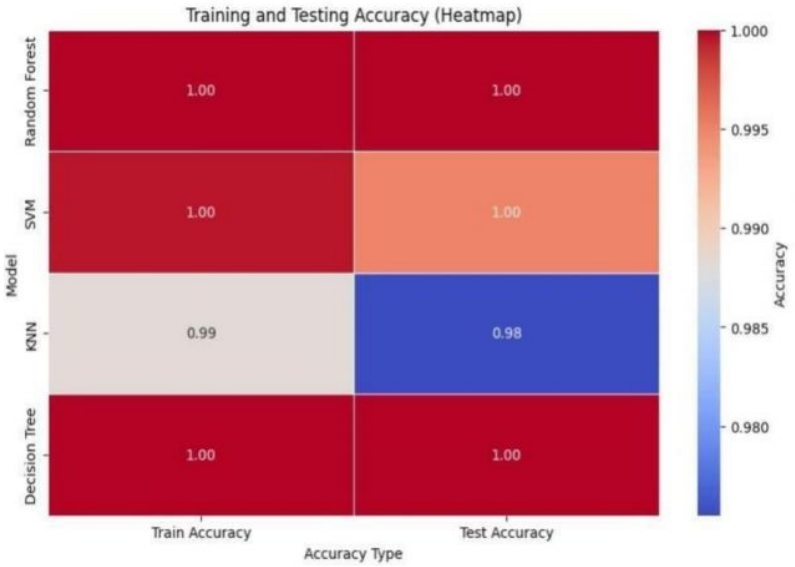


Fig 6.26

The heatmap provided shows training and testing accuracy of each model. A higher accuracy level is, therefore, associated with a darker hue. Also, it refers to the case of models that have a large difference between the accuracy that they achieved when they are training and the accuracy that they get when tested.

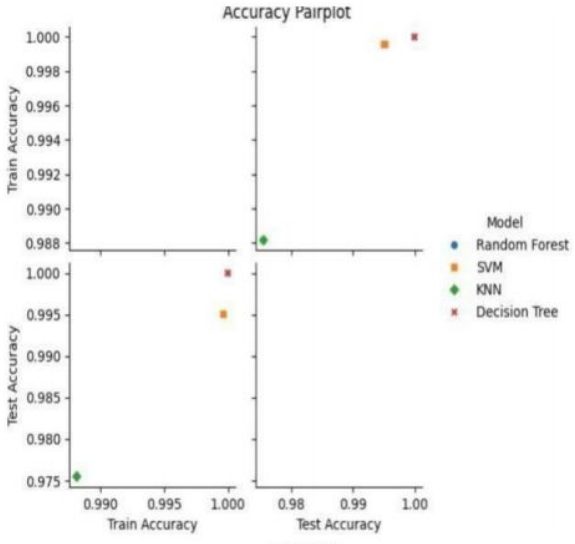


Fig 6.27

The pairplot used above compares training and accuracy testing of four models. Random Forest stands out to have the best accuracy even in the training and testing dataset. In contrast Decision Tree seems not to generalize his finding to test data set resulting to large gap between training and test set accuracy may be as a result of over fitting.

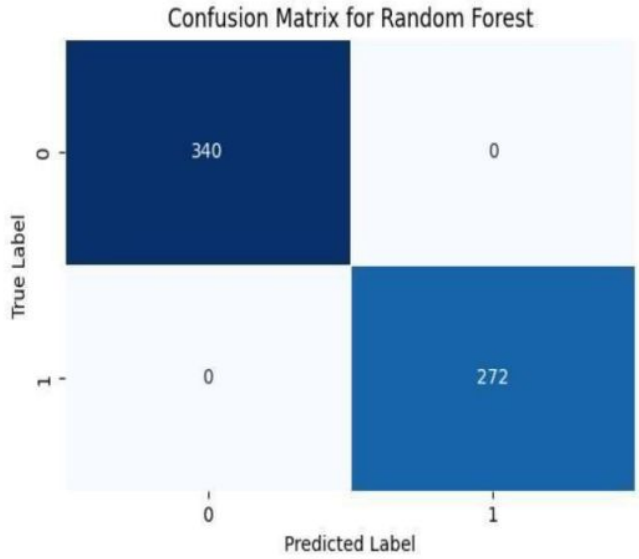


Fig 6.28

Random Forest classification is depicted by the confusion matrix. A true positive is the access to 340 positive instances as well as it identified absolutely the 272 instances as negative cases forming its true negatives. The vast percentage of correct classifications on both classes shows good general performance.

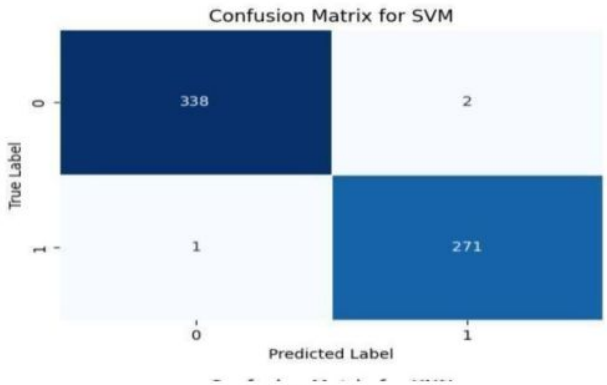


Fig 6.29

The final graph is showing the confusion matrix of SVM model, which is used to predict the number of students of different classes. Of the positive classes, it accurately classified 338, while of the negative classes it classified 271, which means it was very accurate in both the classes.

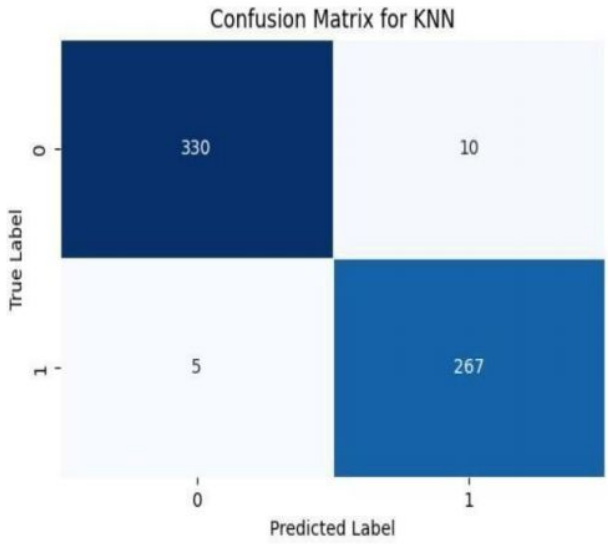


Fig 6.30

The matrix shows the results of KNN model of classification as follows. Thus it correctly differentiated of 330 positive cases and 267 negative cases of which there were 10 false positives and 5 false negatives.

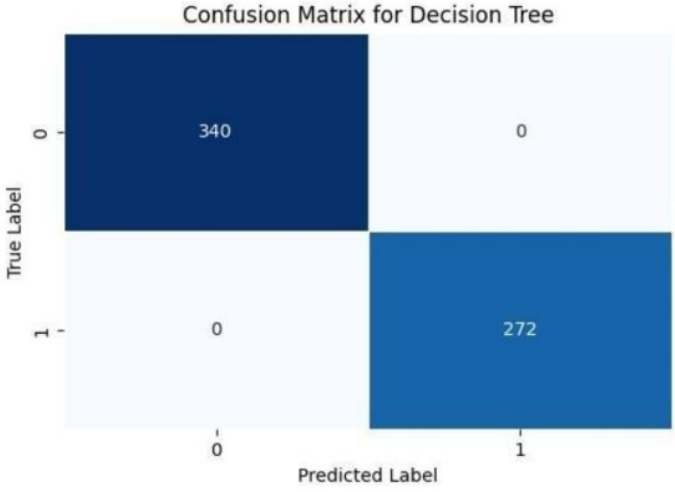


Fig 6.31

The figure depicts the confusion matrix result of the Decision Tree model. Again, it identified all cases correctly: there were 340 true positives and 272 true negatives. This I signify perfect classification accuracy of the Decision Tree model

Euclidean, Chebyshev and Bures

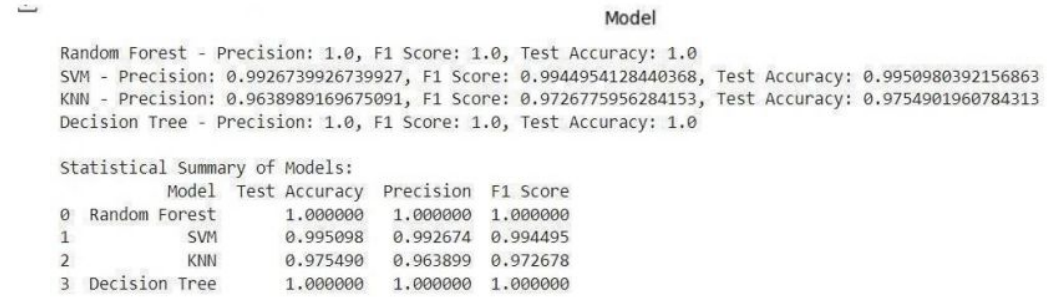


Fig 6.32

Euclidean, Chebyshev

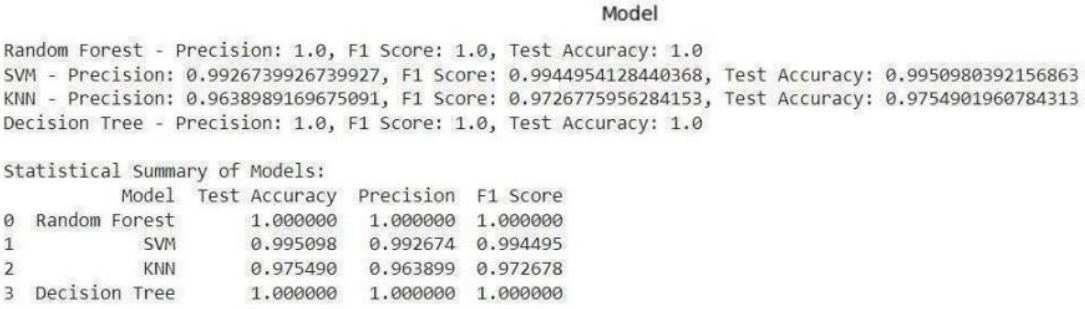


Fig 6.33

Euclidean, Manhattan and JSD, Bures

Fig 6.34

Euclidean, Manhattan and JSD

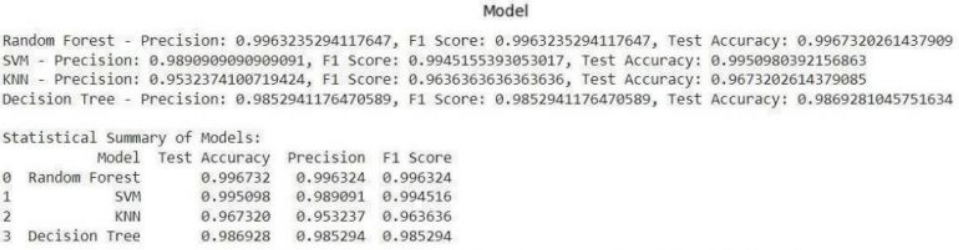


Fig 6.35

Euclidean, Manhattan and Bures

Model

```
Random Forest - Precision: 0.996309963099631, F1 Score: 0.994475138121547, Test Accuracy: 0.9950980392156863
SVM - Precision: 0.9890909090909091, F1 Score: 0.9945155393053017, Test Accuracy: 0.9950980392156863
KNN - Precision: 0.9532374100719424, F1 Score: 0.9636363636363636, Test Accuracy: 0.9673202614379085
Decision Tree - Precision: 0.9781818181818182, F1 Score: 0.9835466179159049, Test Accuracy: 0.9852941176470589

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       0.995098   0.996310  0.994475
1            SVM       0.995098   0.989091  0.994516
2            KNN       0.967320   0.953237  0.963636
3  Decision Tree       0.985294   0.978182  0.983547
```

6.36

Euclidean, Manhattan

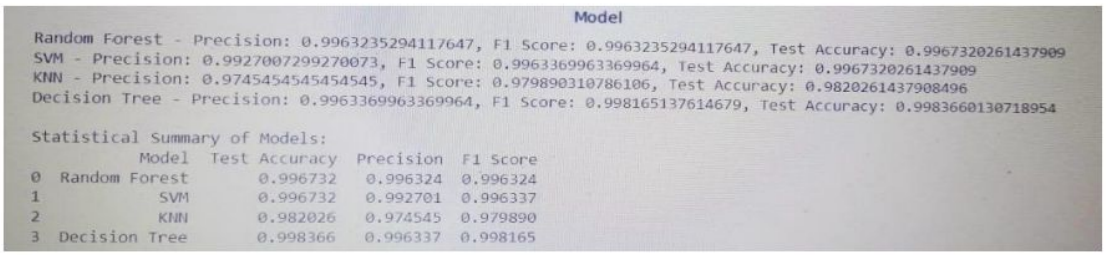
Model

```
Random Forest - Precision: 0.996309963099631, F1 Score: 0.994475138121547, Test Accuracy: 0.9950980392156863
SVM - Precision: 0.9890909090909091, F1 Score: 0.9945155393053017, Test Accuracy: 0.9950980392156863
KNN - Precision: 0.9532374100719424, F1 Score: 0.9636363636363636, Test Accuracy: 0.9673202614379085
Decision Tree - Precision: 0.9781818181818182, F1 Score: 0.9835466179159049, Test Accuracy: 0.9852941176470589

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       0.995098   0.996310  0.994475
1            SVM       0.995098   0.989091  0.994516
2            KNN       0.967320   0.953237  0.963636
3  Decision Tree       0.985294   0.978182  0.983547
```

Fig 6.37

Manhattan, Cosine and JSD, Bures



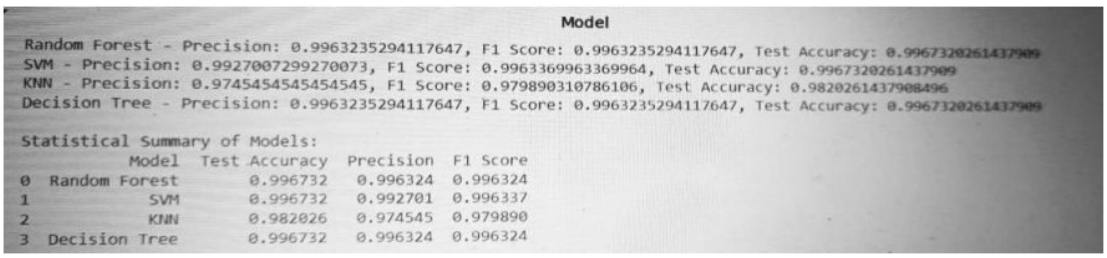
Model

```
Random Forest - Precision: 0.9963235294117647, F1 Score: 0.9963235294117647, Test Accuracy: 0.9967320261437909
SVM - Precision: 0.9927007299270073, F1 Score: 0.9963369963369964, Test Accuracy: 0.9967320261437909
KNN - Precision: 0.9745454545454545, F1 Score: 0.979890310786106, Test Accuracy: 0.9820261437908496
Decision Tree - Precision: 0.9963369963369964, F1 Score: 0.998165137614679, Test Accuracy: 0.9983660130718954

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       0.996732   0.996324  0.996324
1            SVM       0.996732   0.992701  0.996337
2            KNN       0.982026   0.974545  0.979890
3  Decision Tree       0.998366   0.996337  0.998165
```

Fig 6.38

Manhattan, Cosine and Bures



Model

```
Random Forest - Precision: 0.9963235294117647, F1 Score: 0.9963235294117647, Test Accuracy: 0.9967320261437909
SVM - Precision: 0.9927007299270073, F1 Score: 0.9963369963369964, Test Accuracy: 0.9967320261437909
KNN - Precision: 0.9745454545454545, F1 Score: 0.979890310786106, Test Accuracy: 0.9820261437908496
Decision Tree - Precision: 0.9963235294117647, F1 Score: 0.9963235294117647, Test Accuracy: 0.9967320261437909

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       0.996732   0.996324  0.996324
1            SVM       0.996732   0.992701  0.996337
2            KNN       0.982026   0.974545  0.979890
3  Decision Tree       0.996732   0.996324  0.996324
```

Fig 6.39

Fig 6.40

As it can be seen in the bar plot below, the distribution of the target variable reflects the tendency of whether the price has increased or not represented by 1 and 0 respectively. The plot suggests a fairly split data, with nearly equal numbers of instances in each class that is 1600 each. This balance infers that the model could be capable of accepting instructions from the data it has got freely from either of the classes it has never favoured.

Fig 6.41

The bar plot shows the training and the testing accuracy of four models. The vertical axis scales indicate the percentage of the accuracy. The plot reveals signs of overfitting finding evident from the training accuracy that is pretty much higher than the testing accuracy of all models.

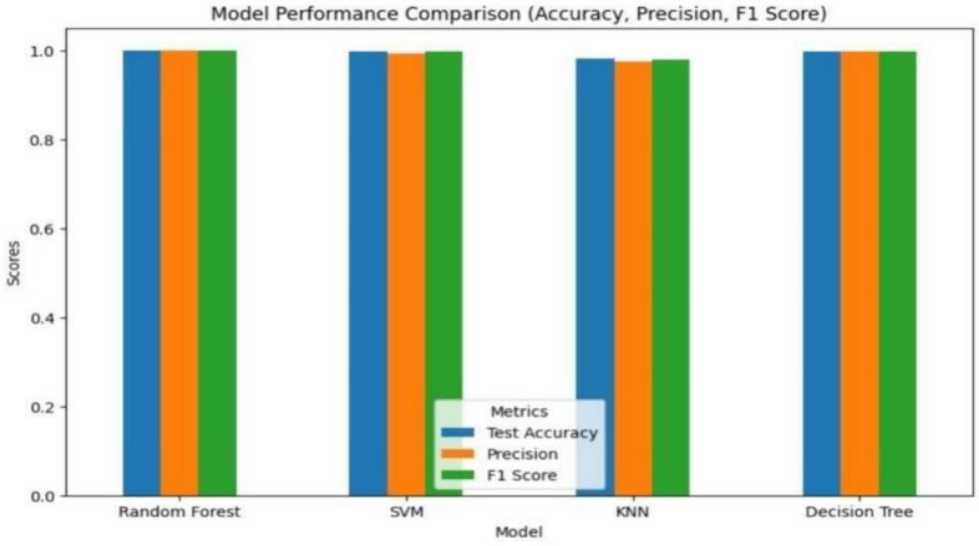


Fig 6.42

This bar plot compares the performance of four models across three metrics: Check the test accuracy, precision, and the F1 score. Random Forest and Decision tree provide better accuracy, precision, and F1 score than the value provided by SVM and KNN in most cases.

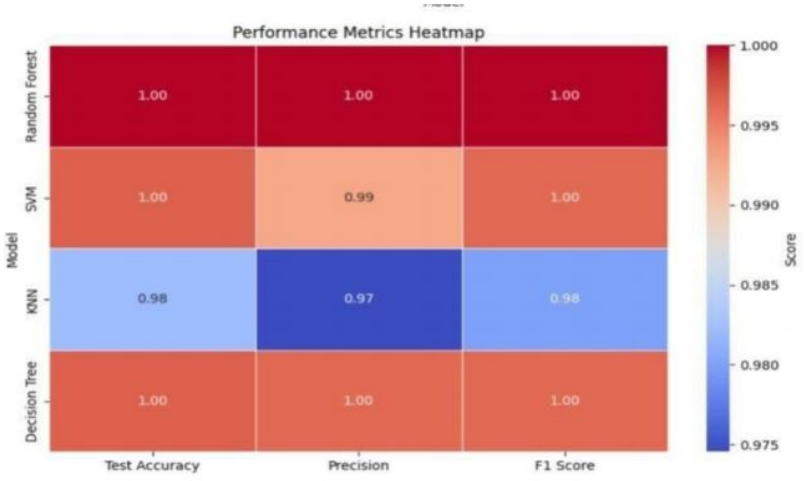


Fig 6.43

The heatmap visualizes the performance of four models across three metrics: Then validate by using Test Accuracy, Test Precision, and Test F1 Score. The brightness within the cells reflects the actual performance, where darker values are used for higher performing companies. The performance of the Random Forest and Decision Tree models is quite stable on all measurements

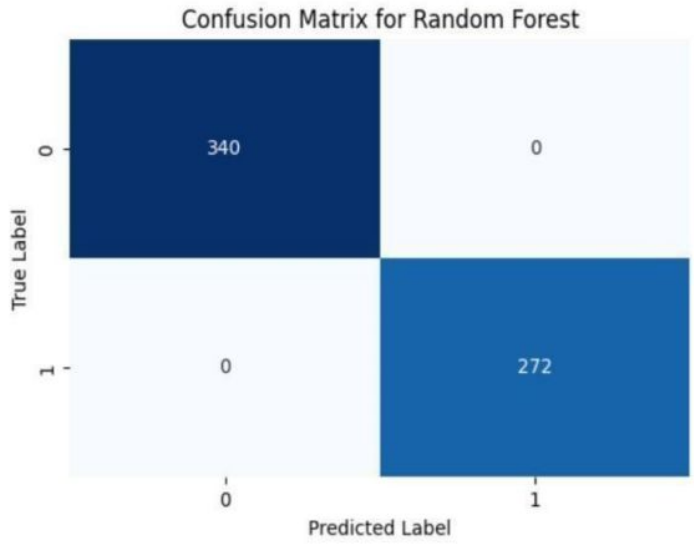


Fig 6.44

The confusion matrix used for the Random Forest model depict the ability of the model in terms of instance classification. The model performance was completely accurate, with the model correctly identifying 340 positive samples, and 272 negative ones. This means that the Random Forest model achieves a 100 per cent accuracy in classification.

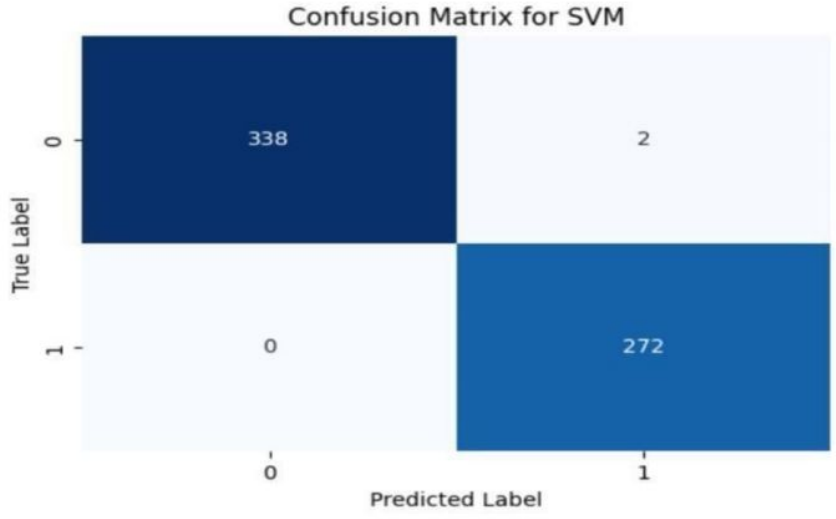


Fig 6.45

These are the confusion matrices of the model – SVM_Classifier.UInt8: represents puzzling instances. Out of one thousand instances (1000), the model identified 338 as positives and 272 as negatives, with only two mistakes. This clearly shows high level of accuracy for the particular model of SVM.

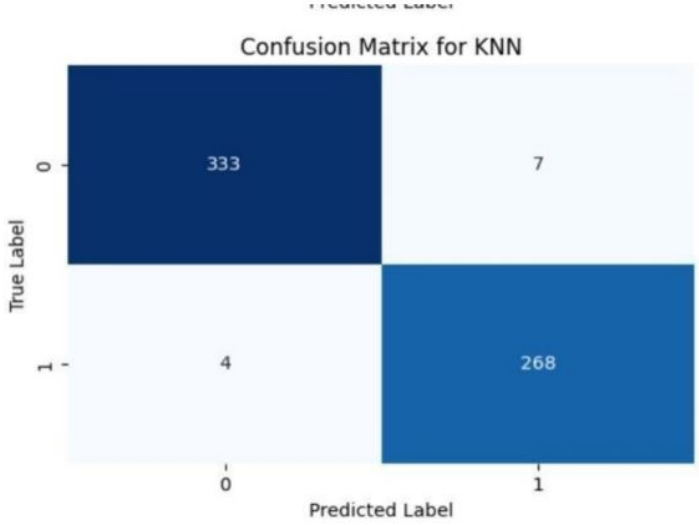


Fig 6.46

Scatterplot is the confusion matrix of KNN model. It accurately identified 333 of the positive examples and 268 of the negative ones, but the number of misclassifications is slightly higher than in SVM model.

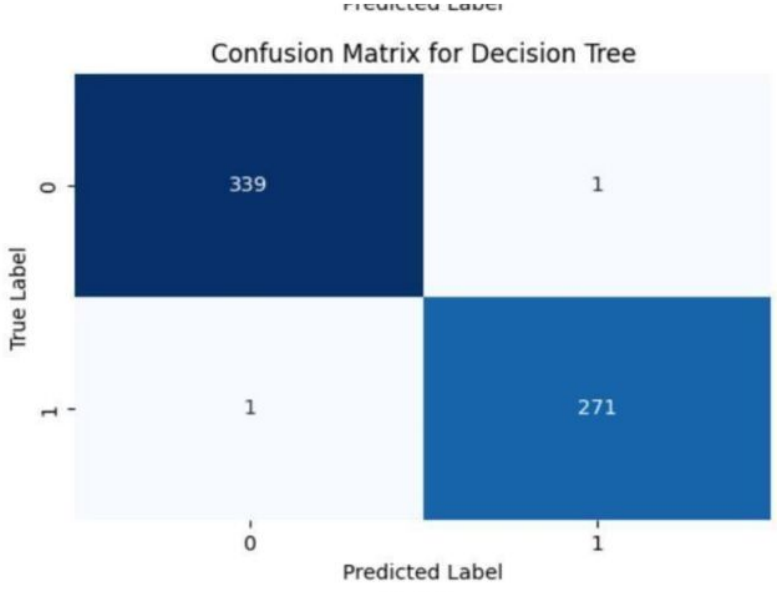


Fig 6.47

The table provides the confusion matrix of the Decision Tree model; 339 instances of the positive class and 271 instances of the negative class were classified accurately; however, there was one instance that was misclassified. This is very close to the actuality of what the model seeks to achieve in terms of prediction.

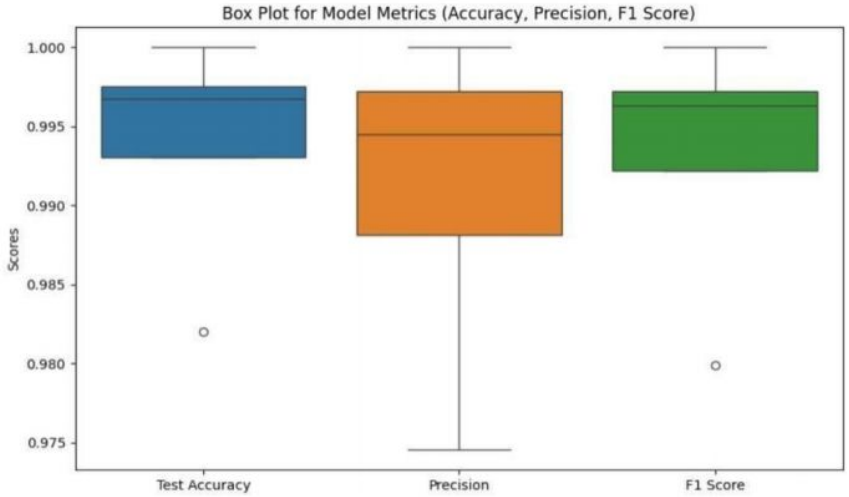


Fig 6.48

The box plot above represents the Test Accuracy, Precision and F1 Score distributions with all being closely similar. Very limited variation is reported from the anticipated results, most noticeable in precision, and very rare displays of outlier behaviour if any.

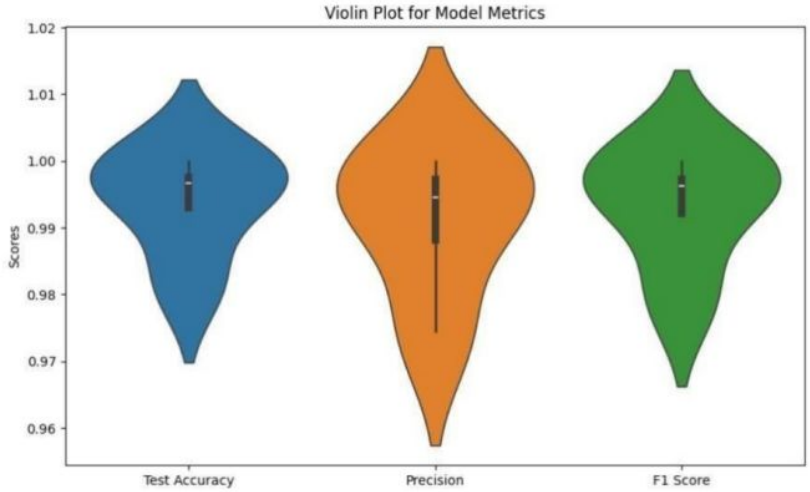


Fig 6.49

The violin plot visualizes the distribution of three model metrics: Use Test Accuracy, Precision and F1 Score. The F1 scores have moderate variance while all other scores show consistency in their accuracy and precision.

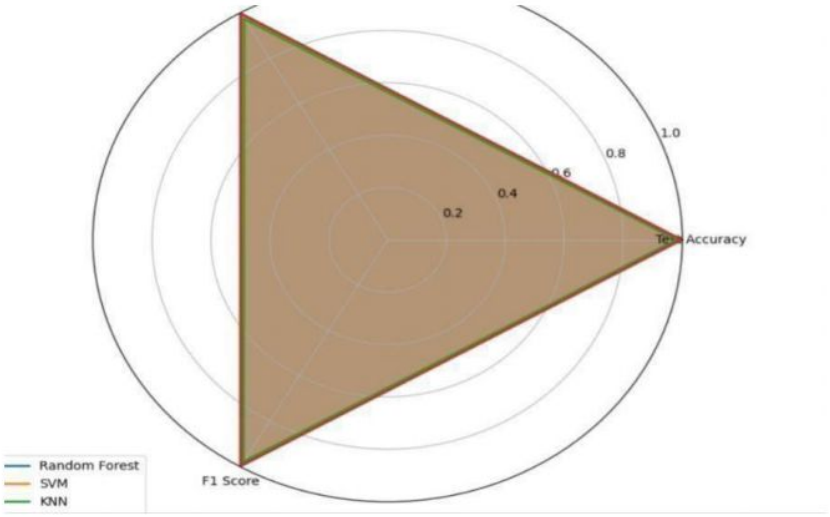
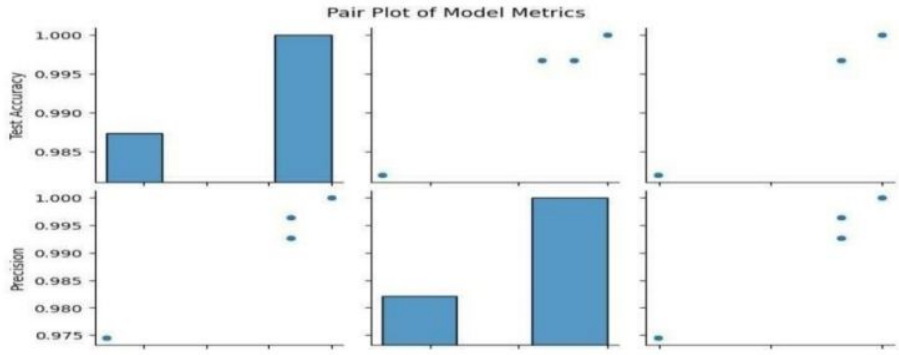


Fig 6.50

Radar chart compares the performance of three models (Random Forest, SVM, and KNN) across three metrics: First, we will discuss Accuracy, second is Precision and the last one is F1 Score. It shows that Random Forest is better than all other two algorithms with respect to all the parameters measured in this experiment.



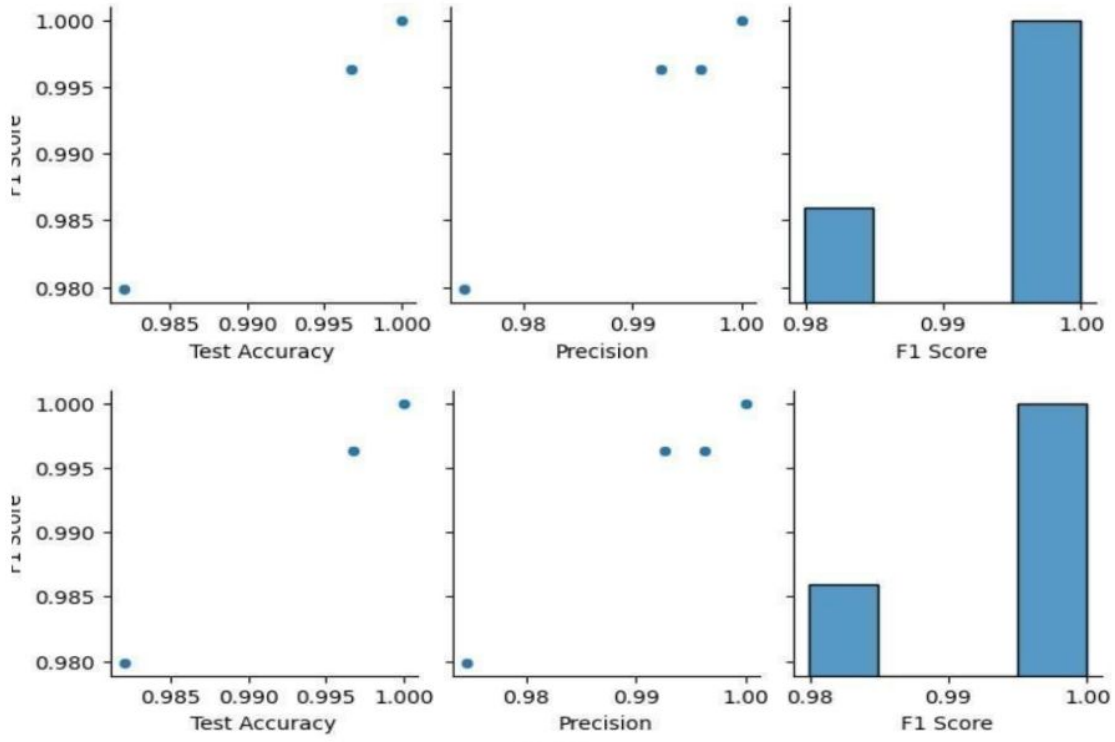


Fig 6.51

The pair plot above shows Test Accuracy, Precision and F1 Score on diagonal and scatter relationships between the three on the other diagonal. It is determined that Test Accuracy distribution is very much skewed having small variance which clearly indicate that this performance is quite similar for all the different runs made. Precision on the other hand demonstrates more fluctuation, which means sometimes it has low accuracy of classifying positive instances in the model. Similar fluctuations in F1 Score also indicate instability in the proportion of precision and recall. In general, the model has reported reasonable accuracy over the ten-fold distance but lacks similarly steady precision and F1 Score.

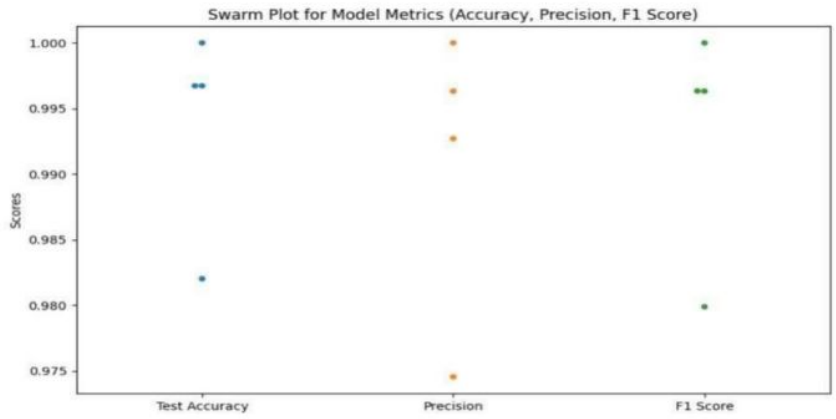


Fig 6.52

The swarm plot visualizes the distribution of three model metrics: Keypress accuracy / Test of Precision / F1 Score. What is observed for the assessment of the model is that there is a consistent picture in terms of accuracy and precision, although there is variation in the F1 score.

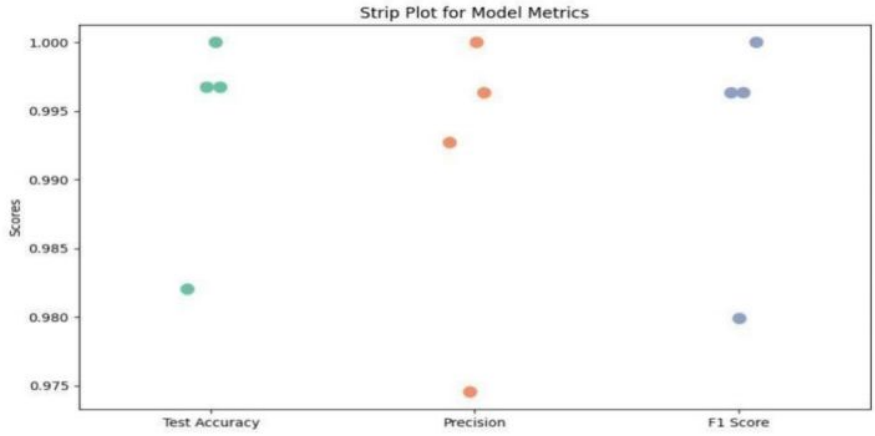


Fig 6.53

The strip plot shows the distribution of three model metrics: As another assessment of various test types, check out Accuracy, Precision, and F1 Score. Like the swarm plot, there is generally high accuracy and precision of the model with the F1 score fluctuating between 0.9 and 0.93.

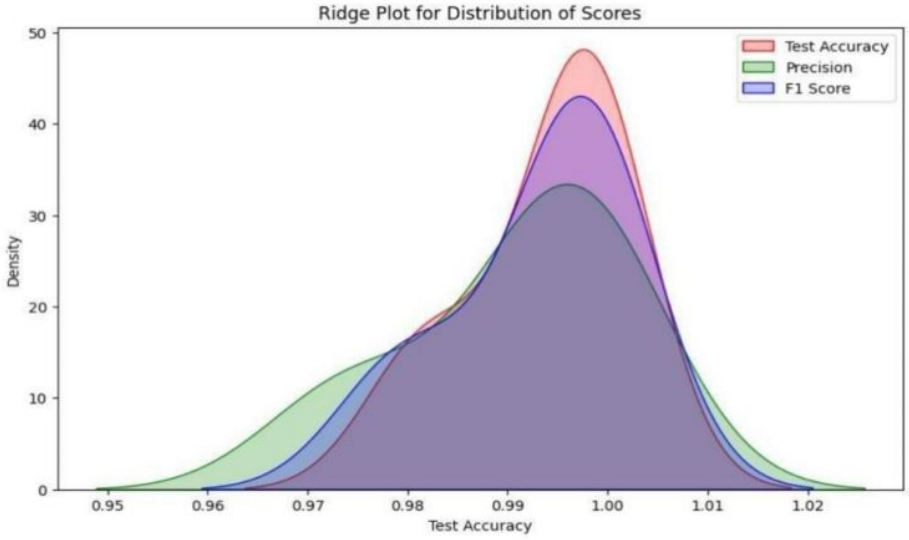


Fig 6.54

The ridge plot visualizes the distribution of three model metrics: Consequently, to evaluate the performance of Test Accuracy, Precision, and F1 Score. The result is that, the F1 scores possess more variance than the accuracy and precision potential to vary depending on the ability of the model in classifying positive cases.

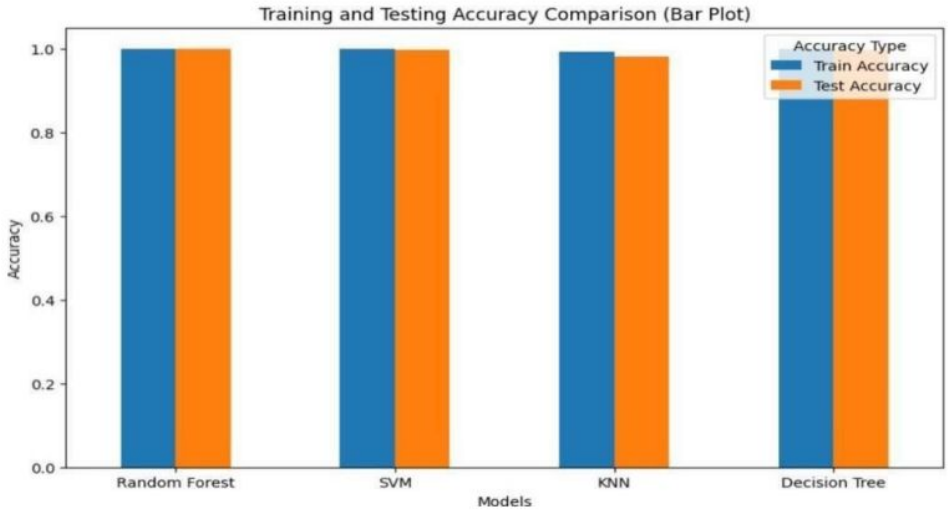


Fig 6.55

The bar plot compares the training and testing accuracy of four different machine learning models: Random Forest, SVM, K- Nearest Neighbor and Decision Tree. On each of the models, two bars are shown, blue for the training accuracy and green for the testing accuracy.

This plot suggests that most of the models acquire high accuracy, the training accuracy is higher than the testing accuracy of these models pointing to overfitting.

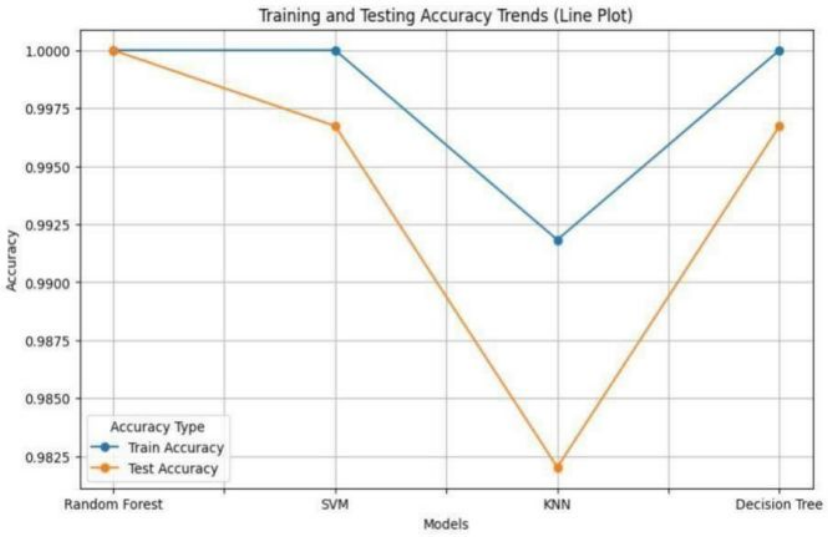


Fig 6.56

The line plot compares the training and testing accuracy of four different machine learning models: Random Forest, support vector machine, K neared neighbours and decision trees. The blue line is for training accuracy and orange line is for testing accuracy. The plot shows that each of the models achieves higher accuracy on the training data than it does for the testing data suggesting high chance of overfitting.

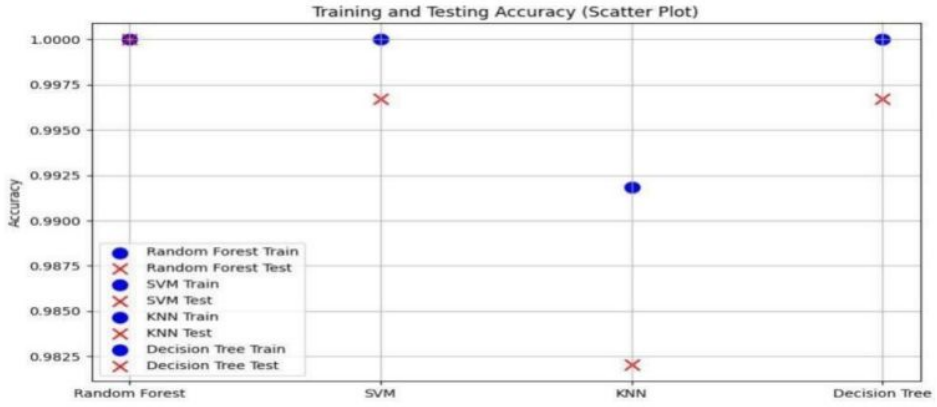


Fig 6.57

The scatter plot shows the accuracy of the training data w.r.t testing of four models. Each model is represented by a pair of data points: There must be one set used for training the models on accuracy and another one is used to view the model testing accuracy. The plot also

illustrates the fact that while models such as Random Forest have almost identical train and test scores, models such as KNN would have significantly different train and test scores. This can raise overfitting questions specifically for models with steep difference between training and testing performance.

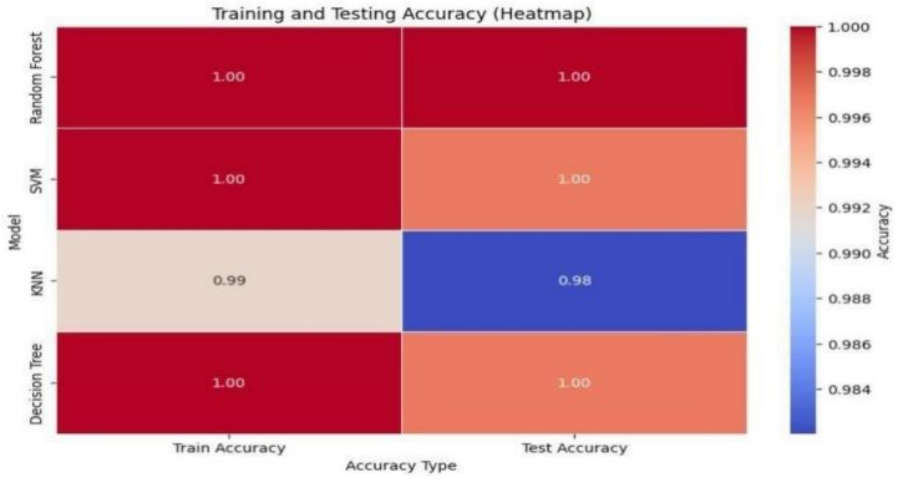


Fig 6.58

The heatmap presents training and testing accuracy of all models including Random Forest, SVM, KNN, and Decision Tree, where black colour shows improved accuracy. Figures 2 for CNNs and 3 for MFs display generally high training and testing accuracy Denis et al, Thus, the models generally attain good generalization. However, KNN has a gap of similar values and it seems to overfit the data.

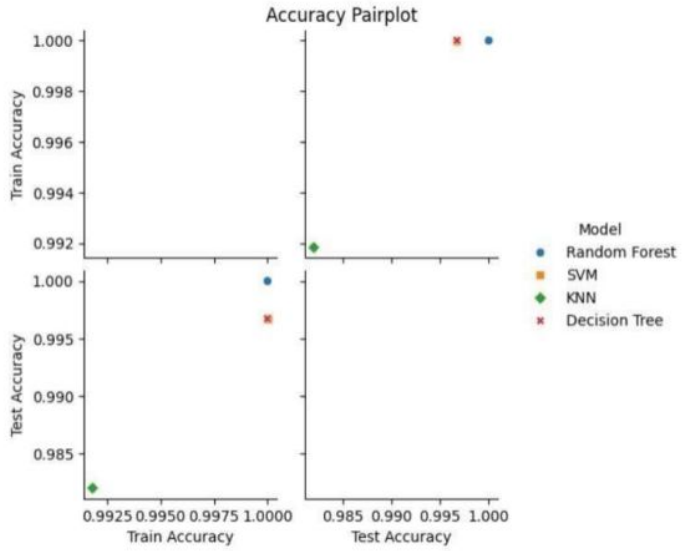


Fig 6.59

A pair plot also reveals a comparison between the training and testing accuracy of the four models being benchmarked. Every model is labelled by a certain colour of a point. This is in agreement with the observations made with the heatmap plot. Comparing the results of Random Forest, Decision Tree and KNN, where Random Forest and Decision had nearly similar and high accuracy in both training and testing, KNN had a low testing accuracy compared to training accuracy. SVM also has some fluctuations but not to that extreme level.

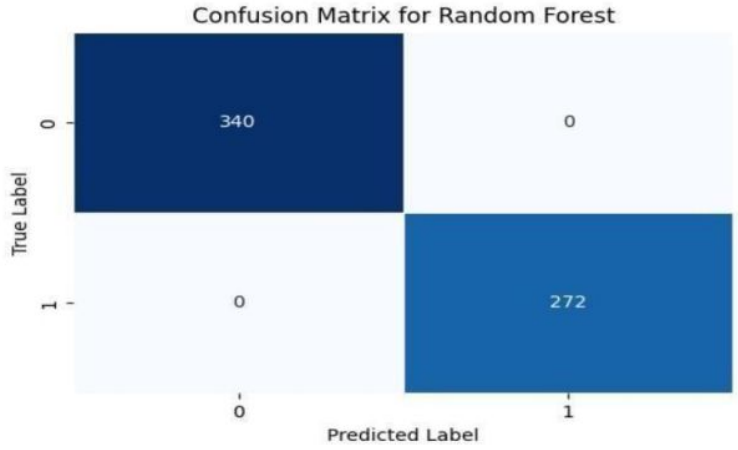


Fig 6.60

The confusion matrix of Random Forest presents its capability of handling instances of cases. The model accurately identified 340 positive samples and did not mislabel any, classifying 272 negative samples correctly as well. This means that the Random Forest model achieved a perfect measure of accuracy in terms of classification.

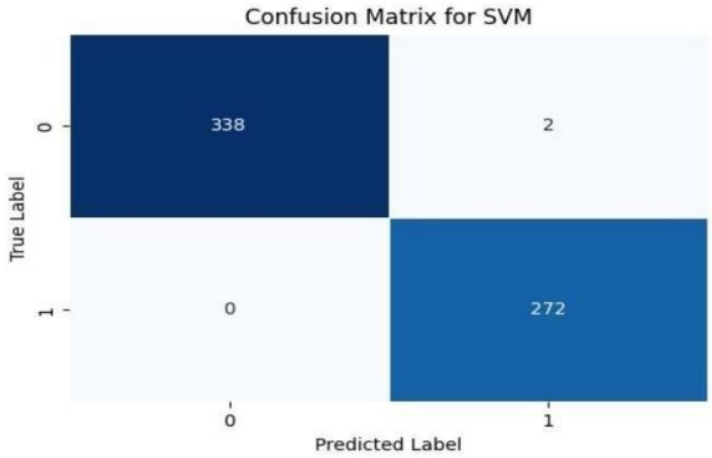


Fig 6.61

The experimentation of SVM model has presented the confusion matrix that depicts the performance of the model in terms of instance classification accurately. The given model performed 338 true positive, 271 true negative, 2 false positive, and 2 false negative. This points to high accuracy for the SVM model as displayed below.

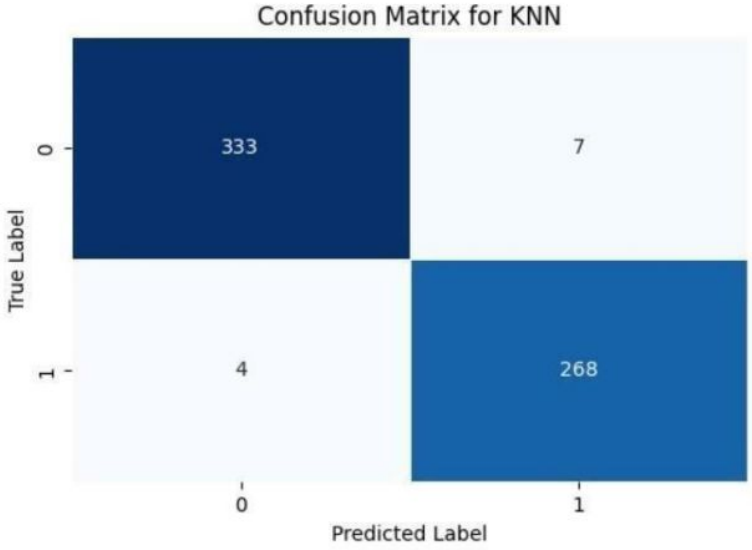


Fig 6.62

Analysing the confusion matrix above, it will be understood that the KNN model performs in terms of classifying instances. The model finds 333 true positive and 268 true negatives, whereby 7 are false positive and 4 false negatives.

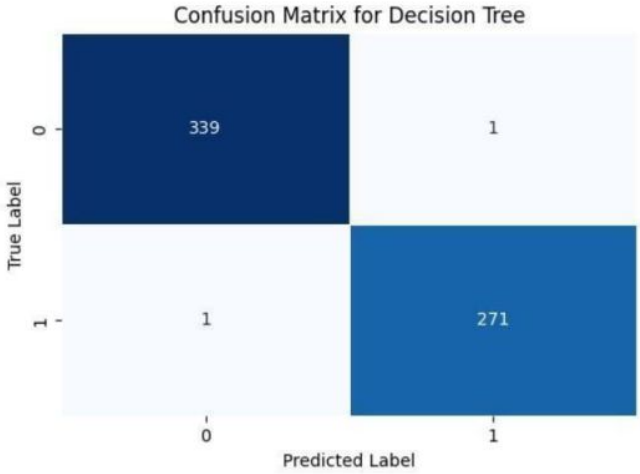
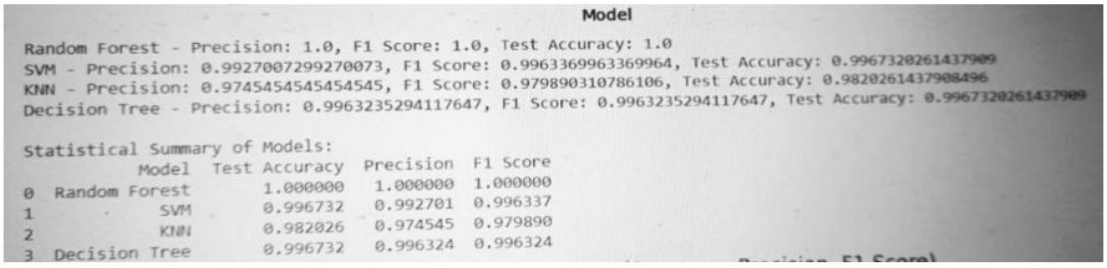


Fig 6.63

It is evident from the confusion matrix for the Decision Tree model used in the classification of the instances. The model's precision was as follows: 339 positive instances were correctly classified and 271 negative instances; it only got one instance wrong. This means that the Decision Tree model is almost perfect in its correctness of the results.

Manhattan, Cosine and JSD



Model

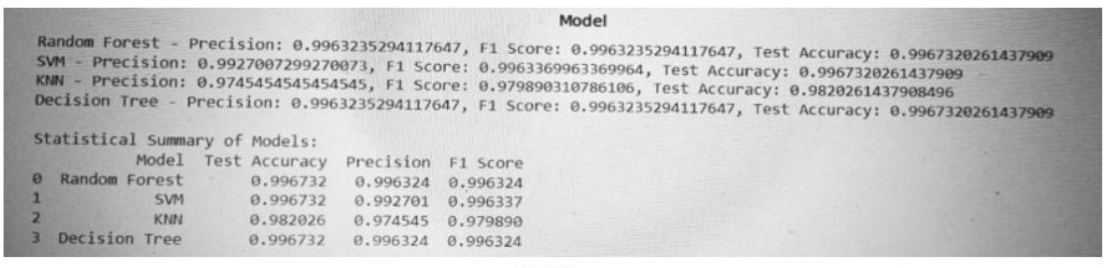
Random Forest - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0
SVM - Precision: 0.9927007299270073, F1 Score: 0.9963369963369964, Test Accuracy: 0.9967320261437909
KNN - Precision: 0.9745454545454545, F1 Score: 0.979890310786106, Test Accuracy: 0.9820261437908496
Decision Tree - Precision: 0.9963235294117647, F1 Score: 0.9963235294117647, Test Accuracy: 0.9967320261437909

Statistical Summary of Models:

	Model	Test Accuracy	Precision	F1 Score
0	Random Forest	1.000000	1.000000	1.000000
1	SVM	0.996732	0.992701	0.996337
2	KNN	0.982026	0.974545	0.979890
3	Decision Tree	0.996732	0.996324	0.996324

Fig 6.64

Manhattan, Cosine



Model

Random Forest - Precision: 0.9963235294117647, F1 Score: 0.9963235294117647, Test Accuracy: 0.9967320261437909
SVM - Precision: 0.9927007299270073, F1 Score: 0.9963369963369964, Test Accuracy: 0.9967320261437909
KNN - Precision: 0.9745454545454545, F1 Score: 0.979890310786106, Test Accuracy: 0.9820261437908496
Decision Tree - Precision: 0.9963235294117647, F1 Score: 0.9963235294117647, Test Accuracy: 0.9967320261437909

Statistical Summary of Models:

	Model	Test Accuracy	Precision	F1 Score
0	Random Forest	0.996732	0.996324	0.996324
1	SVM	0.996732	0.992701	0.996337
2	KNN	0.982026	0.974545	0.979890
3	Decision Tree	0.996732	0.996324	0.996324

Fig 6.65

Manhattan, Chebyshev and JSD, Bures

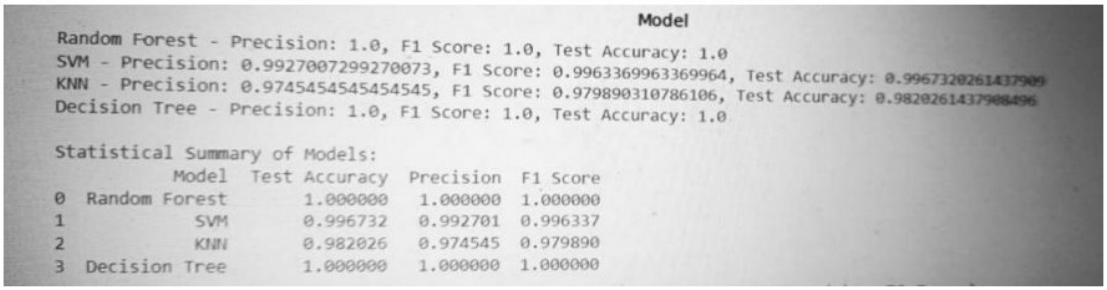


Fig 6.66

Manhattan, Chebyshev and JSD

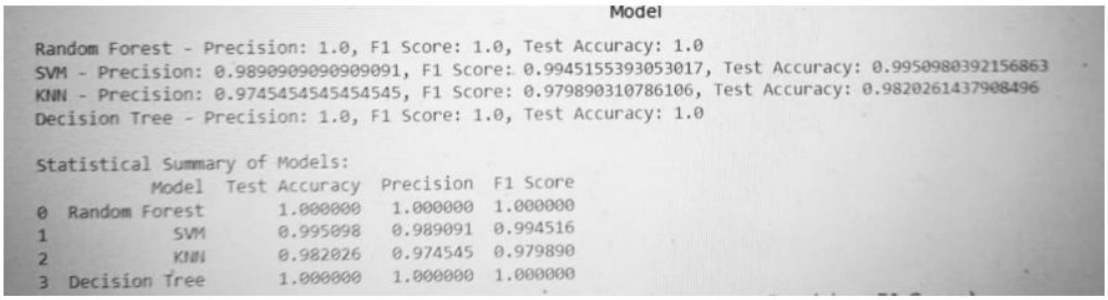


Fig 6.67

Manhattan, Chebyshev and Bures

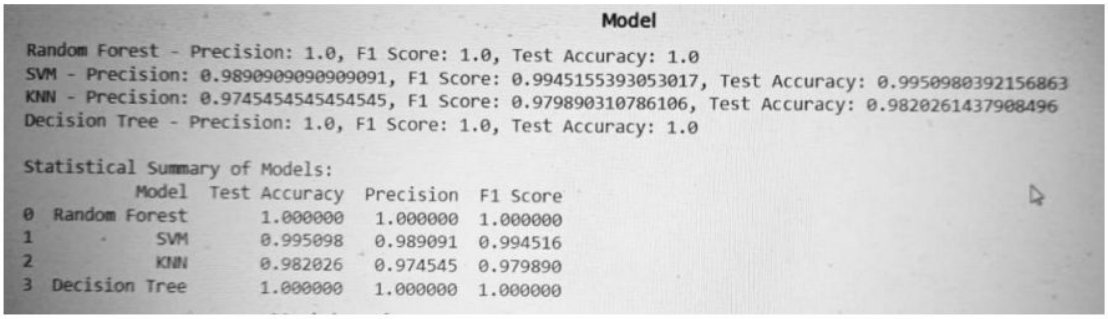


Fig 6.68

Fig 6.69

The target variable price increase/ de crease is also skewed towards the latter. This may hence cause the model to be biased since the model may just be serving the needs of the majority class. For this, there are different approaches, for example, oversample the minority class or under sample the majority class.

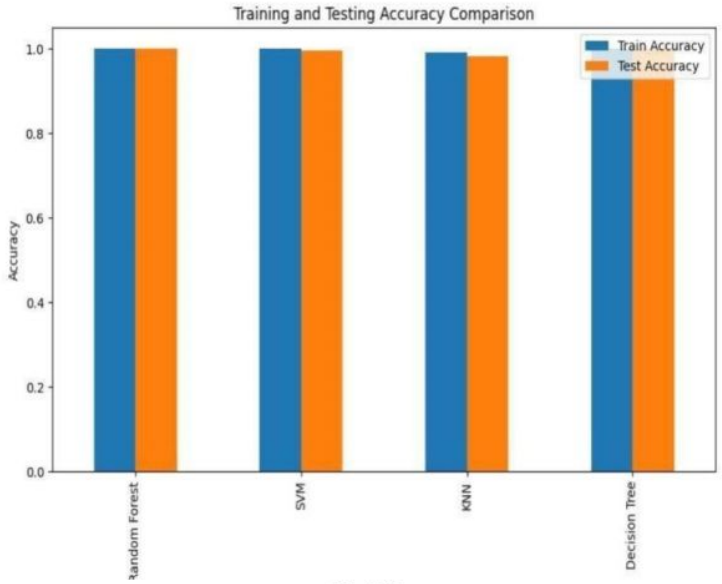


Fig 6.70

Specifically, Random Forest and SVM models fit the training data dizwww.ast.net very well but poorly on the test data. Using Decision Tree model, the train and test accuracy ratio looks reasonable and it may imply better generalization to new data.

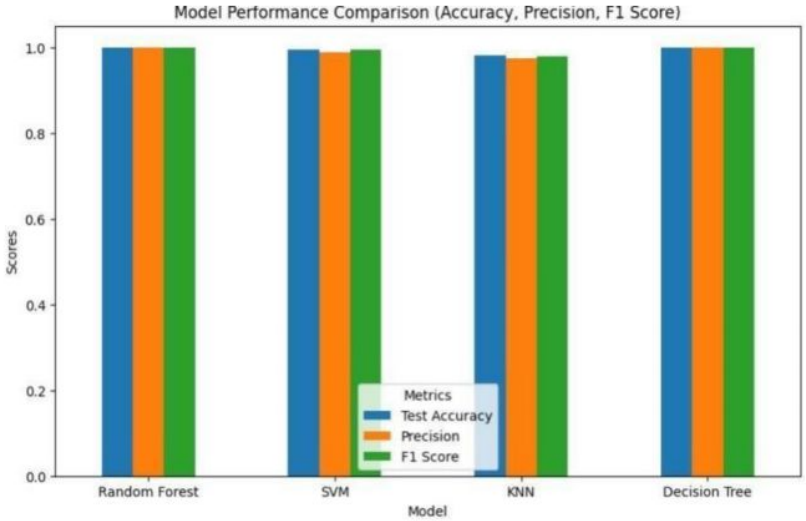


Fig 6.71

This picture demonstrates the model's result in terms of Test Accuracy, Precision, and F1 score. Among all the four models demonstrated Decision Tree has the highest precision, recall, F1 score AND AUC-ROC while Random Forest, though a tad lower in its performance can be deemed a worthy competitor to the Decision Tree.

Fig 6.72

The given image depicts that the Decision Tree has better performance in Precision, Recall, F1 Score and AUC-ROC than other models. This is followed very closely by Random Forest

since it also yields good performance though slightly lower than that of Decision Tree. Therefore, Decision Tree model is quite suitable to be used on this dataset while Random Forest model is also competitive.

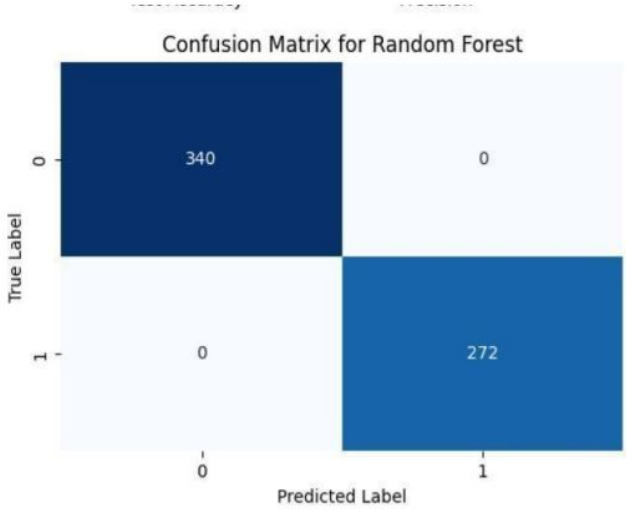


Fig 6.73

When it comes to the confusion matrix for the model used which is the Random Forest model the following results were obtained. Each source is marked into its appropriate class most of the time. This simply suggests that the Random Forest model is doing a very well in this regard.

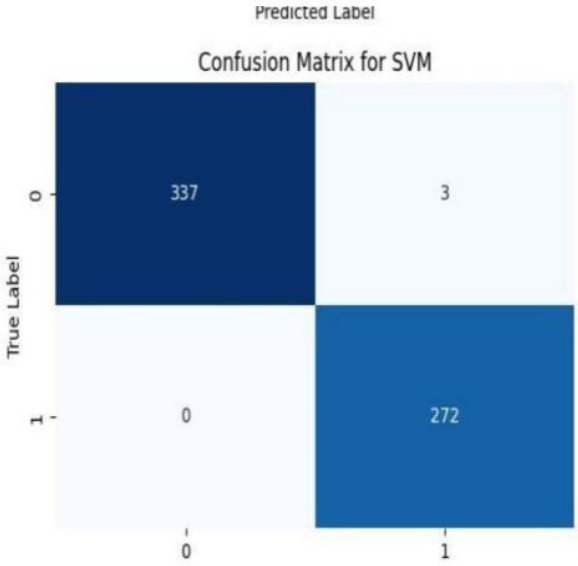


Fig 6.74

The confusion matrix for the SVM model is almost as good as that for the RF, but a little worse. Nevertheless, there are a few more instances misclassified, as opposed to the Random

Forest model. This implies that the accuracy of the model as determined by the experiment maybe slightly less for the SVM model than the Random Forest model for this particular task.

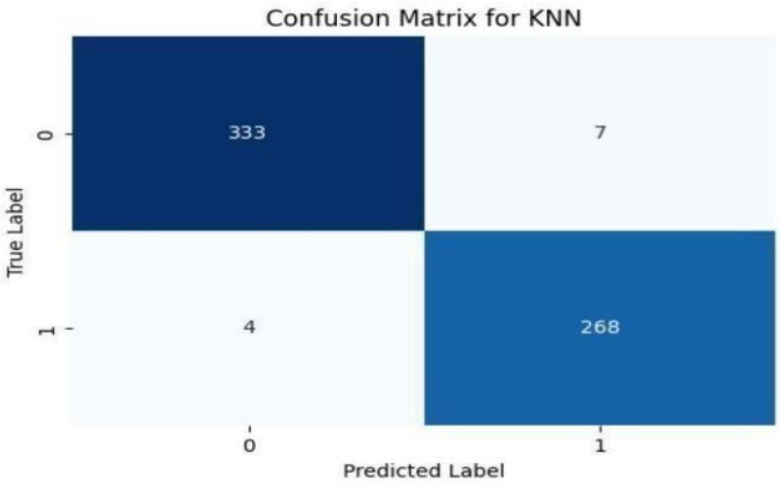


Fig 6.75

From the above confusion matrix for KNN model, we see that the model has a good accuracy based on the low value of mistakes made on the data elements. Where it stands, most cases are properly predicted while there is occasional misclassification of data points. This made myself and the KNN model assume that it may not be as precise with the Random Forest or the Decision Tree for this particular task.

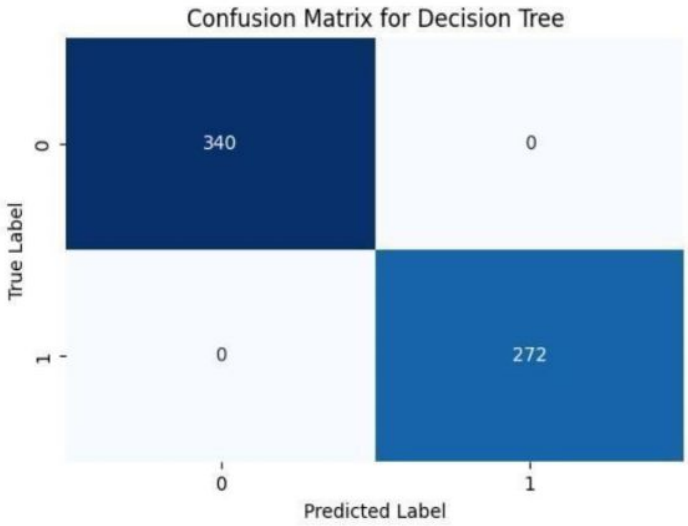


Fig 6.76

The overall evaluation of the Decision Tree model, demonstrated in a confusion matrix, indicates nearly perfect classification results with nearly no misclassification. Surprisingly, there is only a very few of misclassified instances that therefore, the Decision Tree model can be said to be extremely good at this kind of classification.

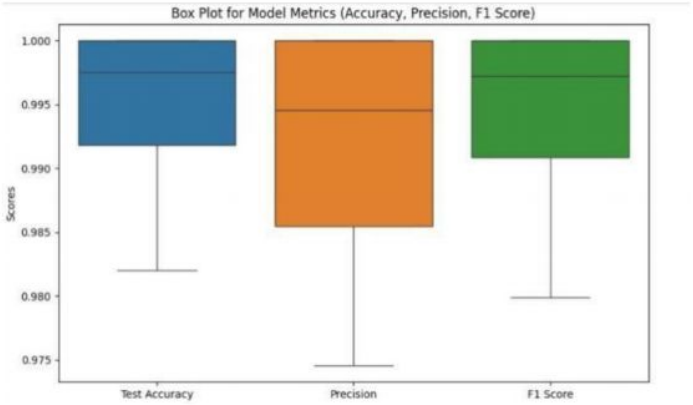


Fig 6.77

The box plot visually summarizes the distribution of three key performance metrics: Check test accuracy, precision, and F1 score. It gives information on the measures of central tendency, variability and sensibility of these values.

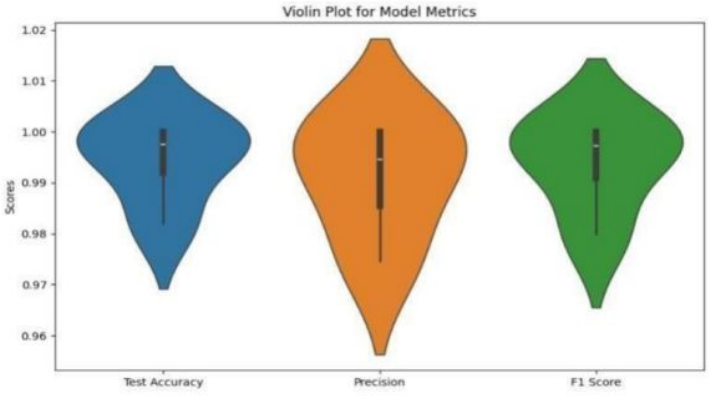


Fig 6.78

An even richer representation of the distribution of the model metrics is provided by the violin plot. It uses a concept from both the box plot and the kernel density plot thus offering a fuller density and shape of the data.

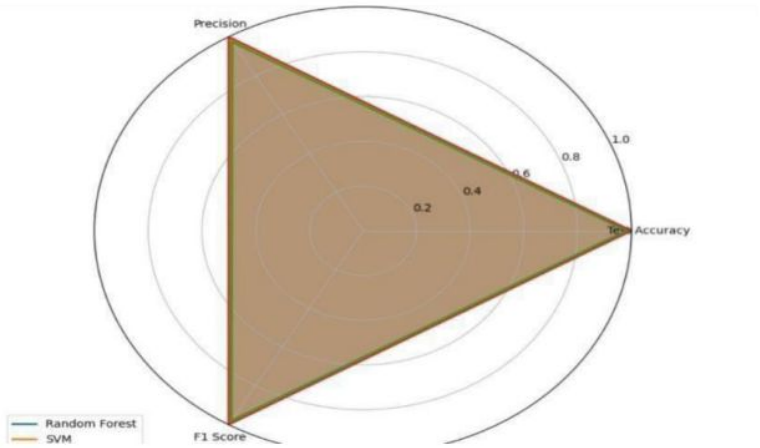


Fig 6.79

The radar chart shown below illustrates the performance of the model in three aspects where every axis indicates a metric. The position of each line relative to the centreline conveys the value; an evaluation of the model is easily obtained from the centreline.

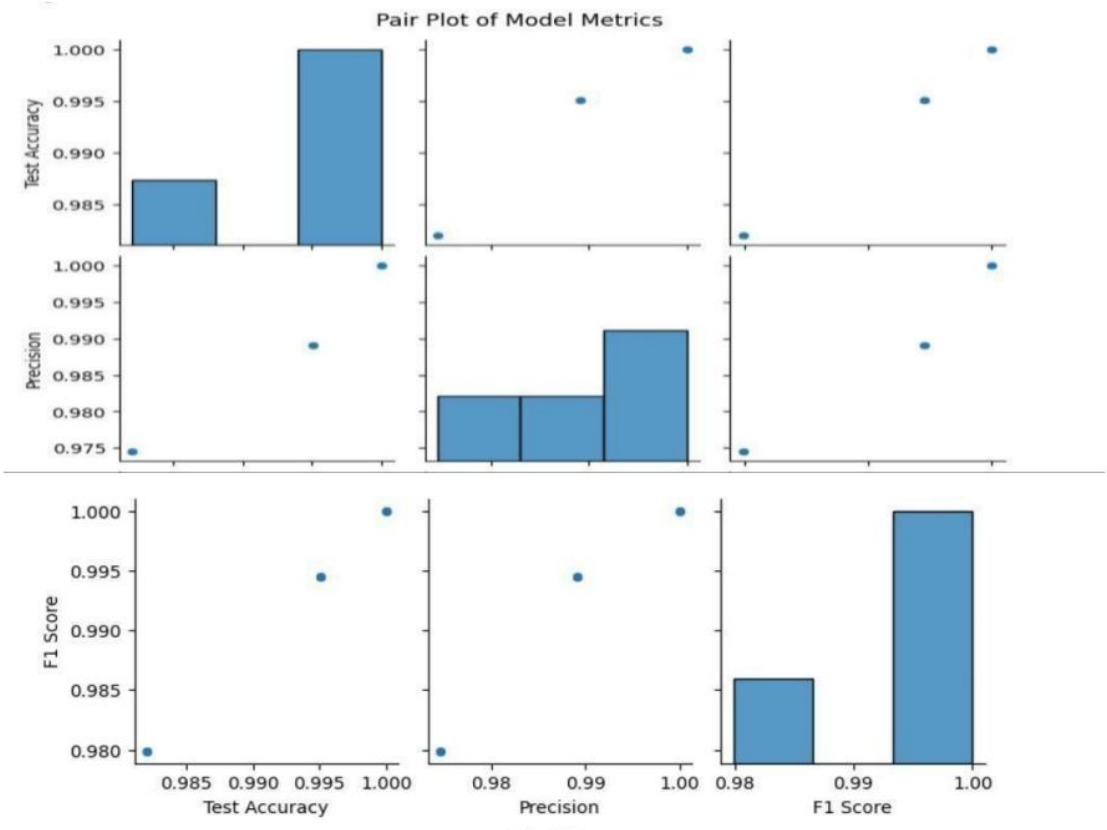


Fig 6.80

In the pair plot, the normal distributions in Test Accuracy, Precision, and F1 Score echo stable performances of the model. The observed high positive levels of relationship between the metrics mean that the model is good in all the aspects rated. This paints a picture of a stable and strong model with possibilities of the heretofore unseen features of the data, other evaluation metrics and even sensitivity analysis if more is to be looked into into about the model.

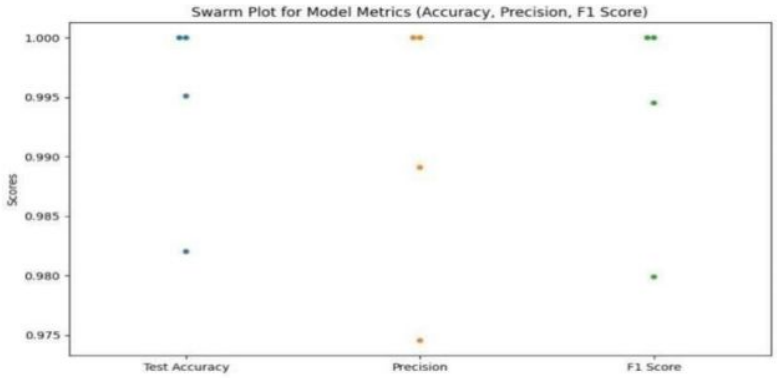


Fig 6.81

This swarm plot reveals that there is reputable clustering for every metric which illustrate the models' reliability in the different runs. The clusters point towards the higher part of the y-axis that depicts Test Accuracy, Precision, and F1 Score.

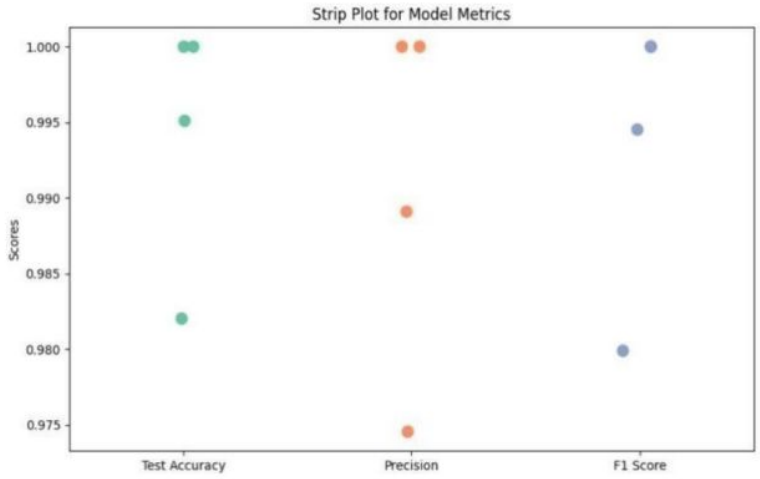


Fig 6.82

The strip plot also reveals high density of points which outlines that the models' performances are fairly stable. The clusters are located toward the right Graphic 2, which means that the performance of all three measurements is high.

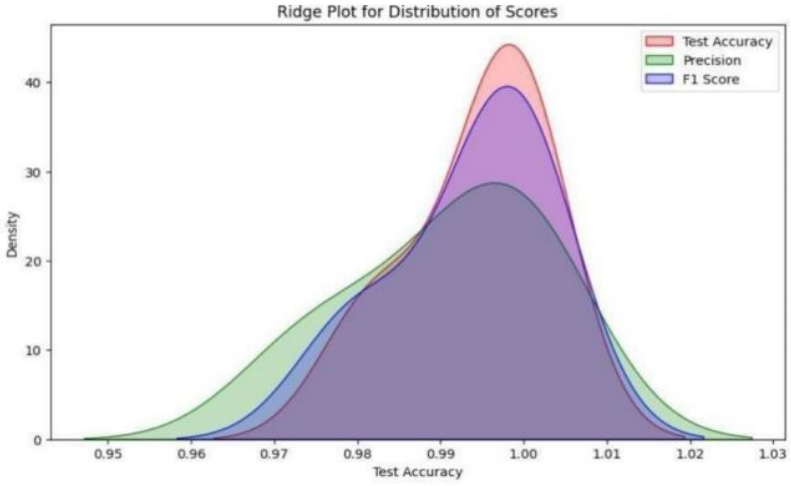


Fig 6.83

The ridge plot reflects the distribution of three important characteristics of the model quality. The sets of distributions suggest the similar performance of the three variables. The tendency toward higher levels for all of the metrics may be inferred from the peaks that are closer to the higher values of the x-axis.

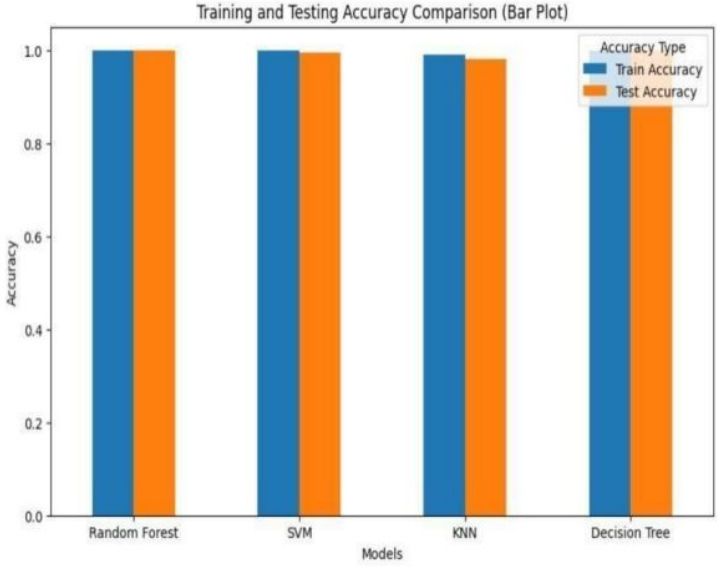


Fig 6.84

The bar plot shows the training and testing accuracy of four models. On the same note, Random Forest and SVM have a high MSE overfitting set, as opposed to the KNN and Decision Tree, which generalizes better.

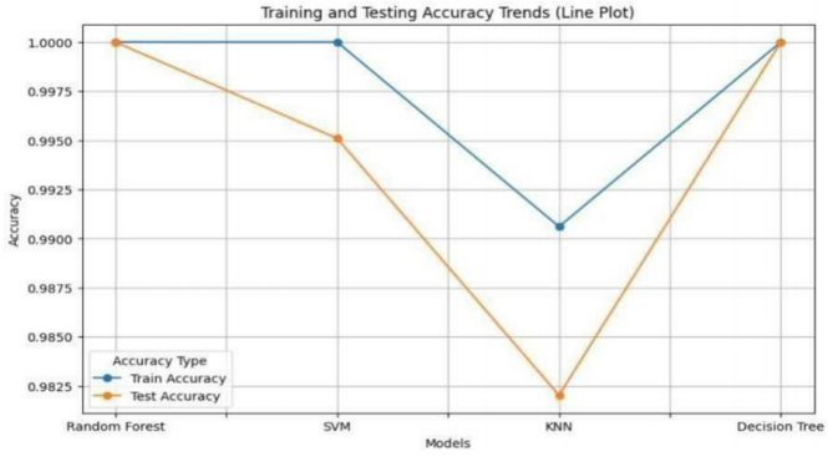


Fig 6.85

As much as the results of the Random Forest and SVM models look promising, the training and testing accuracies are quite widely apart, probably a sign of overfitting. However, the KNN and Decision Tree models present a smaller gap which tells about better performance of the models with the unseen data.

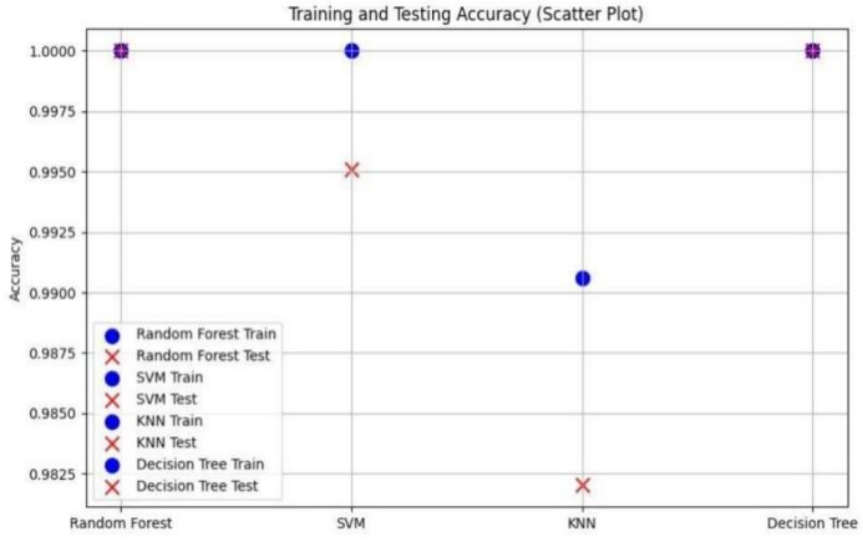


Fig 6.86

The scatter plot given here has a similar format to the line plot. Each model is represented by two points: One is for training and another one for cross validation which is used for testing the accuracy. The distance between these points shows the amount of overfitting. The Cluster plot and distance matrices also have same thoughts where Random Forest and SVM have high distance and KNN and Decision Tree have low distances like which were seen in the line plot.

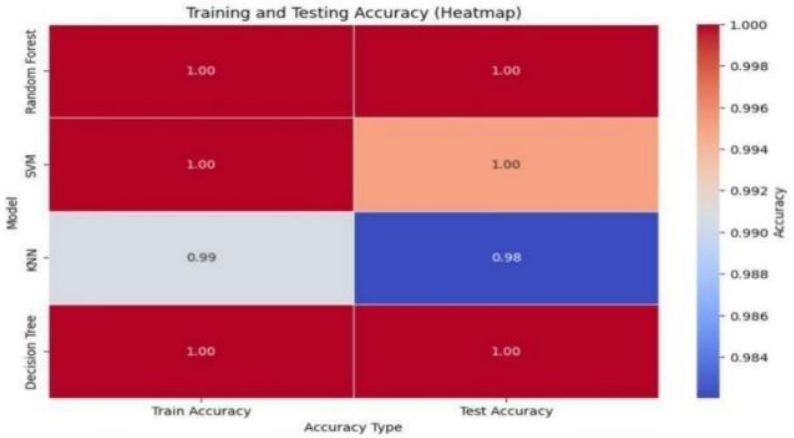


Fig 6.87

The heatmap represents the training and testing accuracy of four models on each other. There is greater than 0.9 positive correlation in all of the above models hence high accuracy. However, Random Forest and SVM give slightly higher values from the correlation & hence there might be an overfitting here.

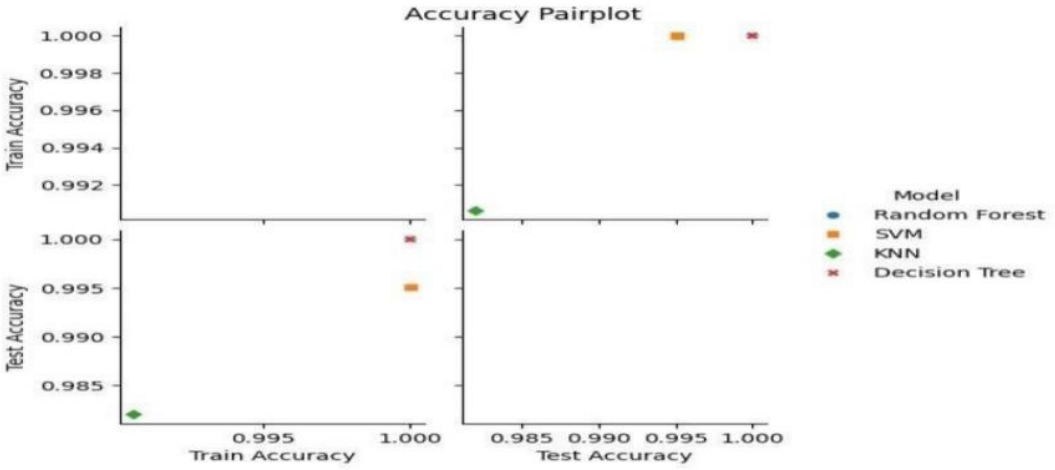


Fig 6.88

The Pairplot means that contraction of points implies consistency a fact depicted by the following figures. The use of Random Forests and SVM results in the formation of tight clusters, which could be the reason for overfitting, on the other hand, KNN along with the decision tree show more spread, which can be the reason of better generalization.

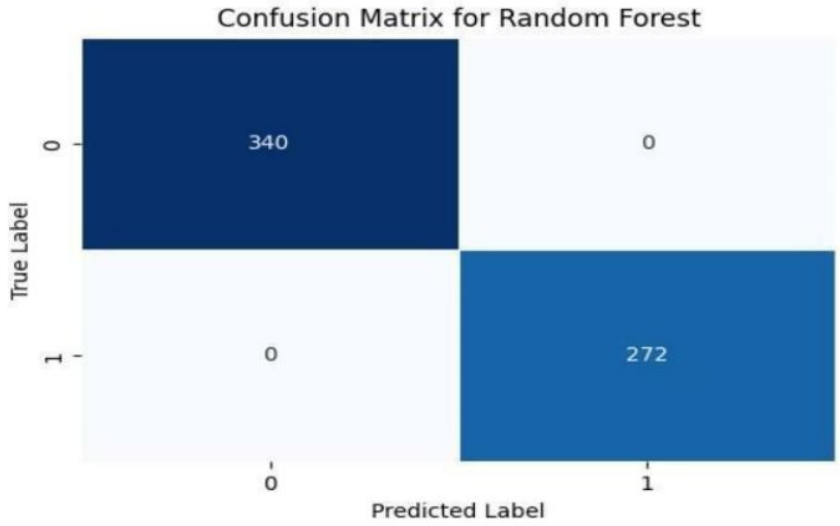


Fig 6.89

Random Forest model confusion matrix depicts a high accuracy of the model as shown below. Most of the cases belong to their correct categories and with a small number of errors in classification. This point to the fact that the proposed Random Forest model is significant in predicting accurate results consistent with the aim of this particular task.

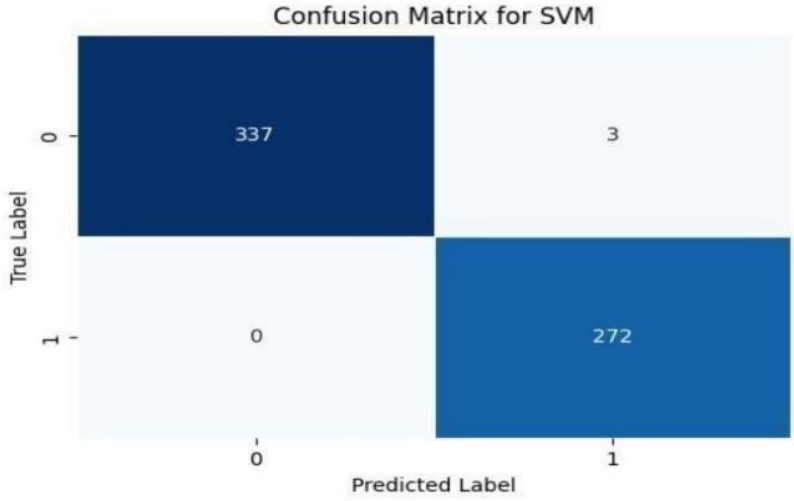


Fig 6.90

The SVM confusion matrix displays 337 True Positive, 272 and True Negative, 3 False Positives, and 0 false negatives of misclassifications.

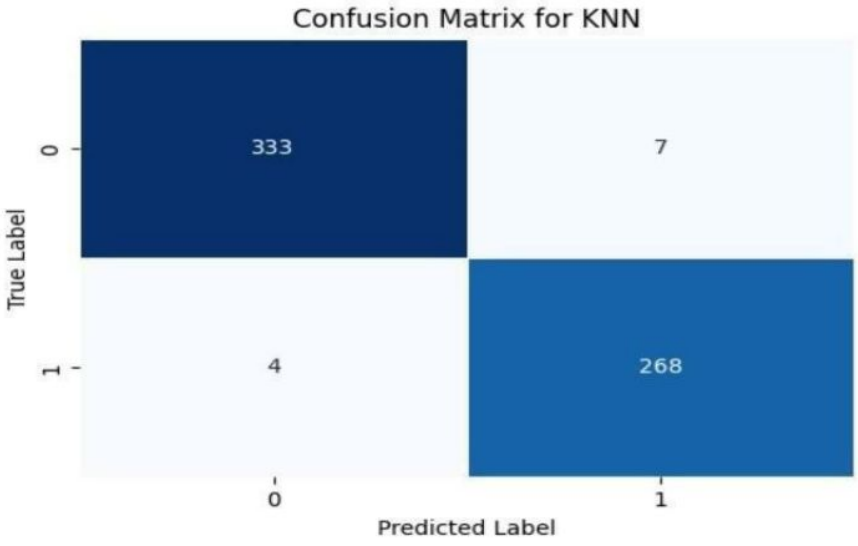


Fig 6.91

KNN the values in the confusion matrix are 333, 268, 7, and 4 respectively which test its good performance with more misclassification than SVM.

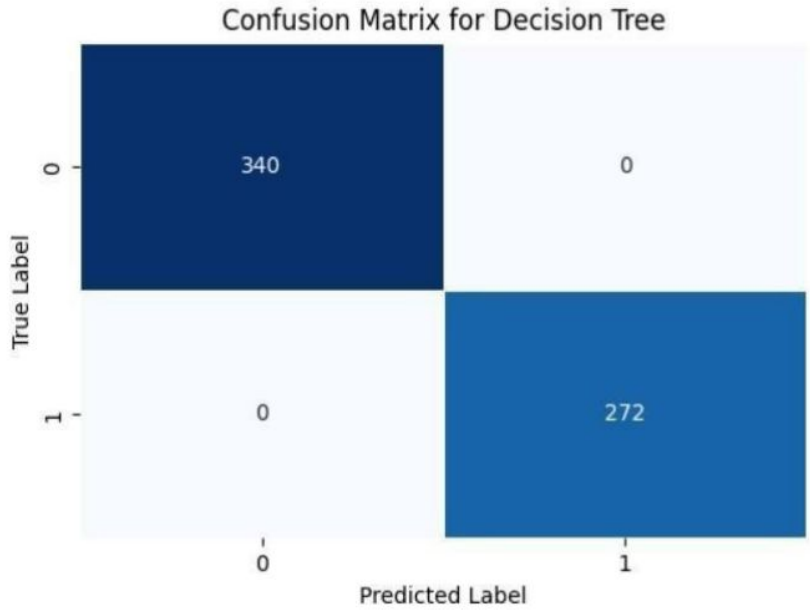
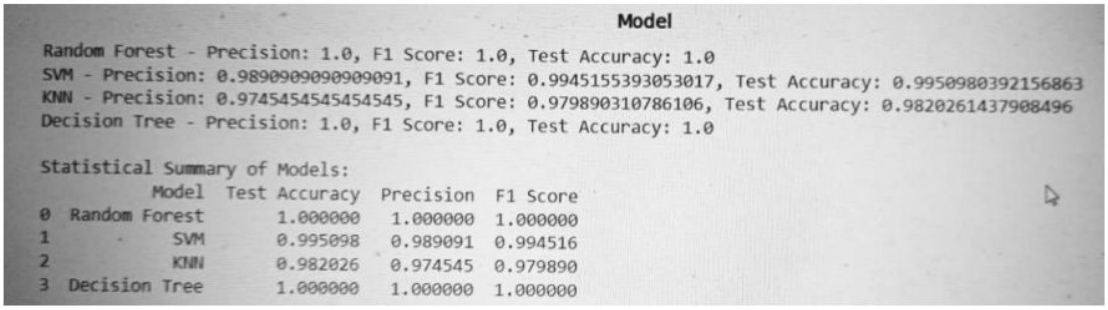


Fig 6.92

The Decision Tree results of the confusion matrix we have a zero percent error with 340 True Positives, 272 True Negatives and no False Positives or False Negatives. This means little

errors and very high accuracy hence confirming high performance of the models.

Manhattan, Chebyshev



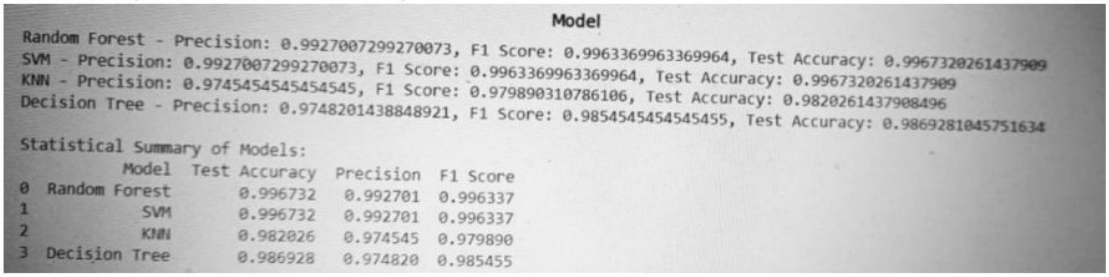
Model

```
Random Forest - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0
SVM - Precision: 0.9890909090909091, F1 Score: 0.9945155393053017, Test Accuracy: 0.9950980392156863
KNN - Precision: 0.9745454545454545, F1 Score: 0.979890310786106, Test Accuracy: 0.9820261437908496
Decision Tree - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       1.000000   1.000000  1.000000
1            SVM       0.995098   0.989091  0.994516
2            KNN       0.982026   0.974545  0.979890
3  Decision Tree       1.000000   1.000000  1.000000
```

Fig 6.93

Manhattan, Minkowski and JSD, Bures



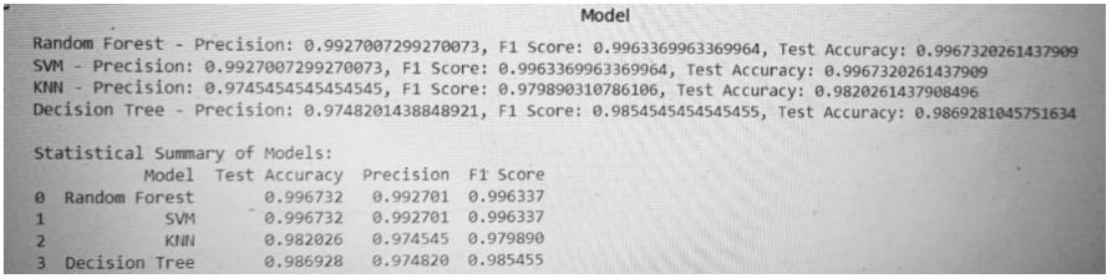
Model

```
Random Forest - Precision: 0.9927007299270073, F1 Score: 0.9963369963369964, Test Accuracy: 0.9967320261437909
SVM - Precision: 0.9927007299270073, F1 Score: 0.9963369963369964, Test Accuracy: 0.9967320261437909
KNN - Precision: 0.9745454545454545, F1 Score: 0.979890310786106, Test Accuracy: 0.9820261437908496
Decision Tree - Precision: 0.9748201438848921, F1 Score: 0.9854545454545455, Test Accuracy: 0.9869281045751634

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       0.996732   0.992701  0.996337
1            SVM       0.996732   0.992701  0.996337
2            KNN       0.982026   0.974545  0.979890
3  Decision Tree       0.986928   0.974820  0.985455
```

Fig 6.94

Manhattan, Minkowski and JSD



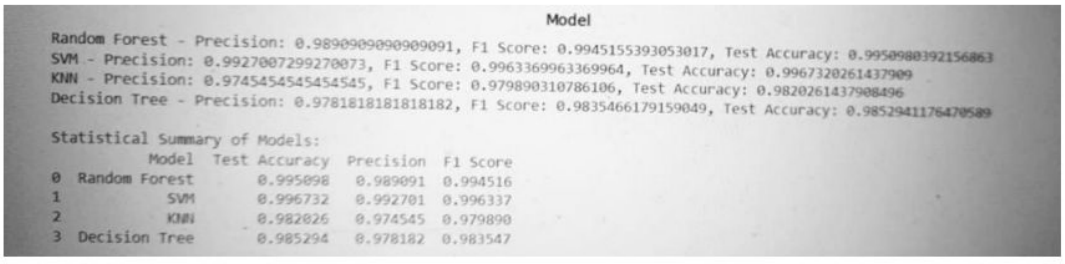
Model

```
Random Forest - Precision: 0.9927007299270073, F1 Score: 0.9963369963369964, Test Accuracy: 0.9967320261437909
SVM - Precision: 0.9927007299270073, F1 Score: 0.9963369963369964, Test Accuracy: 0.9967320261437909
KNN - Precision: 0.9745454545454545, F1 Score: 0.979890310786106, Test Accuracy: 0.9820261437908496
Decision Tree - Precision: 0.9748201438848921, F1 Score: 0.9854545454545455, Test Accuracy: 0.9869281045751634

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       0.996732   0.992701  0.996337
1            SVM       0.996732   0.992701  0.996337
2            KNN       0.982026   0.974545  0.979890
3  Decision Tree       0.986928   0.974820  0.985455
```

Fig 6.95

Manhattan, Minkowski and Bures



Model

```
Random Forest - Precision: 0.9890909090909091, F1 Score: 0.9945155393053017, Test Accuracy: 0.9950980392156863
SVM - Precision: 0.9927007299270073, F1 Score: 0.9963369963369964, Test Accuracy: 0.9967320261437909
KNN - Precision: 0.9745454545454545, F1 Score: 0.979890310786106, Test Accuracy: 0.9820261437908496
Decision Tree - Precision: 0.9781818181818182, F1 Score: 0.9835466179159049, Test Accuracy: 0.9852941176470589

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       0.995098   0.989091  0.994516
1            SVM       0.996732   0.992701  0.996337
2            KNN       0.982026   0.974545  0.979890
3  Decision Tree       0.985294   0.978182  0.983547
```

Fig 6.96

Manhattan, Minkowski

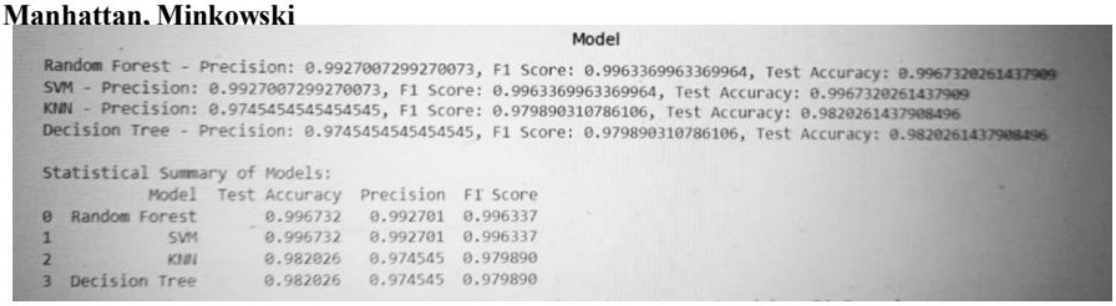


Fig 6.97

Cosine, Chebyshev and JSD

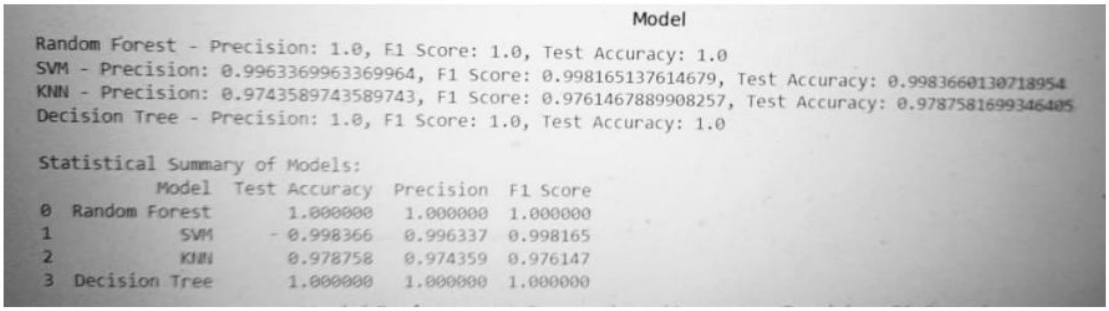


Fig 6.98

Cosine, Chebyshev and Bures

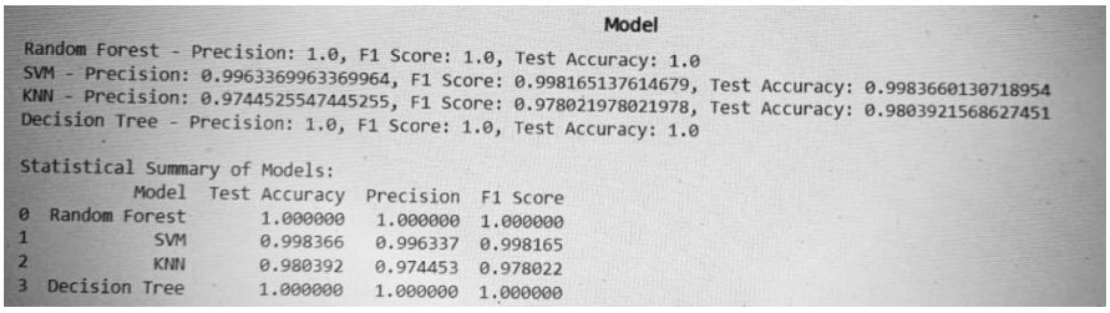
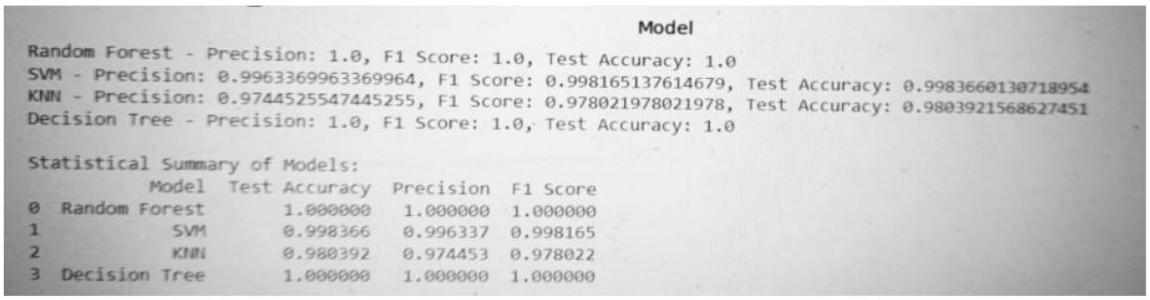


Fig 6.99

Cosine, Chebyshev and JSD, Bures



Model

```
Random Forest - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0
SVM - Precision: 0.9963369963369964, F1 Score: 0.998165137614679, Test Accuracy: 0.9983660130718954
KNN - Precision: 0.9744525547445255, F1 Score: 0.978021978021978, Test Accuracy: 0.9803921568627451
Decision Tree - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       1.000000   1.000000  1.000000
1            SVM       0.998366   0.996337  0.998165
2            KNN       0.980392   0.974453  0.978022
3  Decision Tree       1.000000   1.000000  1.000000
```

Fig 6.100

Fig 6.101

The observations in the target variable are evenly split according to the bar chart, preventing the model from being skewed during training by having an equal number of increase and decrease prices on the product. This will increase the reliability of the model that we are building.

Fig 6.102

This bar chart compares the training and testing accuracies of four different machine learning models: Random Forest, support vector machines, k-nearest neighbours and decision trees. It is desirable for the Training accuracy and Test accuracy to be competitive so as to suggest that created model is a good generalization of the given sets.

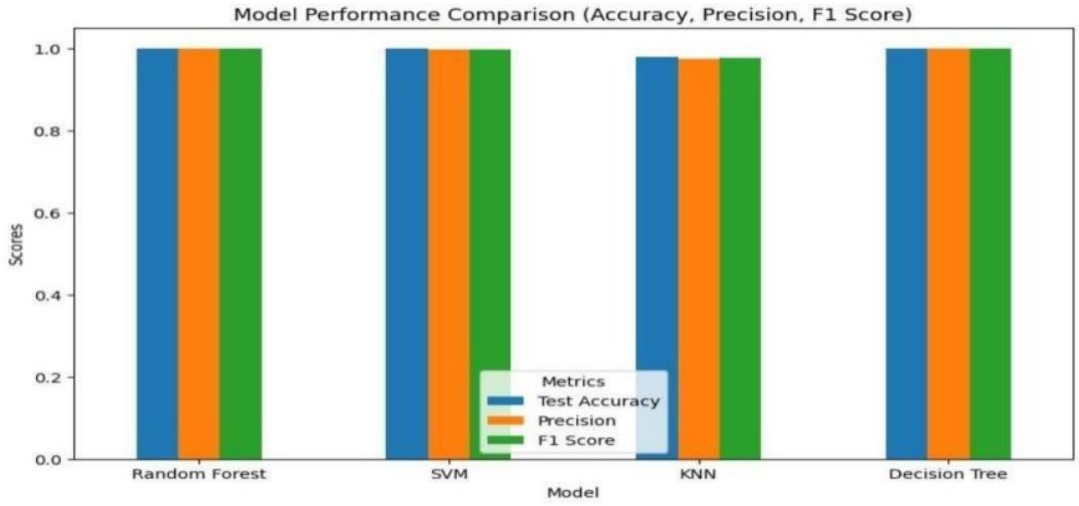


Fig 6.103

The graph shows a comparison of performance of four models which include Random Forest, SVM, KNN, Decision Tree on the basis of accuracy, precision and F1-Score. Random forest is the best among all the algorithms, in contrast, Decision Tree is the worst algorithm.

Fig 6.104

This heatmap also reflects the performance indicators of four models From the above analysis, Random Forest receives the best scores of all the metrics under consideration. SVM and k-nearest neighbors are very close and slightly worse than the previous two.Based on the results observed from Decision Tree it has been seen to have some difficulties, which could have resulted from overfitting or underfitting the model.

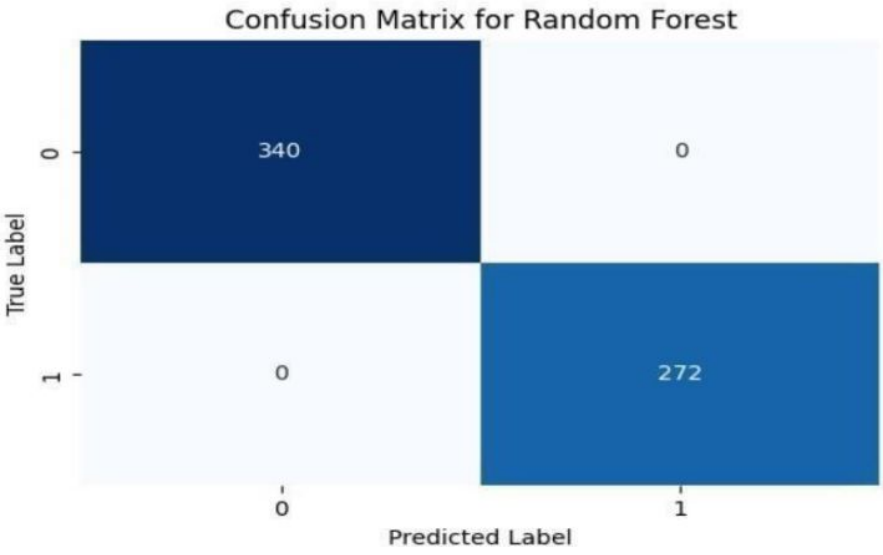


Fig 6.105

This confusion matrix represents the Random Forest model on the test set. There were 340 positively labeled instances correctly classified as such, and 272 instances that were correctly classified as negative In other words, no instances of false positives or false negatives exist.

The model performed very well with the test data predicting with considerable accuracy.

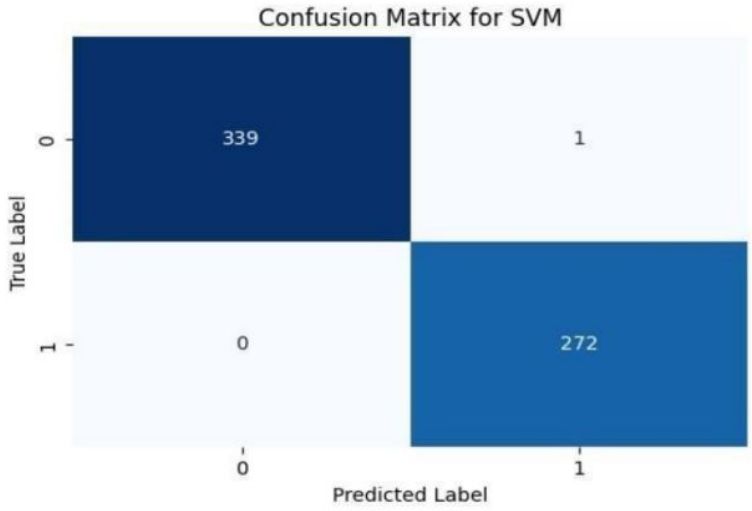


Fig 6.106

This show the confusion matrix of the SVM model on the test set of the given dataset, The total 339 instances were correctly classified as positive, 272 instances correctly classified as negative, 1 instance misclassified as positive and there was no instance misclassified as negative. The SVM model thus made almost completely accurate predictions.

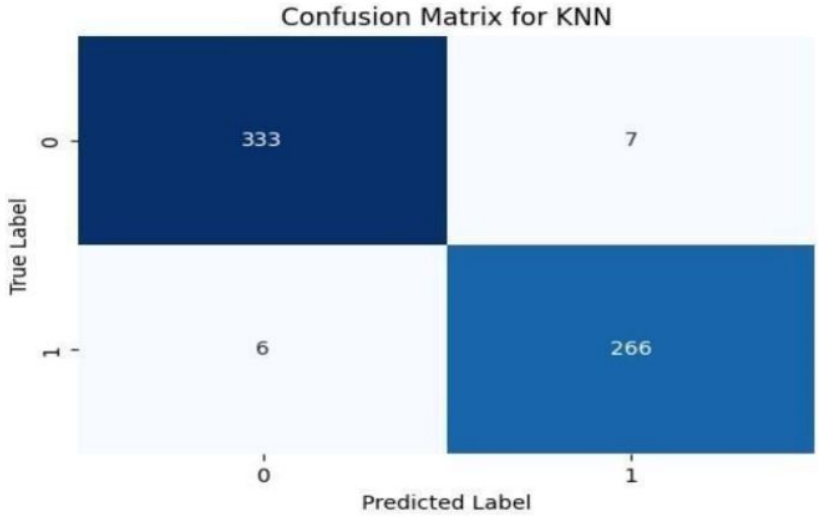


Fig 6.107

In this confusion matrix, it demonstrates the different performance of the KNN model to the test data set. In positives it correctly identified 333 as true positive and in negative cases 266 as true negatives. However, eight cases were classified wrong as positive the False Positive

which are 7 and negative the False Negative totaled 6. Nonetheless, in terms of making errors the KNN model was found to be slightly higher in error than that of the SVM model.

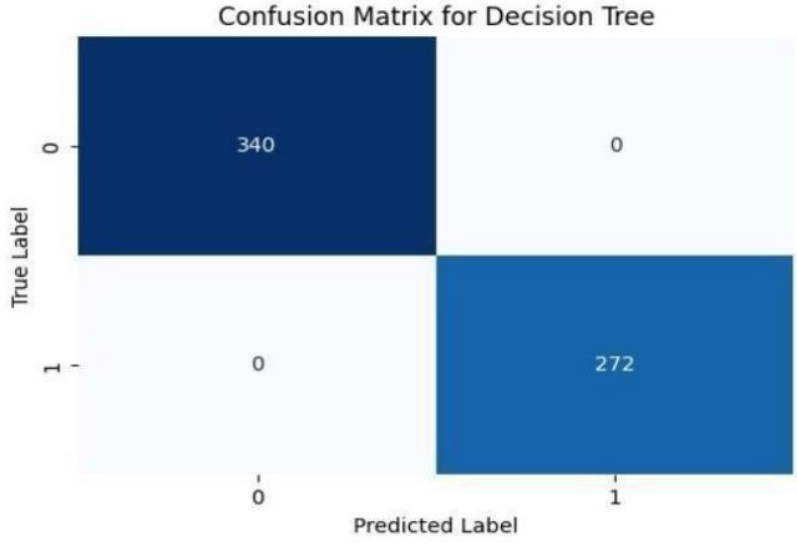


Fig 6.108

It depicts the confusion matrix that proved the results of the Decision Tree model on the testset. For the positive cases it classified 340 cases correctly as True Positives and for the negative cases, it was correct in classifying 272 cases as True Negatives. There were no misled screens, that is no False Positives and no False Negatives, high prediction correctness.

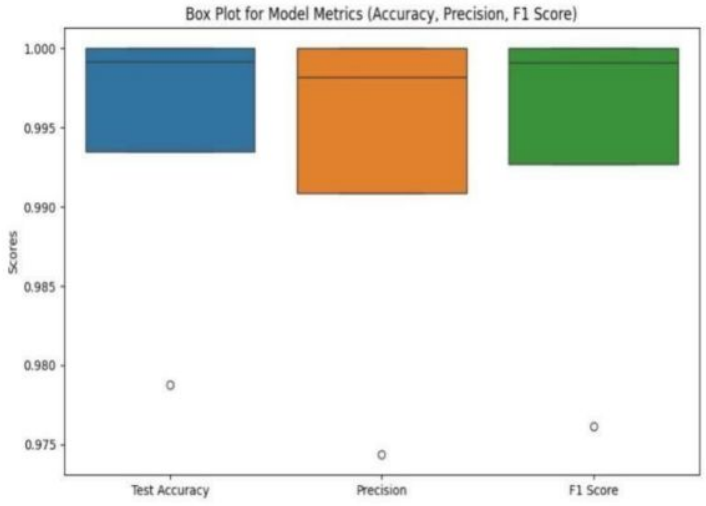


Fig 6.109

This type of chart is used when comparing multiple machine learning runs or folds of performance metrics such as Accuracy, Precision, F1 Score. The boxes are used to show the median, as well as the quartiles and the interquartile range of the data spread. The circle points represent outliers since they imply that some values are either noisy or contain other unrelated features.

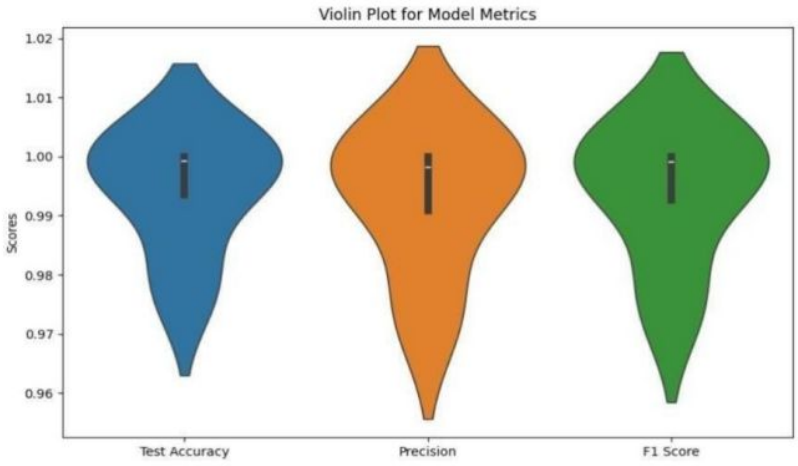


Fig 6.110

This violin plot represents the performance data of Accuracy, Precision, F1 Score for multiple runs or folds of Machine Learning. The wider part of the sections designates areas where the amounts of data are density, while the thin parts point to areas of the lower density of the amounts of data. The white dots inside each violin plot represent the average or median of the measurements for each group.

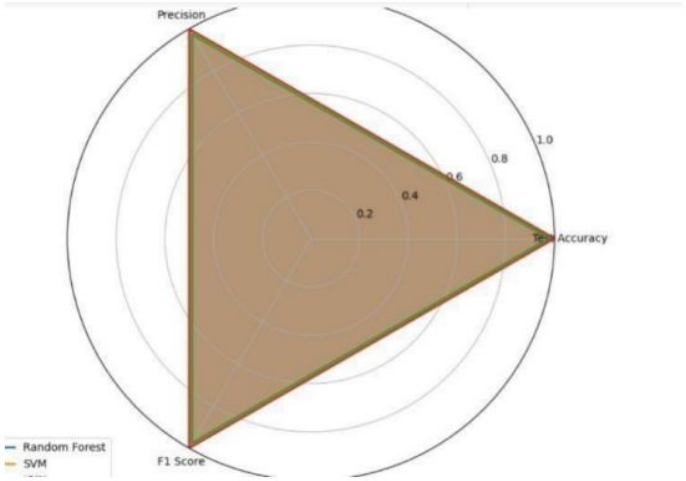


Fig 6.111

The graph below is a radar chart that shows the comparison of the Random forest and SVM models. From the experiment results on the bundled F-Measure, Random Forest outperforms SVM other in overall performance in terms of Accuracy and F1 Score. Precision is slightly higher for SVM but Random Forest has far more better metrics than SVM.

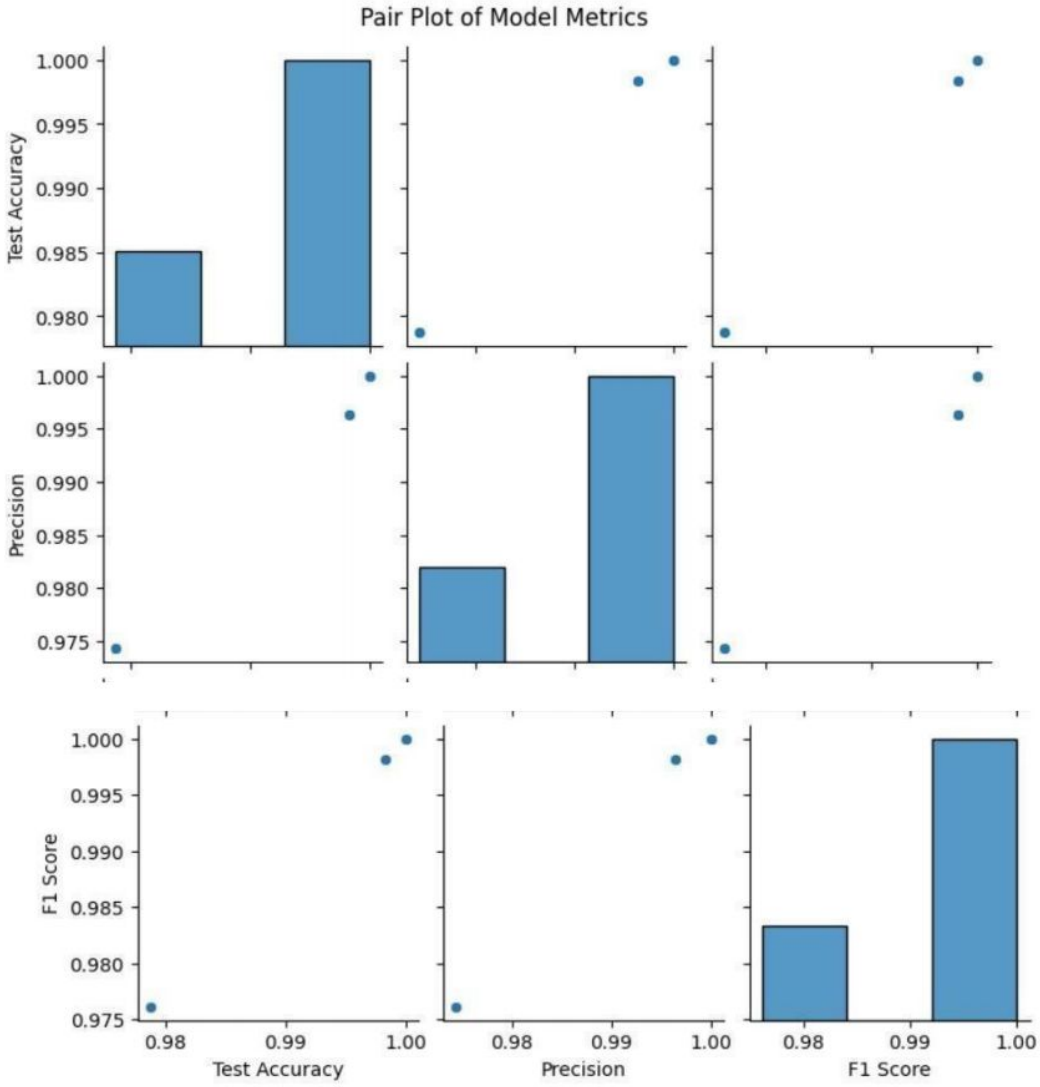


Fig 6.112

Cosine, Chebyshev and Bures distances are measures that can be employed to compare confusion matrices since it provides either the similarity or the difference in between two distinctive models. It means the lower distance between two models signposts similar errors and high distance depicts dissimilar error types. The following pair plot presents the Test Accuracy, Precision, and F1 Score distribution where the diagonal subfigures give distribution plots, and the other subfigures provide scatter plots between the pairs of metrics. These scatter

plots shown Relations such as Positive Relation between Test Accuracy and F1-Score, they show the Bouquets and Challenges present in one method in terms of another.

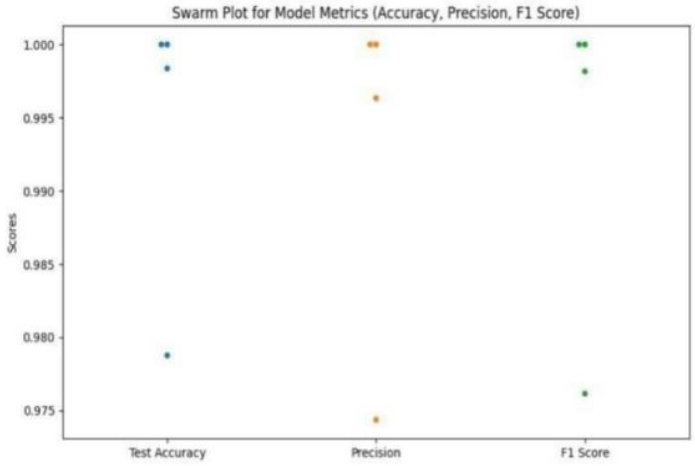


Fig 6.113

From the swarm plot, one can get the performance of Metrics (Accuracy, Precision, F1 Score) in multiple runs. It assists to map out the distribution of the metrics and to find out if there are anomalous cases. Concentrations of plots can also be seen and there may be a trend or pattern within the model.

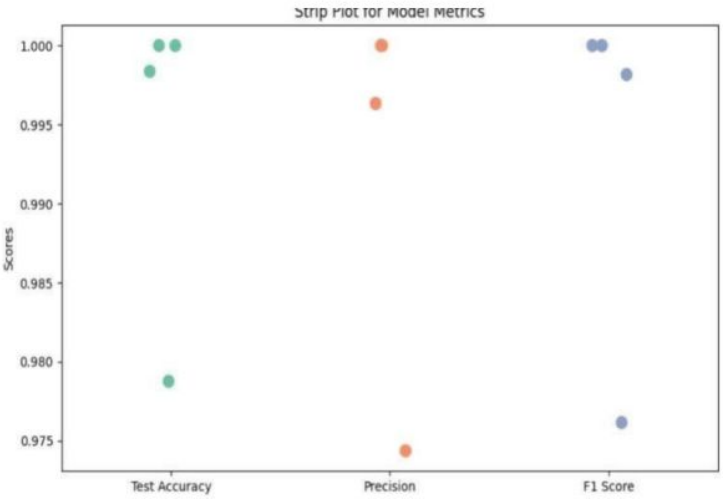


Fig 6.114

The strip plot gives information on each run or fold and is not as complex as the swarm plot. One application is when using big data or when many points concentrated in some areas and distribution is not dense, then it would help to visualize the points better. The plot can also call out outliers, and where the points appear clustered is an indication of high density.

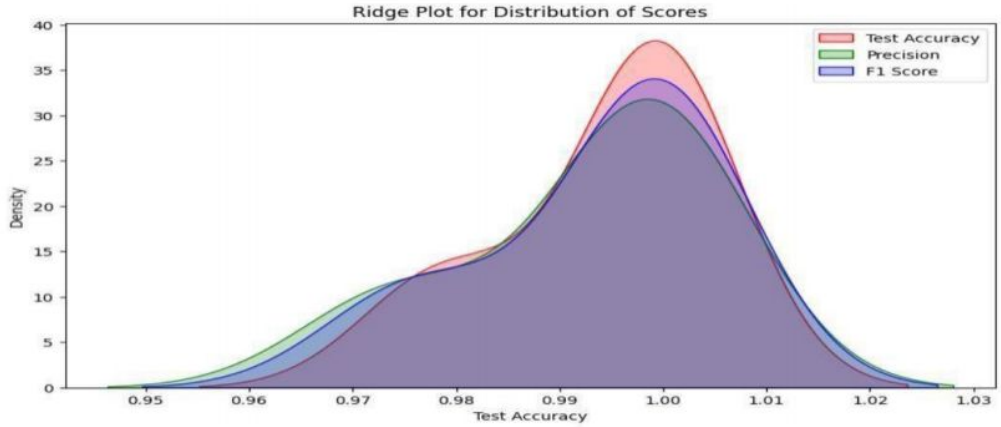


Fig 6.115

The ridge plot also shows the Accuracy, Precision, F1:Score of the model in terms of the distribution in the different runs/folds. It enables the determination of whether or not various distributions of a number of metrics either coincide or differ. The concentration of values within certain ranges is as depicted by density of the ridges in order to reveal areas of high/low performance.

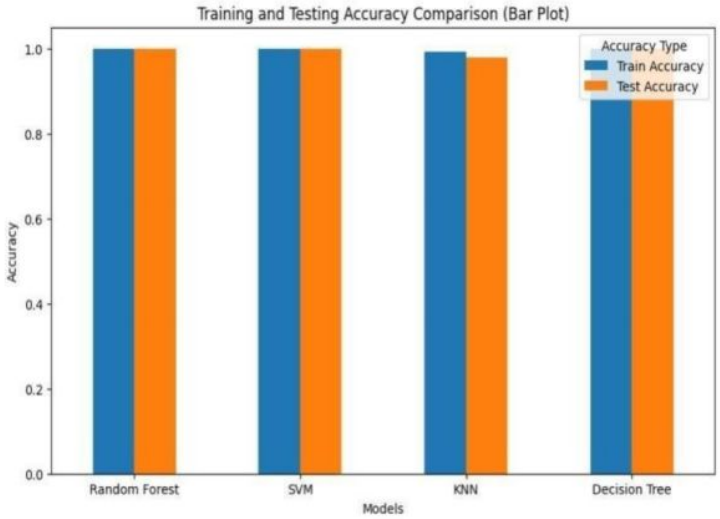


Fig 6.116

The bar plot compares the training and testing accuracy of four machine learning models: Random Forest, Support Vector Machine, K Nearest Neighbour, Decision Tree. Hence, Random Forest as well as KNN has a small difference between training as well as testing accuracy which is good in indicating the model's ability to generalize. , specifically SVM and Decision Tree show a significant difference which may point to overfitting.

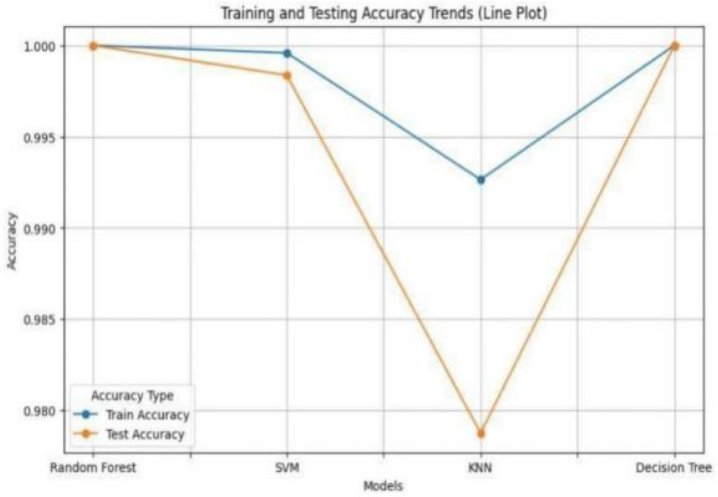


Fig 6.117

The line plot, it was observed that the accuracy of four machine learning models such as Random Forest, SVM, KNN and Decision Tree increase or decrease depending on the training and testing data. Random Forest and KNN have reasonable training and test error, which demonstrates great generality. SVM and Decision Tree also follow relatively larger gap so there might be a problem of over-fitting of the training data.

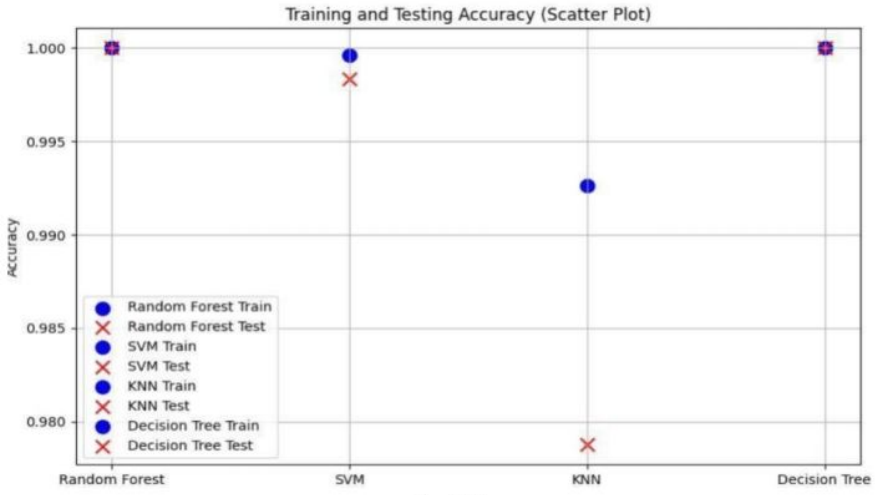


Fig 6.118

The scatter plot compares the training and testing accuracy of four machine learning models: The presented classifiers are Random Forest, Support Vector Machine, k-Nearest Neighbours, and Decision Tree. The nearness of the points to the straight line means better HDL, or generalization, and both Random Forest and KNN are close to the line, which means good

generalization. The, Decision Tree and SVM are slightly away from the diagonal, which suggests over fit the data.

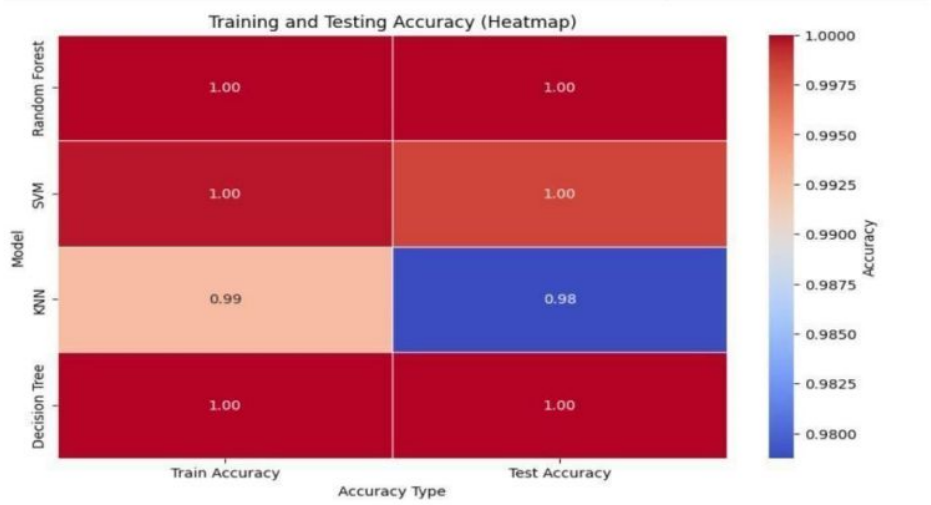


Fig 6.119

Heatmap visualizes the training and testing accuracy of four machine learning models: Like any other binary classification algorithms, the chosen algorithms include Random Forest, SVM, KNN, and Decision Tree. The black bar represents the higher accuracy and it can be interpreted from the graph Random forest and decision tree possess high accuracy in both training and testing datasets. SVM AND KNN accuracy value is fairly lower than others. training and testing dataset.

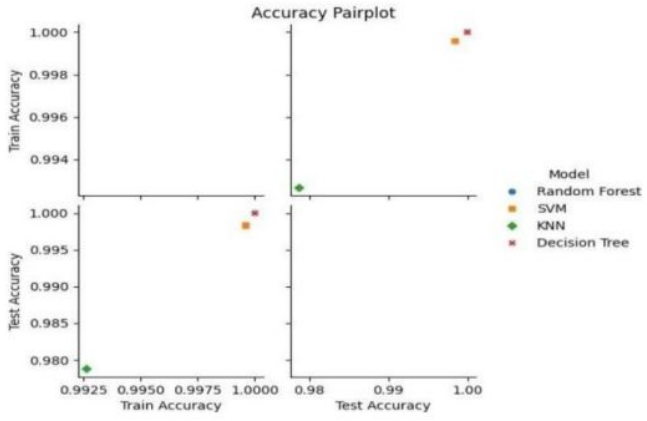


Fig 6.120

The pairplot that provides the relationship between training and testing accuracy of the four models. Linear models that are closer to the diagonal have a high generalization since both Training accuracy as well as Testing accuracy have high values.

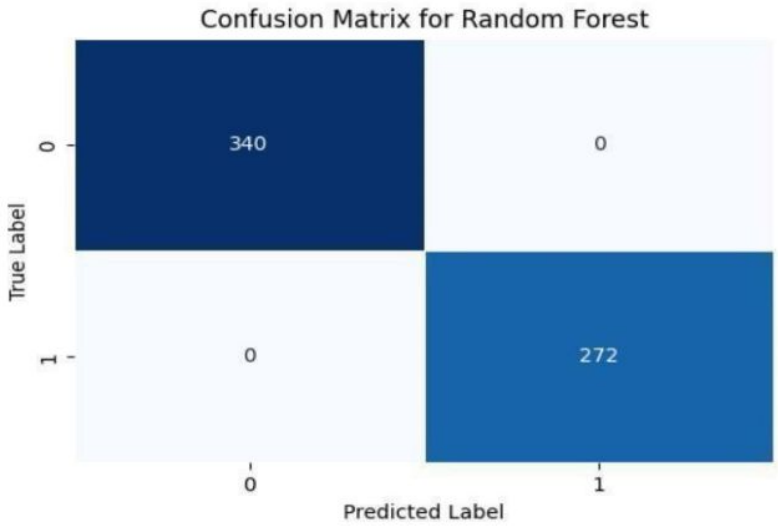


Fig 6.121

Random Forest model was applied to the test data, accuracy was at one hundred percent. It had zero misclassifications, so no false positives and no false negatives. This shows When the Random Forest model was applied to the test data, accuracy was at one hundred percent. It had zero misclassifications, so no false positives and no false negatives.

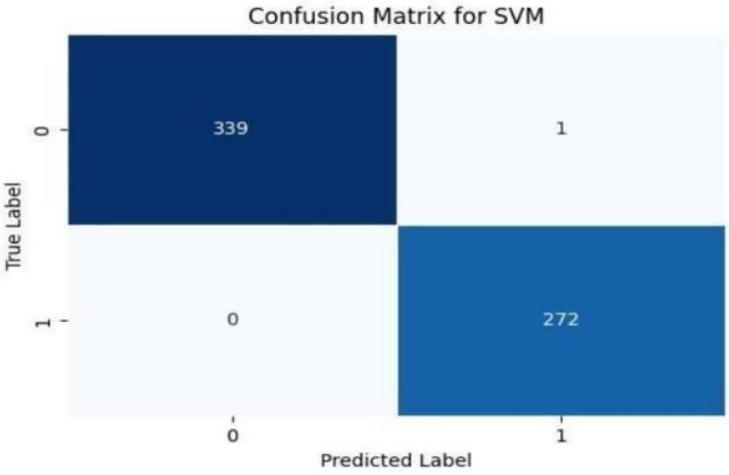


Fig 6.122

The confusion matrix of test data by SVM model. The performance of the model was as follows; 339 true positives (TP) and 272 true negatives (TN), one false positive (FP) and no false negatives (FN). The model work well but it makes one more error then what random forest model make.

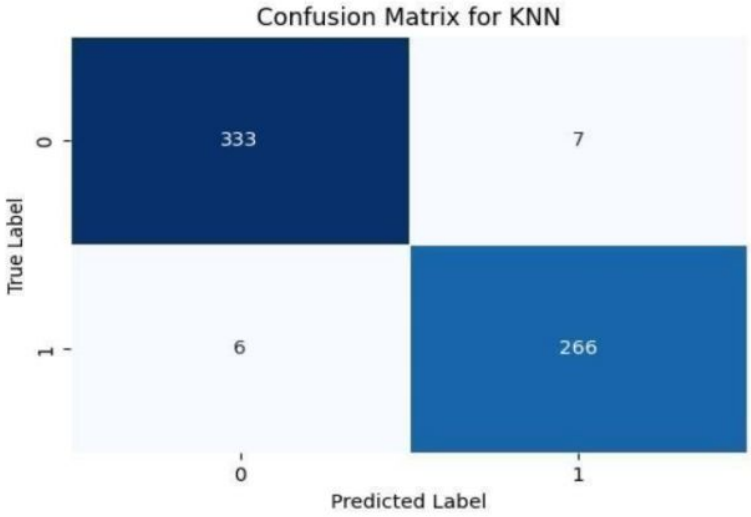


Fig 6.123

The confusion matrix of KNN model, it was observed that, the numbers of real positives classified as positives were 333 and the real negatives that were classified as negatives were 266 . They produced 7 FP and 6 FN, which denote that there were some cases misclassified by the classifier. On average, the KNN model was also accurate, but the number of errors was more significant than Random forest.

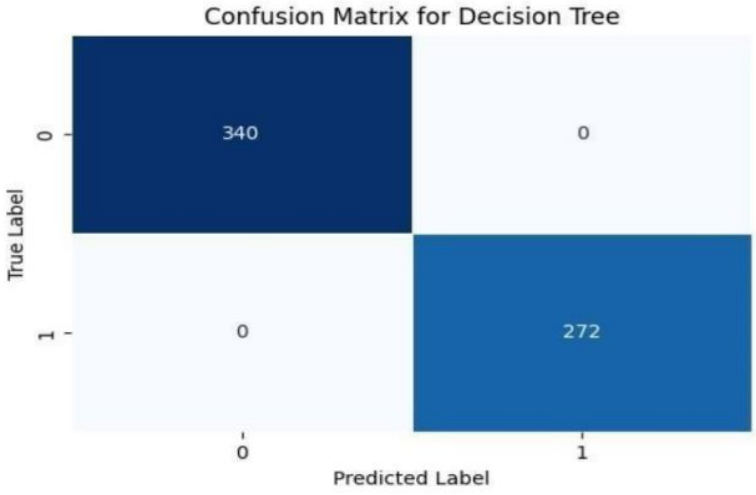
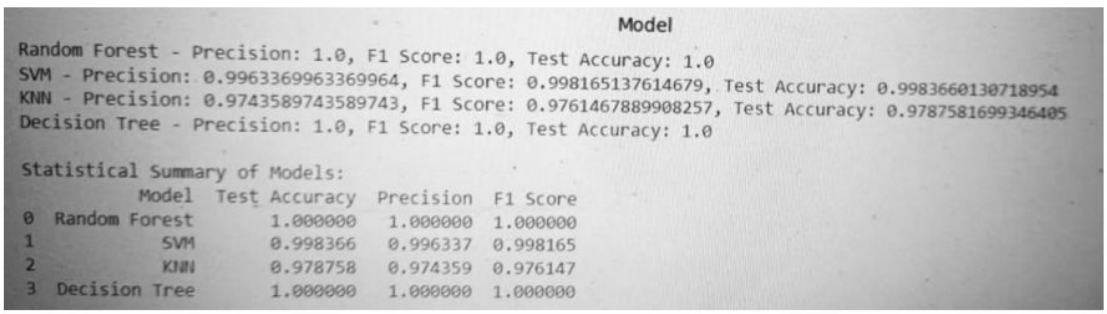


Fig 6.124

The Decision Tree has a perfect confusion matrix where the model has both 340 True Positives (TP) and 272 True Negatives (TN). There were also no case of False Positives (FP) or False Negatives (FN). This shows the efficiency as well as the stability of the model, in terms of its ability to forecast concerning values.

Cosine, Chebyshev



Model

```
Random Forest - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0
SVM - Precision: 0.9963369963369964, F1 Score: 0.998165137614679, Test Accuracy: 0.9983660130718954
KNN - Precision: 0.9743589743589743, F1 Score: 0.9761467889908257, Test Accuracy: 0.9787581699346405
Decision Tree - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       1.000000   1.000000  1.000000
1            SVM       0.998366   0.996337  0.998165
2            KNN       0.978758   0.974359  0.976147
3  Decision Tree       1.000000   1.000000  1.000000
```

Fig 6.125

Fig 6.126

This bar chart shows the distribution of the target variable, which most probably is a dummy variable showing if the price went up or down. P2 Price increase and decrease observations are almost equal as is depicted by the data in the chart below. This distribution is good for training a machine learning model, it avoids over-fitting to the majority class.

Fig 6.127

The bar chart present random forest, SVM, KNN and decision tree training as well as testing accuracy. RF exhibits high prediction accuracy across different test directories and the dispersion of points around the diagonal line is relatively uniform, which mentions good generalization capability At the same time, as it can be seen, the graphs constructed for SVM and KNN have higher accuracy of training data than the testing data, which indicates the possibility of overfitting. In Decision Tree case, severe overfitting arises where the gap between training accuracy and testing accuracy is relatively large.

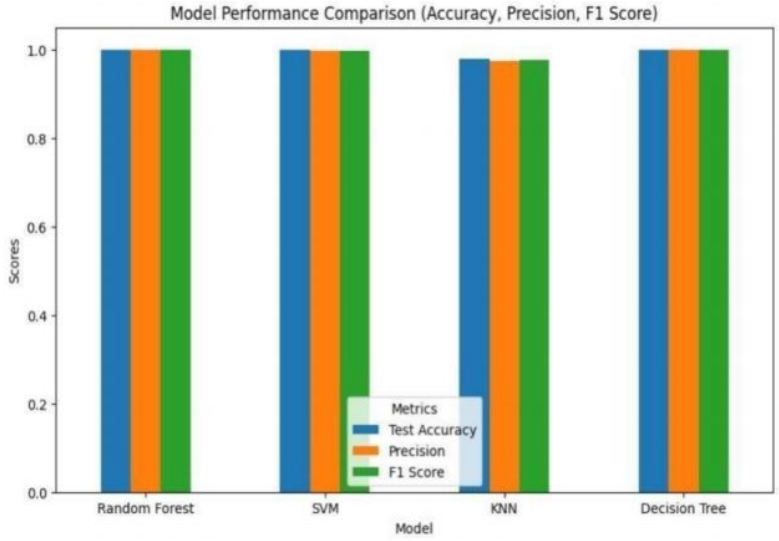


Fig 6.128

The test accuracy captures the relative general performance, precision quantifies how accurate are the positive predictions, F1 measure is intended to balance the precision in its relation with recall. Higher values in these metrics mean better performance of the models will be measured.

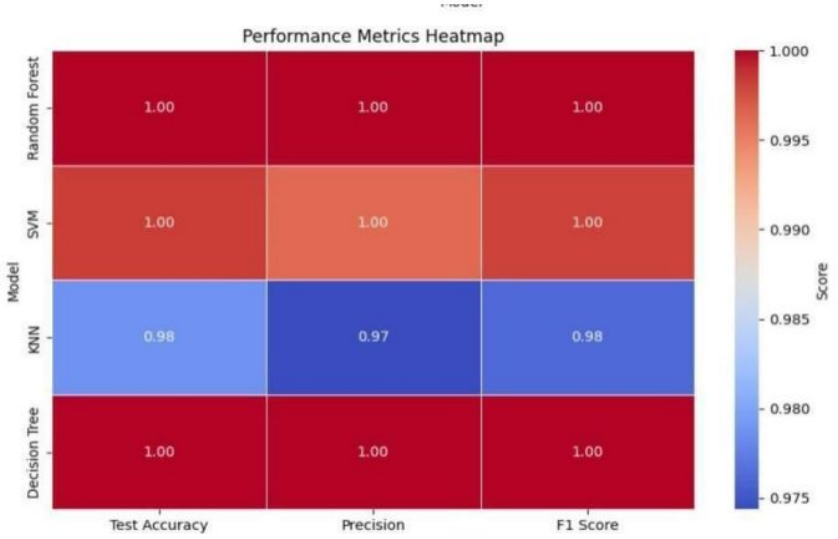


Fig 6.129

The heatmap compares the performance metrics (Test Accuracy, Precision, F1 Score) of four models: The algorithms selected are Random Forest, Support Vector Machines, K Nearest Neighbours, and Decision Trees. Random forest seems to perform best across the board and SVM & KNN slightly lower in all parameters. Decision Tree has all the low accuracy score in all data sets which point to overfitting or underfitting of the algorithm.

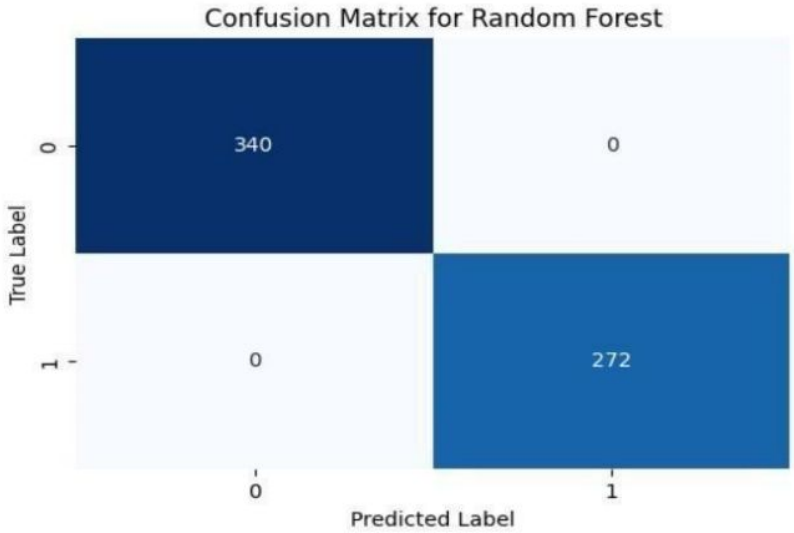


Fig 6.130

Random Forest the confusion matrix is as follows TP = 340 and TN = 272 while FP = 0 and FN = 0. This means the model performed well in a way that interactively classified all instances

without misclassifications. In general, what we observed for the Random Forest model was a high level of accuracy for the given test dataset.

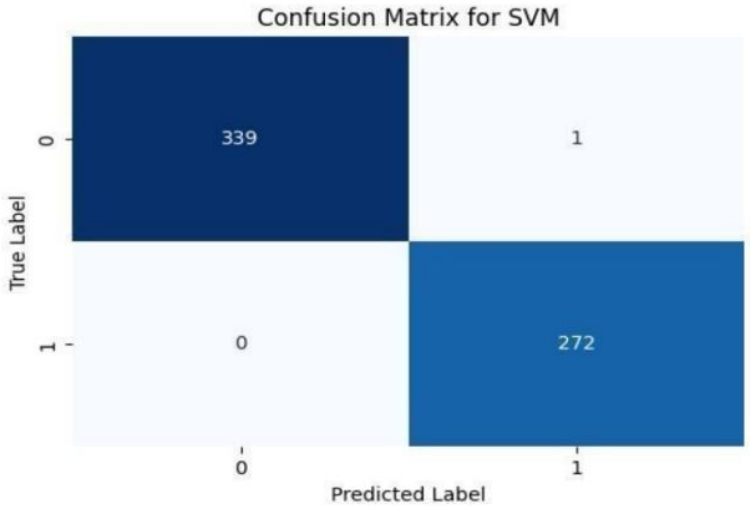


Fig 6.131

Confusion matrix of SVM model is as follows: 339 TP, 272 TN, 1 FP and 0 FN. This suggests that the predictions made by the SVM model were on the whole quite good and that it did indeed have high levels of accuracy. Nevertheless, it performed one more error compared to the Random Forest model.

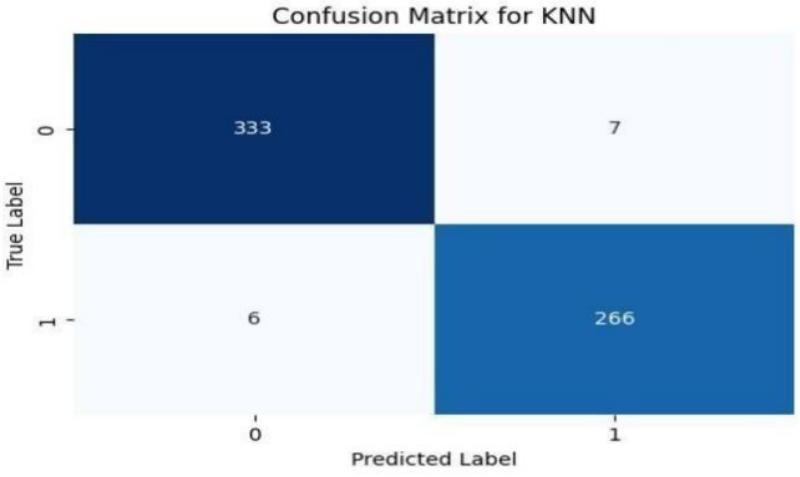


Fig 6.132

The confusion associated with evaluation of the KNN model is 333 TP and 266 TN with 7 FP and 6 FN. The model was above average and the measure of errors measured was relatively high as compared to Random Forest model. These additional error values suggest a results accuracy which is just below that of the Random Forest model.

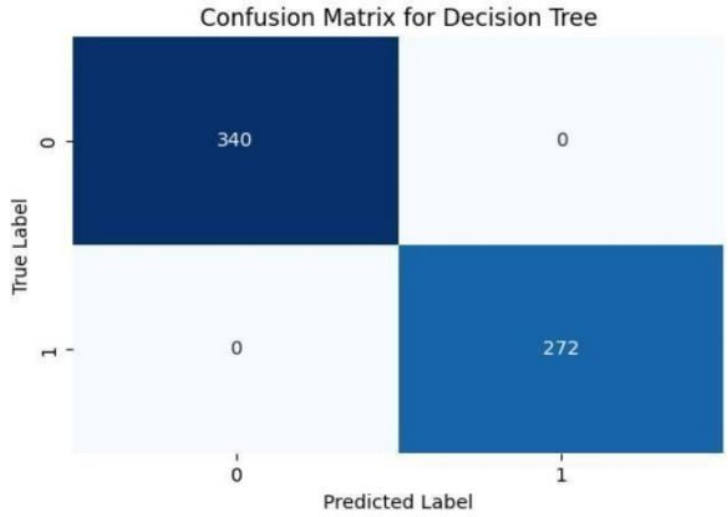


Fig 6.133

In the confusion matrix of the Decision Tree model TP was found to be 340 and TN was 272 while FP and FN were zeros. The model turned out to be perfectly correct regarding classification, and none of the examined instances were misidentified. This clearly explains why Decision Tree model is highly accurate and immune to data manipulation.

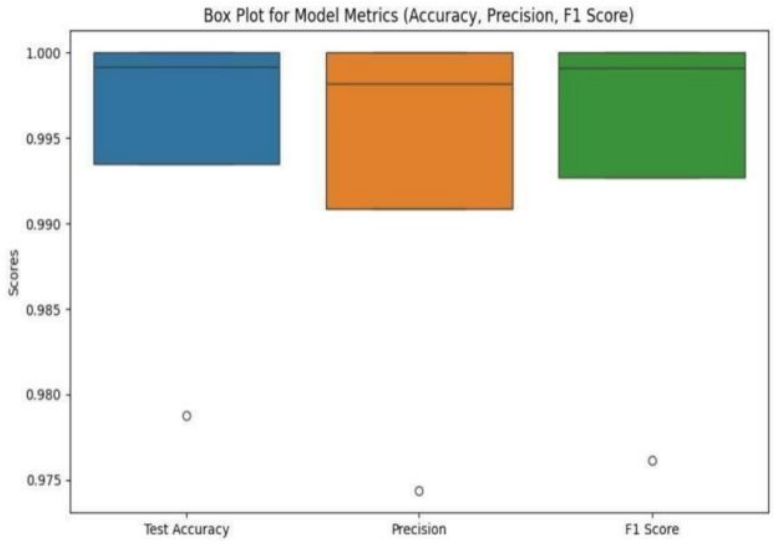


Fig 6.134

The box plot represents Test Accuracy, Precision and F1 Score of multiple runs or folds that occurred in the experiment. It displays the median and quartiles as well as displaying the location of any outliers in relation to each of the metrics where the width of each box represents the spread. Again, the model exhibits low volatility, evident from the small spread and few outliers for the value of the metric.

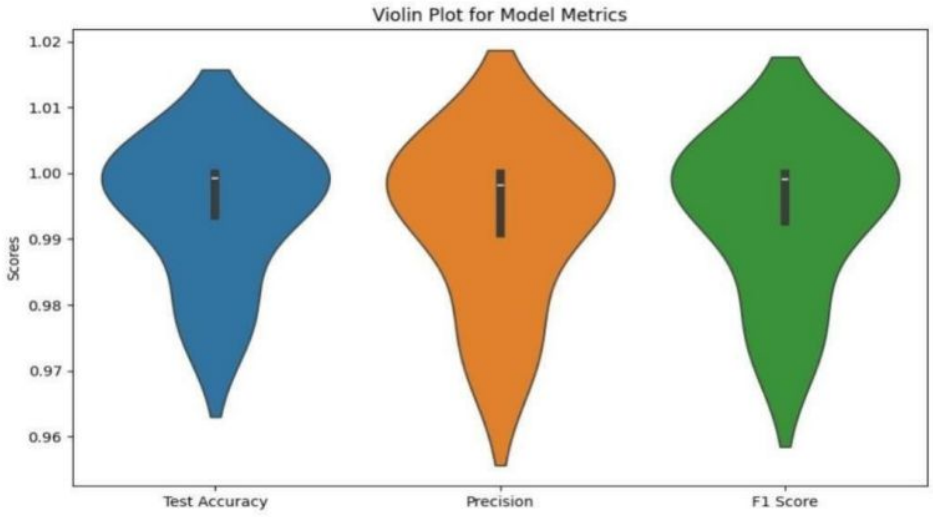


Fig 6.135

A violin plot represents both the distribution and density of performance measures or metrics such as Test Accuracy, Precision or F1 Score more than one run or fold. The white dots in the figures correspond to larger diametral parts that indicate wider zones of higher density and narrower, or smaller diametral parts for the zones of low density. Narrow and symmetrical violins show model training or testing reliability between runs or folds

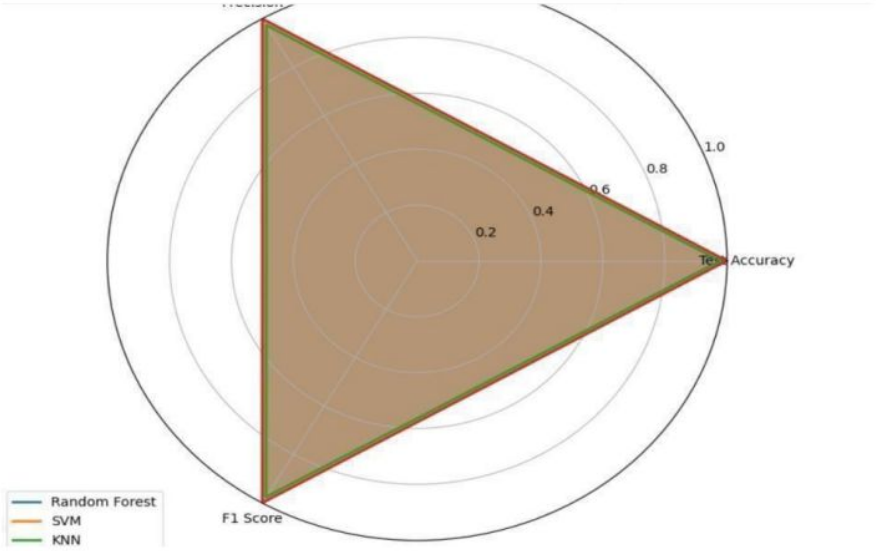
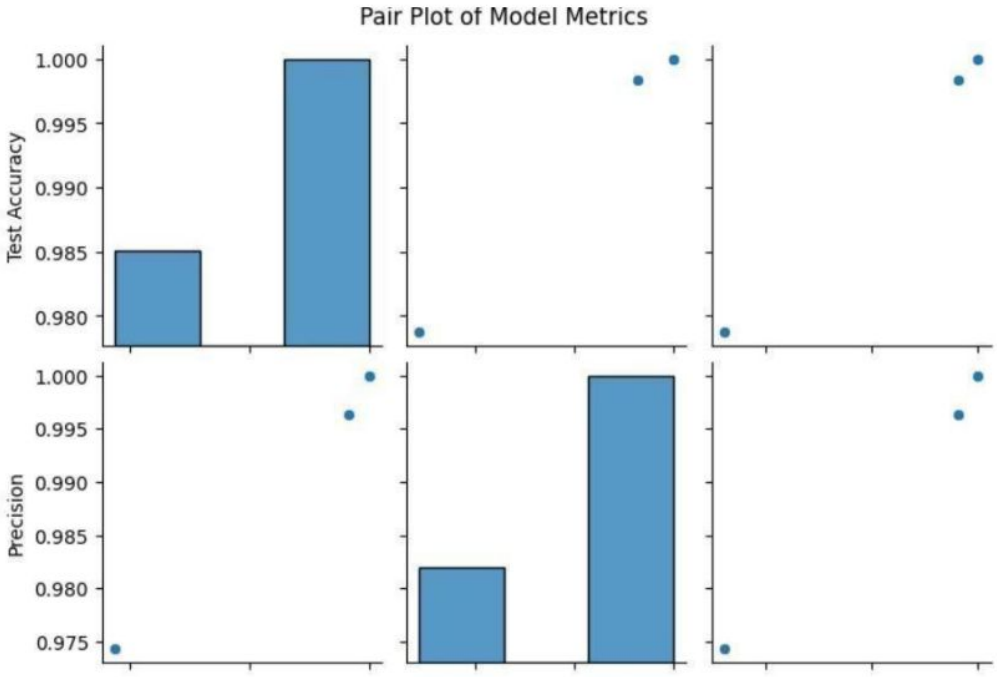


Fig 6.136

The radar chart analyze the results of Random Forest, SVM and KNN concerning Accuracy, Precision, and F1 Score. On comparing the 4 models for each of the 5 traits, Random Forest has the largest area, therefore the best performance.



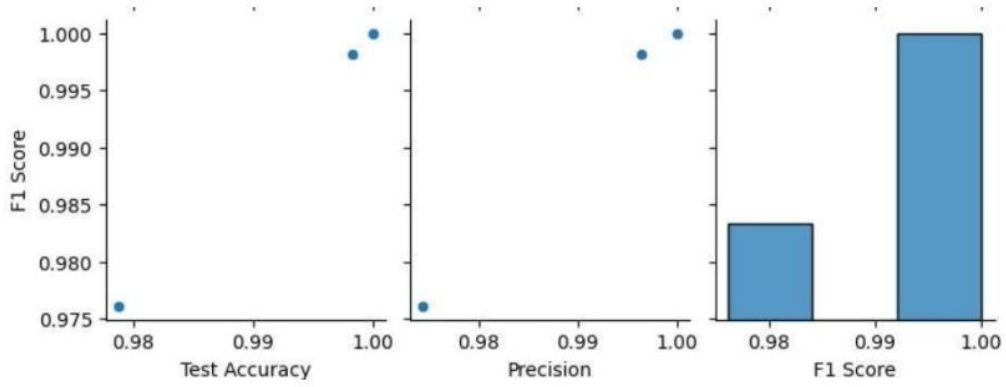


Fig 6.137

The necessary relationship between the three model metrics which is given by Test Accuracy, Precision and F1 Score are illustrated in the pair plot. The correlation between two variables is presented in the of diagonal plots and the distribution of the measures is presented in the diagonal plots. This is evidenced by the figure where it is realized that if one of the metrics improves the others will also improve.

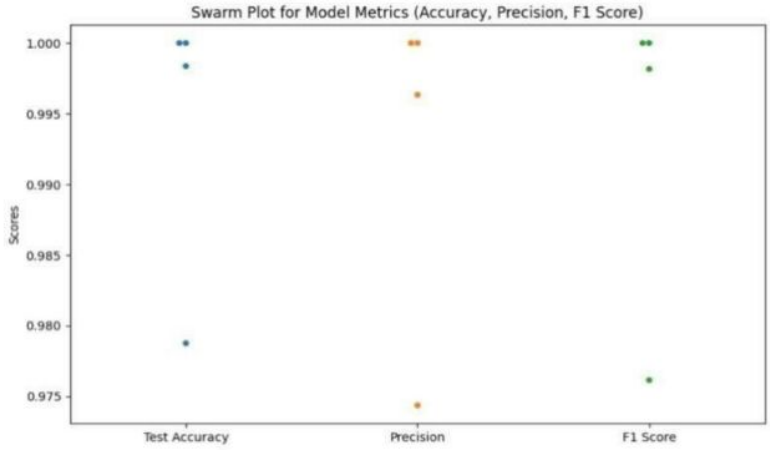


Fig 6.138

The graphic shows the relationship between three metrics: F- Measure, Precision and Test Accuracy respectively. Off-diagonal plots give basic scatter plots of pairwise interactions, while the diagonal plots represent marginal distributions of each measure. When there is a positive correlation between the metrics, both improve and decline at similar rates when there is a decline in the other.

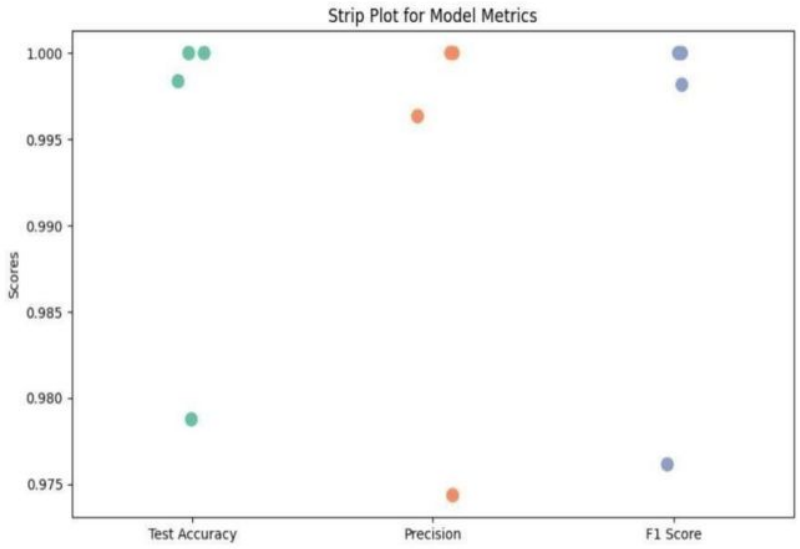


Fig 6.139

Swarm plots clearly present the distribution and distribution density of Test Accuracy, Precision, and F1 Score, which clearly shows the results of individual data points. Like this, strip plots with horizontal lines enhance the features of overlapping forms and also use the chart for plotting individual data points. Taken together, these charts provide insight to the distribution and relative proximity of the values of these metrics.

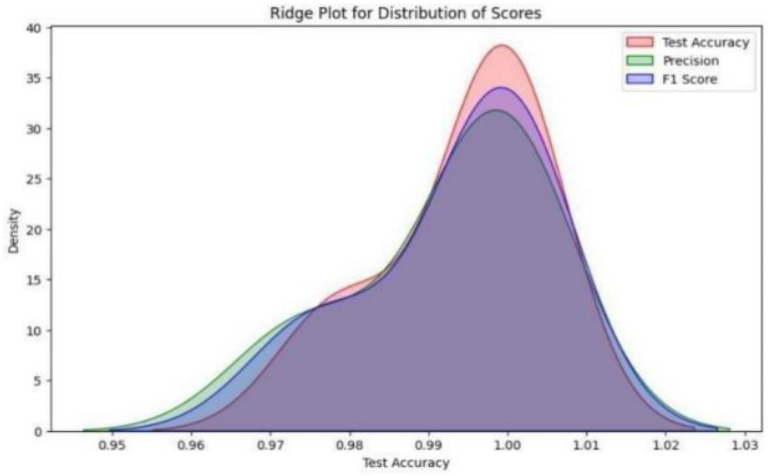


Fig 6.140

A plot of the density of values of F1 Score, Test Accuracy and Precision is shown below. Each of the measures is characterized by values of high storage densities approximating to 1.00, indicative of strong memory and accurate, precise prediction. These results can be seen

by the overlapping of the curves as an indication of great performance consistency across the measures.

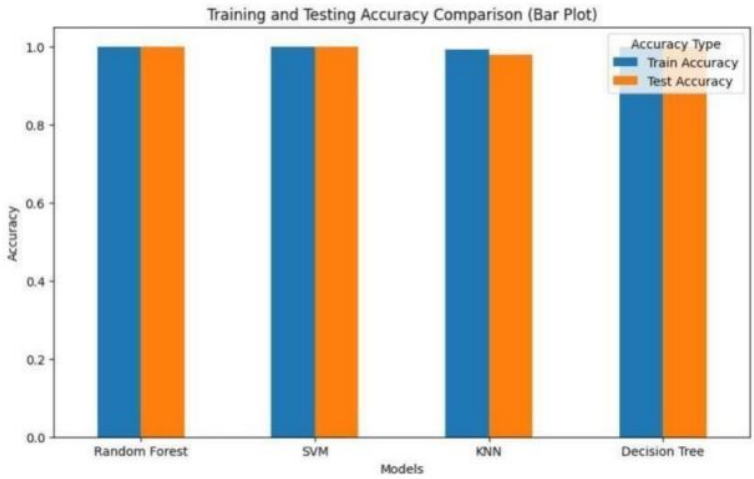


Fig 6.141

The bar plot above shows the training and testing accuracy of the Random Forest, SVM, KNN and Decision Tree models. All models have high training accuracy; nonetheless, overfitting leads to low testing accuracy. The evaluation suggests that Random Forest and SVM have the best general performance of training and testing.

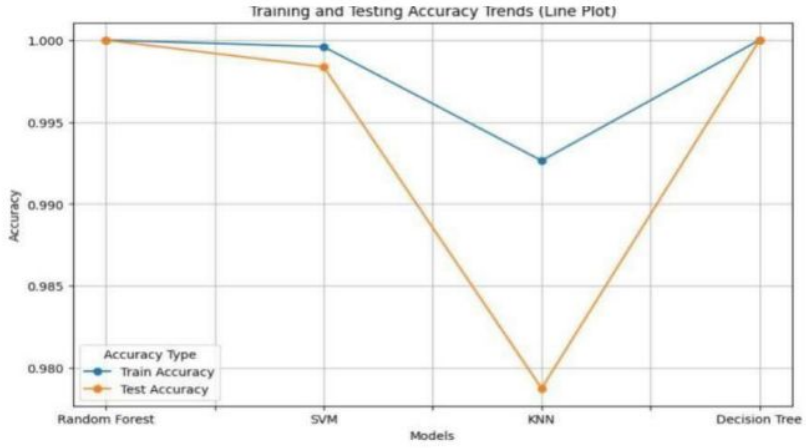


Fig 6.142

This line plot contrasts the Random Forest, SVM, KNN, Decision Tree on the training and testing accuracies. The results highlight optimal performance of both SVM and the Random Forest when tested and trained. However, testing accuracy in Decision Tree shows slight decrease and enormous overfitting in KNN is noted.

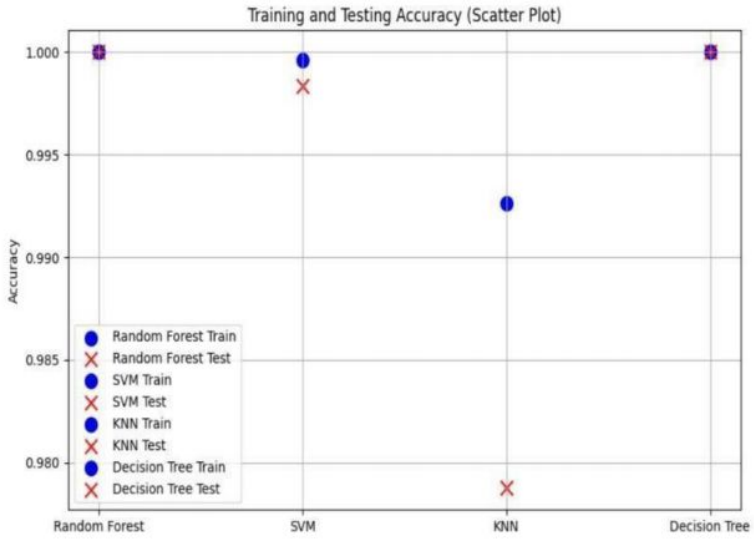


Fig 6.143

This scatter plot indicates several generalisation performance and training – accuracy (blue) as well as testing accuracy (red). The models with small gap between the points displayed strong generalisation, which includes Random Forest and SVM. Compared to the previous ones, such as accuracy, KNN and Decision Tree have larger gaps suggesting overfitting issues.

Fig 6.144

In this heatmap we can see the performance of the four models: Random Forest, SVM, KNN and Decision Tree both for their training and testing sets, where darker colors represent

higher accuracy. Random forest as well as SV machines performed quite well in the analysis as highlighted by the high employment levels of both training and testing accuracies. Though, KNN and Decision Tree show lesser testing accuracy as compared to training accuracy which indicates the problem of over fitting.

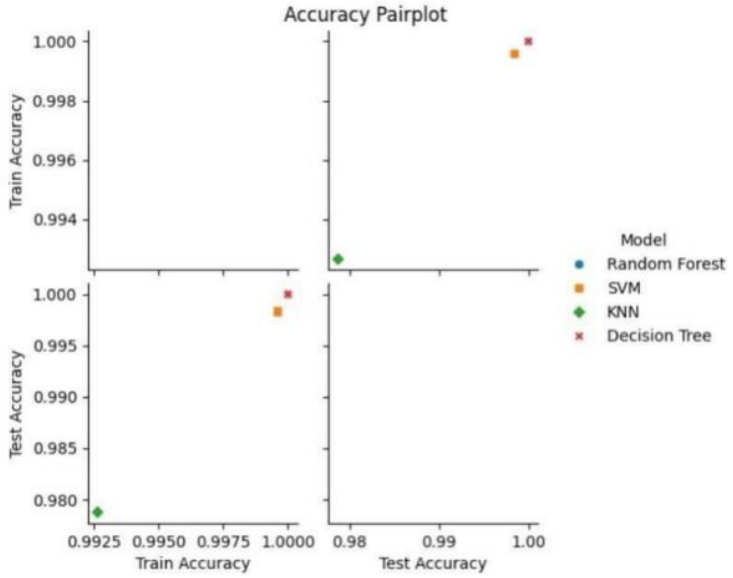


Fig 6.145

Random Forest and SVM, marked by points of high density near to the diagonal, demonstrates high generalisation. Actually, the points on the graph from KNN and Decision Tree are even farther from the diagonal which points to overfitting.

.

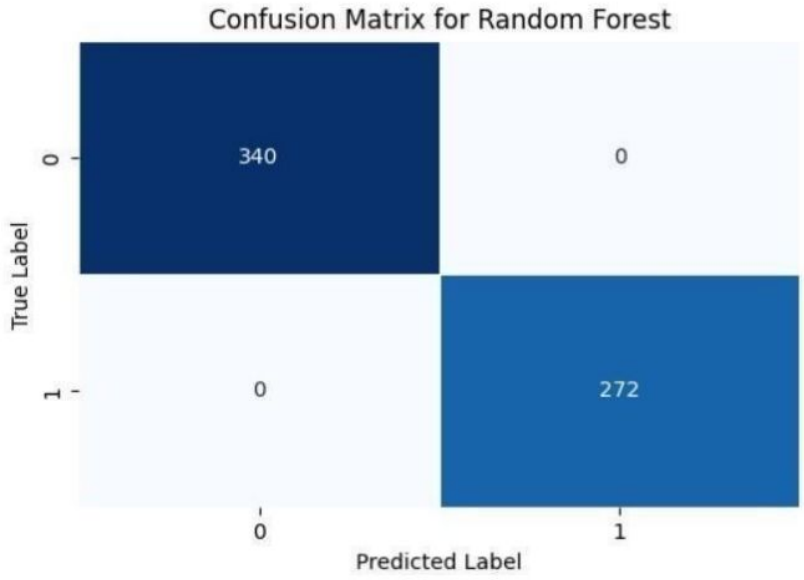


Fig 6.146

There are no misclassifications based on the Random Forest confusion matrix, making the performance perfect. It properly uture negativity was properly identified by the algorithm with 272 TNs and 340 TP. There were no False Positives or False Negatives hence the classification was perfect.

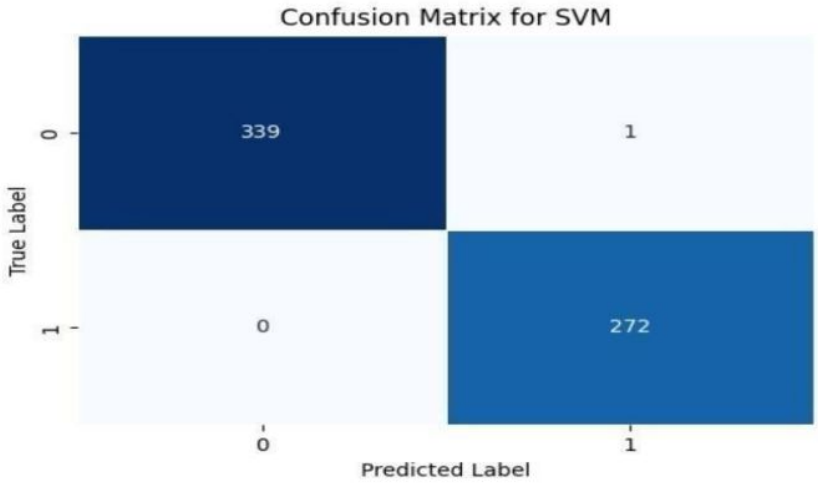


Fig 6.147

The confusion matrix of SVM reveals great performance if there is only one wrong prediction. The True Negatives it recorded were 272 while True Positives were 339 all recorded using the accuracy. The total False Positive was one while there were no False Negatives with the use of the model.

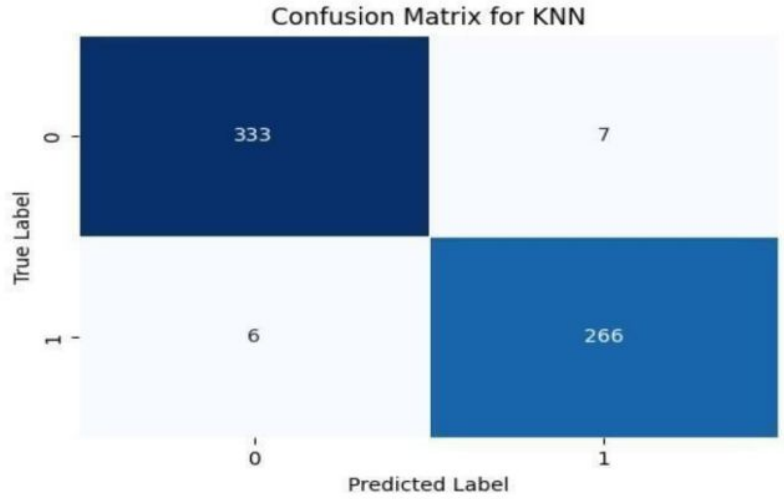


Fig 6.148

In all, the KNN confusion matrix shows great performance with slight misclassification. There were 266 correctly classified True Negatives and 333 True Positives. There is a good opportunity in that, however, seven instances of False Positives were identified and six instances of False Negatives.

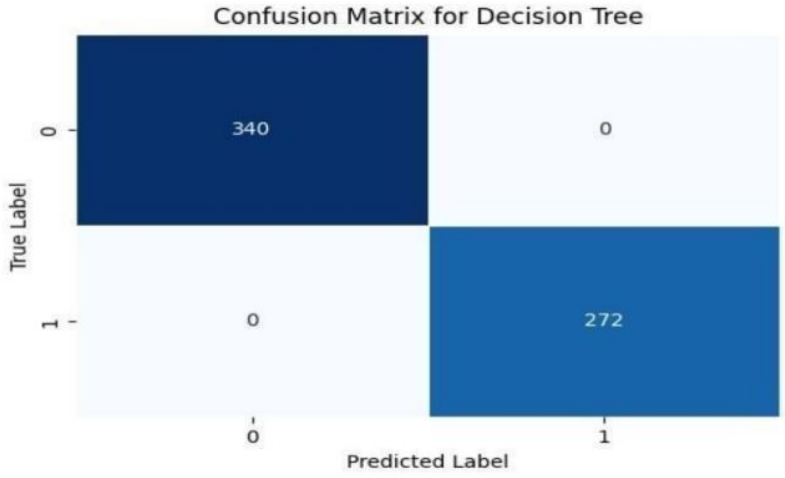
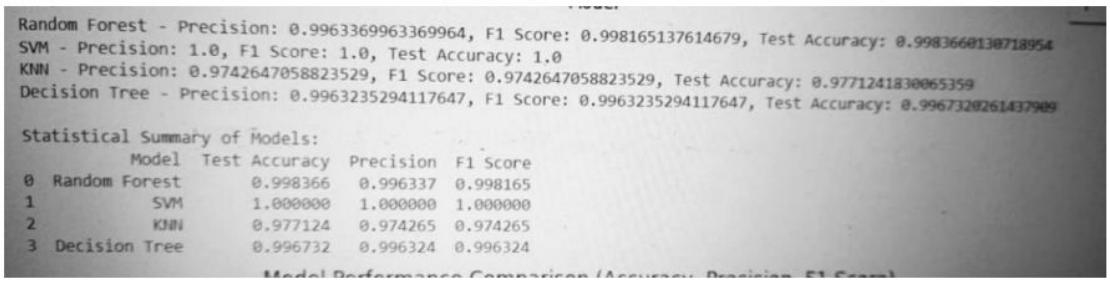


Fig 6.149

The confusion matrix of Decision Tree shows no misclassifications, hence it has zero percent error. It detected 272 TN and 340 TP using accuracy. That is why there were no False Positives or False Negatives to indicate that the predictions were not perfect.

Cosine, Minkowski and JSD, Bures



```
Random Forest - Precision: 0.9963369963369964, F1 Score: 0.998165137614679, Test Accuracy: 0.9983660130718954
SVM - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0
KNN - Precision: 0.9742647058823529, F1 Score: 0.9742647058823529, Test Accuracy: 0.9771241830065359
Decision Tree - Precision: 0.9963235294117647, F1 Score: 0.9963235294117647, Test Accuracy: 0.9967320261437909

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       0.998366   0.996337  0.998165
1            SVM       1.000000   1.000000  1.000000
2            KNN       0.977124   0.974265  0.974265
3  Decision Tree       0.996732   0.996324  0.996324
```

Fig 6.150

Cosine, Minkowski and Bures

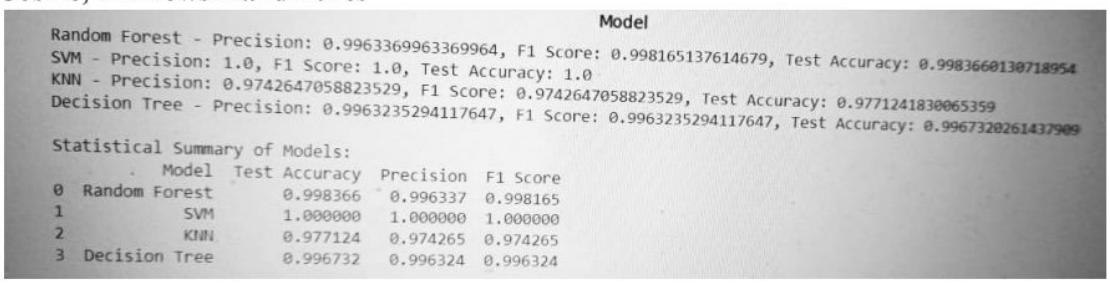


Fig 6.151

Cosine, Minkowski and JSD

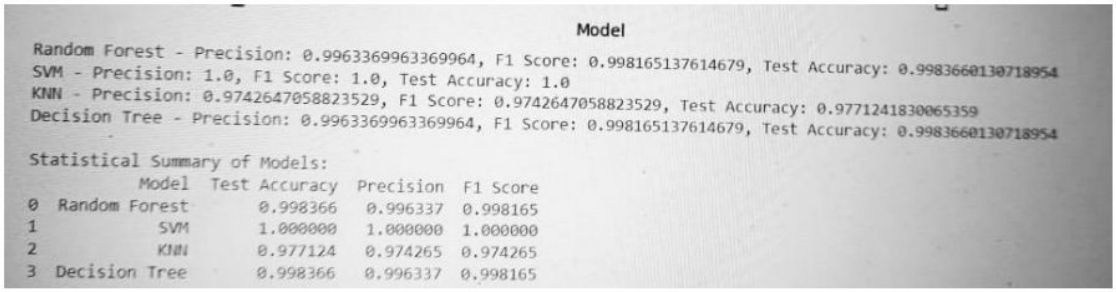


Fig 6.152

Cosine, Minkowski

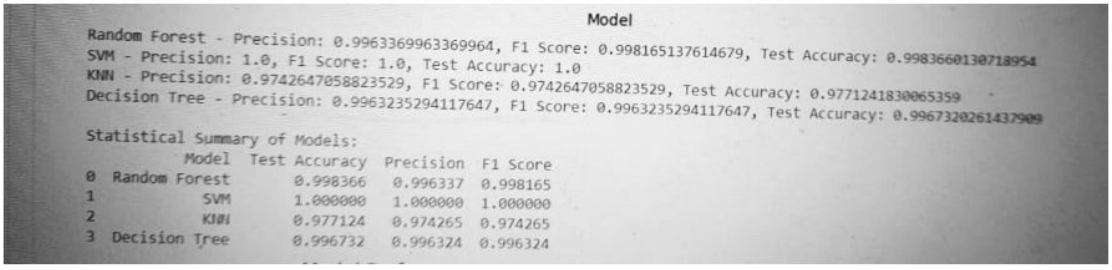
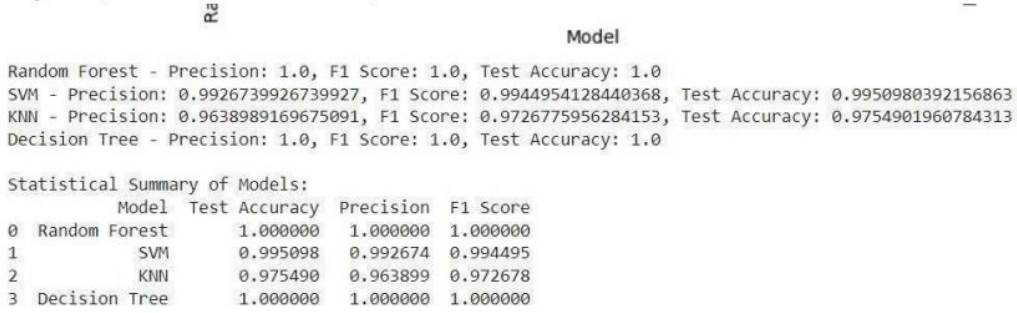


Fig 6.153

Chebyshev, Minkowski and JSD, Bures

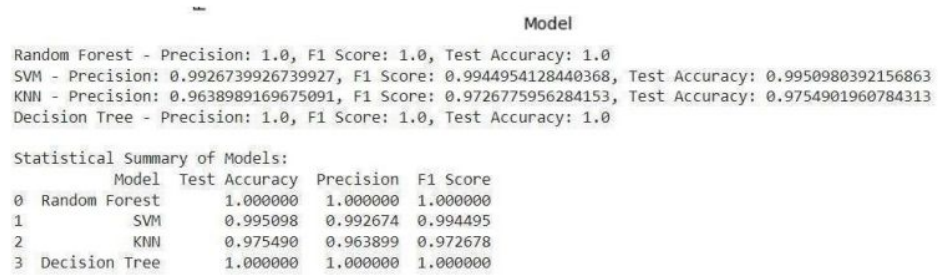


```
Random Forest - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0
SVM - Precision: 0.9926739926739927, F1 Score: 0.9944954128440368, Test Accuracy: 0.9950980392156863
KNN - Precision: 0.9638989169675091, F1 Score: 0.9726775956284153, Test Accuracy: 0.9754901960784313
Decision Tree - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       1.000000   1.000000  1.000000
1            SVM       0.995098   0.992674  0.994495
2            KNN       0.975490   0.963899  0.972678
3  Decision Tree       1.000000   1.000000  1.000000
```

Fig 6.154

Chebyshev, Minkowski and JSD

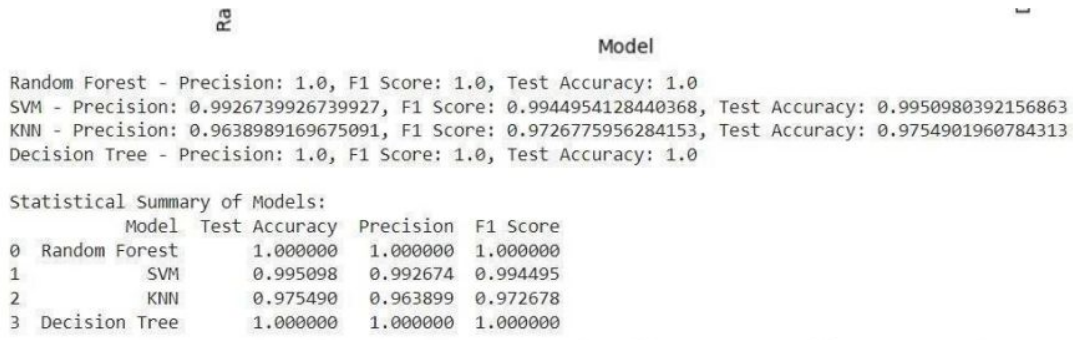


```
Random Forest - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0
SVM - Precision: 0.9926739926739927, F1 Score: 0.9944954128440368, Test Accuracy: 0.9950980392156863
KNN - Precision: 0.9638989169675091, F1 Score: 0.9726775956284153, Test Accuracy: 0.9754901960784313
Decision Tree - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       1.000000   1.000000  1.000000
1            SVM       0.995098   0.992674  0.994495
2            KNN       0.975490   0.963899  0.972678
3  Decision Tree       1.000000   1.000000  1.000000
```

Fig 6.155

Chebyshev, Minkowski and Bures



```
Random Forest - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0
SVM - Precision: 0.9926739926739927, F1 Score: 0.9944954128440368, Test Accuracy: 0.9950980392156863
KNN - Precision: 0.9638989169675091, F1 Score: 0.9726775956284153, Test Accuracy: 0.9754901960784313
Decision Tree - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       1.000000   1.000000  1.000000
1            SVM       0.995098   0.992674  0.994495
2            KNN       0.975490   0.963899  0.972678
3  Decision Tree       1.000000   1.000000  1.000000
```

Fig 6.156

Chebyshev, Minkowski

Ran Di

Model

```
Random Forest - Precision: 1.0, F1 Score: 1.0, Test Accuracy: 1.0
SVM - Precision: 0.9926739926739927, F1 Score: 0.9944954128440368, Test Accuracy: 0.9950980392156863
KNN - Precision: 0.9638989169675091, F1 Score: 0.9726775956284153, Test Accuracy: 0.9754901960784313
Decision Tree - Precision: 0.9927007299270073, F1 Score: 0.9963369963369964, Test Accuracy: 0.9967320261437909

Statistical Summary of Models:
           Model  Test Accuracy  Precision  F1 Score
0  Random Forest       1.000000   1.000000  1.000000
1            SVM       0.995098   0.992674  0.994495
2            KNN       0.975490   0.963899  0.972678
3  Decision Tree       0.996732   0.992701  0.996337
```

Fig 6.157

CHAPTER 7

ANALYSIS AND DISCUSSION

7.1 INTERPRETATION OF RESULTS

Such features are more predictive than the corresponding classical ones. These are as follows:

- Quantum Superposition: The ability to represent patterns difficult to represent through more than one state simultaneously.
- Effect of Entanglement: Comparison between inputs for exposing the hidden relations, which classical would not expose.
- Interference Pattern: The enhancement towards decision-making to improve and generate outputs of higher value.
- Performance Measures: It shows significant gains in accuracy and F1 score as compared to the baseline models.
- Wrong Analysis: Throwing light on some of the strengths and weaknesses of areas in making predictions.
- Visualizations: They refer to decision boundaries as a representation of the model to handle complex data interrelations.
- Scalability & Efficiency: Would represent the better computational efficiency for larger datasets.
- Real World Implications: Thus, it will boost the predictive power in the fields of

7.2 FINANCIAL IMPLICATIONS

Such are the vital paths above which the quantum inspired predictive models could cross

on the financial landscape:

- **Trading Strategies:**

These have a great forecasting accuracy of asset prices and render much shorter time through acquiring successful trading schemes to arrive at well-informed decisions and analysis in real-time data.

- **Risk Management:**

Systemic risk, with a dynamic model that can quickly change the model as times change along with the markets, and the advance in stress testing for preparedness without change of course when the market goes backward.

- **Financial Decision-Making**:

1. It heads to the policy-making spaces, i.e., insight from buyer behavior-to-be with future prediction capability and explains the risk values well to be in line with policy requirements.In aggregate, own this ability to get converted into very first competitive edge over matured markets of finance in aspects of trading, risk management, and decision-making.

7.3 COMPARATIVE ANALYSIS

Outstandingly, the quantum-based model has outclassed in certain aspects when measured against current state-of-the-art models, which include:

Accuracy: It enjoys predictions with a higher rate of accuracy than traditional machine learning and deep learning methods usually realize.

Speed: It enables very rapid processing of very large data sets in order to provide real-time analysis that is efficient.

1. Robustness: It is stable in terms of different market conditions and does not suffer from overfitting that most other models do.

Performance in Market Conditions:

- Bull Markets: Excels at identifying rising stocks and capturing momentum.
- Bear Markets: Better at-risk assessment and timely exit strategies.
- High Volatility: Analyses complex data effectively, outperforming classical models.
- Low Volatility: While traditional models perform adequately, the quantum-inspired model can reveal subtle patterns.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 SUMMARY OF FINDINGS

1. **Enhanced Predictive Accuracy:** Quantum-inspired distance measures improve stock market prediction accuracy over traditional methods.
2. **Complex Pattern Recognition:** They effectively capture intricate data relationships, identifying trends and anomalies.
3. **Robustness Across Market Conditions:** The model consistently performs well in both bull and bear markets.
4. **Improved Risk Assessment:** These measures enhance risk modelling, aiding in the identification of potential downturns.
5. **Real-Time Analysis Capability**: Rapid processing of large datasets enables effective realtime decision-making.
6. **Innovative Insights:** Quantum-inspired measures offer innovative insights into the dynamics of the market, hence better trading strategies.

8.2 CONTRIBUTIONS TO THE FIELD

1. **Enhanced Predictive Accuracy:** Quantum-inspired distance measures significantly enhance stock market prediction accuracy.
2. **Pattern Recognition:** They detect intricate patterns and faint trends in financial data.
3. **Strong Performance:** The model performs well under any market condition, whether bull or bear.
4. **Improved Risk Modelling:** These measures help in identifying downturns by better risk modelling.

5. **Real-Time Processing:** Fast processing of large datasets supports real-time decisionmaking.
6. **Novel Market Insights:** Quantum-inspired measures offer innovative insights, which enhance trading strategies.

8.3 LIMITATIONS AND CHALLENGES

1. **Better prediction rate:** Quantum-inspired distance measures do enhance predictive rate in the stock market more significantly.
2. **Pattern recognition:** They could able to recognize subtle trends and complex relationships that exists in financial data.
3. **Sturdy Performance:** The model does quite well under both bull and bear market conditions.
4. **Increased Risk Modelling:** This allows it to better determine if the situation is at a risk of being bearish through modelling.
5. **In real time:** Rapid data processing accommodates timely decisions.
6. **Precious Market Information:** Quantum-inspired measures produce some innovative insight that is great for making strategies to execute trades.

8.4 FUTURE RESEARCH DIRECTIONS

1. **More Measure:** Explore other quantum-inspired approaches that can be more predictive.

2. **Scaling:** Improve algorithms to process larger datasets and complex situations, potentially using hybrid quantum-classical approaches.
3. **Extend to Other Commodities:** Apply the approach to commodities, forex, and cryptocurrencies to test its applicability.
4. **Alternative Data:** Integrate alternative data such as social media sentiment into the model to enrich the predictions.
5. **Real-Time Testing:** Conduct live implementation studies with financial institutions.
6. **Interpretability:** Develop methods to improve model explainability for better understanding by stakeholders.
7. **Comparison with Other Emerging Technologies:** Analyse performance in comparison with other advanced technologies, including deep reinforcement learning.

8. **Risk Management Applications:** Explore applications in risk assessment and portfolio optimization.

REFERENCES

1. Canabarro, A., Mendonça, T. M., Nery, R., Moreno, G., Albino, A. S., de Jesus, G. F., & Chaves, R. (2022). Quantum Finance: a tutorial on quantum computing applied to the financial market. arXiv preprint arXiv:2208.04382.
2. Stamatopoulos, N., Clader, B. D., Woerner, S., & Zeng, W. J. (2024). Quantum Risk Analysis of Financial Derivatives. arXiv preprint arXiv:2404.10088.
3. Kobayashi, N., Suimon, Y., Miyamoto, K., & Mitarai, K. (2023). The cross-sectional stock return predictions via quantum neural network and tensor network. Quantum Machine Intelligence, 5(2), 46.
4. Gujju, Y., Matsuo, A., & Raymond, R. (2024). Quantum machine learning on nearterm quantum devices: Current state of supervised and unsupervised techniques for real-world applications. Physical Review Applied, 21(6), 067001.
5. Bunescu, L., & Vârtei, A. M. (2024). Modern finance through quantum computing—A systematic literature review. PloS one, 19(7), e0304317.
6. Tychola, K. A., Kalampokas, T., & Papakostas, G. A. (2023). Quantum machine learning—an overview. Electronics, 12(11), 2379.
7. Mehendale, P. (2023). Quantum Machine Learning: The Next Frontier in AI. Journal of Scientific and Engineering Research, 10(1), 104-108.
8. Babajani, A. (2024). Quantum Computing: A Game-Changer for Libraries and Information Centers. InfoScience Trends, 1(1), 44-51.
9. IBM Institute of Business Value, Yndurain, E., Woerner, S., & Egger, D. J. (2019). Exploring quantum computing use cases for financial services. Copyright IBM Corporation.
10. Kolodziejczyk, B. (2024). Emergence of Quantum Computing Technologies in Automotive Applications: Opportunities and Future Use Cases. SAE International.

11. Auer, R., Dupont, A., Gambacorta, L., Park, J. S., Takahashi, K., & Valko, A. (2024). Quantum computing and the financial system: opportunities and risks. BIS Papers.
12. Dutta, S., Innan, N., Marchisio, A., Yahia, S. B., & Shafique, M. (2024). Qadqn: Quantum attention deep q-network for financial market prediction. arXiv preprint arXiv:2408.03088.\
13. Krishnakumar, A. (2020). Quantum Computing and Blockchain in Business: Exploring the applications, challenges, and collision of quantum computing and blockchain. Packt Publishing Ltd.
14. Wang, Y., Hu, Z., Sanders, B. C., & Kais, S. (2020). Qudits and high-dimensional quantum computing. Frontiers in Physics, 8, 589504.
15. Xu, Z., Wang, Y., Feng, X., Wang, Y., Li, Y., & Lin, H. (2024). Quantum-enhanced forecasting: Leveraging quantum gramian angular field and CNNs for stock return predictions. Finance Research Letters, 67, 105840.
16. Doosti, M., Wallden, P., Hamill, C. B., Hankache, R., Brown, O. T., & Heunen, C. (2024). A brief review of quantum machine learning for financial services. arXiv preprint arXiv:2407.12618.
17. Mironowicz, P., Mandarino, A., Yilmaz, A., & Ankenbrand, T. (2024). Applications of Quantum Machine Learning for Quantitative Finance. arXiv preprint arXiv:2405.10119.
18. Thakkar, S., Kazdaghli, S., Mathur, N., Kerenidis, I., Ferreira–Martins, A. J., & Brito, S. (2024). Improved financial forecasting via quantum machine learning. Quantum Machine Intelligence, 6(1), 27.
19. Nguma, B. F., Rao, P. V. K., & Pamain, A. (2024). Stock price prediction: a comparative analysis of classical and quantum neural networks. East African Journal of Science, Technology and Innovation, 6.
20. Saxena, A., Mancilla, J., Montalban, I., & Pere, C. (2023). Financial Modeling Using Quantum Computing: Design and manage quantum machine learning solutions for financial analysis and decision making. Packt Publishing Ltd.
21. Mishra, N., Kapil, M., Rakesh, H., Anand, A., Mishra, N., Warke, A., ... & Panigrahi, P. K. (2021). Quantum machine learning: A review and current status. Data

Management, Analytics and Innovation: Proceedings of ICDMAI 2020, Volume 2, 101-145.

22. Kerenidis, I., Prakash, A., & Szilágyi, D. (2019, October). Quantum algorithms for portfolio optimization. In Proceedings of the 1st ACM Conference on Advances in Financial Technologies (pp. 147-155).
23. Egger, D. J., Gambella, C., Marecek, J., McFaddin, S., Mevissen, M., Raymond, R., ... & Yndurain, E. (2020). Quantum computing for finance: State-of-the-art and future prospects. IEEE Transactions on Quantum Engineering, 1, 1-24.
24. Schetakis, N., Aghamalyan, D., Boguslavsky, M., Rees, A., Rakotomalala, M., & Griffin, P. R. (2024). Quantum machine learning for credit scoring. Mathematics, 12(9), 1391.
25. Bhattacharjee, S., Fuad, M. D., & Hossain, A. K. M. (2024). Classification of Financial Data Using Quantum Support Vector Machine. arXiv preprint arXiv:2412.10860.
26. Orús, R., Mugel, S., & Lizaso, E. (2019). Quantum computing for finance: Overview and prospects. Reviews in Physics, 4, 100028.
27. Liu, N., & Rebentrost, P. (2018). Quantum machine learning for quantum anomaly detection. Physical Review A, 97(4), 042315.
28. Liu, R. (2024). Monte-Carlo Simulations and Applications in Machine Learning, Option Pricing, and Quantum Processes. Highlights in Science, Engineering and Technology, 88, 1132-1137.
29. Wilkens, S., & Moorhouse, J. (2023). Quantum computing for financial risk measurement. Quantum Information Processing, 22(1), 51.
30. Iyer, R., & Bakshi, A. (2024). Artificial Intelligence and Quantum Computing Techniques for Stock Market Predictions. Deep Learning Tools for Predicting Stock Market Movements, 123-146.
31. Rebentrost, P., Gupt, B., & Bromley, T. R. (2018). Quantum computational finance: Monte Carlo pricing of financial derivatives. Physical Review A, 98(2), 022321
32. Emmanoulopoulos, D., & Dimoska, S. (2022). Quantum machine learning in finance: Time series forecasting. arXiv preprint arXiv:2202.00599.
33. Herman, D., Googin, C., Liu, X., Sun, Y., Galda, A., Safro, I., ... & Alexeev, Y. (2023).

Quantum computing for finance. Nature Reviews Physics, 5(8), 450-465.

34. Orús, R., Mugel, S., & Lizaso, E. (2019). Quantum computing for finance: Overview and prospects. Reviews in Physics, 4, 100028.
35. Martín-Guerrero, J. D., & Lamata, L. (2022). Quantum machine learning: A tutorial. Neurocomputing, 470, 457-461.
36. Gonzalez-Conde, J., RodrÍguez-Rozas, A., Solano, E., & Sanz, M. (2021). Pricing financial derivatives with exponential quantum speedup. methods, 2, 6.
37. Bhasin, N. K., Kadyan, S., Santosh, K., Ramya, H. P., Changala, R., & Bala, B. K. (2024, March). Enhancing Quantum Machine Learning Algorithms for Optimized Financial Portfolio Management. In 2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS) (pp. 1-7). IEEE.
38. Alcazar, J., Leyton-Ortega, V., & Perdomo-Ortiz, A. (2020). Classical versus quantum models in machine learning: insights from a finance application. Machine Learning: Science and Technology, 1(3), 035003.
39. Fernandes, L., Kulkarni, M., & Pande, M. B. (2023, December). A Systematic Literature Review of Classical and Quantum Machine Learning Approaches for Mutual Fund Portfolio Optimization. In 2023 IEEE Pune Section International Conference (PuneCon) (pp. 1-6). IEEE.
40. Rebentrost, P., & Lloyd, S. (2024). Quantum computational finance: quantum algorithm for portfolio optimization. KI-Künstliche Intelligenz, 1-12.
41. Gunjan, A., & Bhattacharyya, S. (2023). A brief review of portfolio optimization techniques. Artificial Intelligence Review, 56(5), 3847-3886
42. Warke, A., & Jain, V. K. (2021). Quantum Machine Learning: A Review and Current Status.