ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY
COLLEGE OF ENGINEERING
DEPARTMENT OF SOFTWARE ENGINEERING
Machine Learning Group Assignment

Regression and Classification Problems

Group Members	ID
1. Saron Abebe	ETS1440/14
2. Sefina Kamile	ETS1448/14
3. Selamawit Demissie	ETS1453/14
4. Solomon Abate	ETS 1496/14
5.Yeabsira Molalign Alefe	ETS1609/14

SUBMITTED TO: INST. Tesfaye
SUBMISSION DATE: April 8, 2025

Part-1 :Stock Price Prediction Assignment Report

1. Introduction

This report presents the analysis and prediction of Tesla's stock prices using historical data. The model employs various machine learning techniques to forecast future stock prices based on past performance.

2. Data Overview

2.1 Dataset Description

The dataset used in this analysis is a CSV file containing Tesla's stock prices from June 29, 2010, to the present. The relevant columns include:

- Date: The trading date.
- Open: Opening price of the stock.
- High: Highest price during the trading day.
- Low: Lowest price during the trading day.
- Close: Closing price of the stock.
- Adj Close: Adjusted closing price.
- Volume: Number of shares traded.

2.2 Basic Information

- Total Entries: 2457
- Data Types:
    - Float64 for stock prices and volume.
    - Object for the date.

2.3 Statistical Summary

- Mean Closing Price: 194.34
- Standard Deviation: 134.45
- Minimum Closing Price: 15.80
- Maximum Closing Price: 917.42

3. Data Preprocessing

3.1 Date Conversion

The 'Date' column was converted to datetime format and set as the index for time-series analysis.

3.2 Feature Engineering

- Moving Averages: Added 7-day and 14-day moving averages to capture trends.
- Target Variable: Created a target variable representing the next day's closing price.

3.3 Data Scaling

The dataset was scaled to a range of 0 to 1 using `MinMaxScaler` to improve the performance of the machine learning models.

4. Model Selection

4.1 Models Used

- Linear Regression: A simple linear model.
- Ridge Regression: Linear regression with L2 regularization.
- Support Vector Regression (SVR): A regression technique using support vector machines.
- Optimized SVR: Achieved through hyperparameter tuning using `GridSearchCV`.

5. Model Training and Evaluation

5.1 Train-Test Split

The data was split into training (80%) and testing (20%) sets without shuffling to maintain the time-series order.

5.2 Performance Metrics

The following metrics were calculated for each model:

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)

- R^2 Score

5.3 Results Summary

- Linear Regression:
    - MAE: 10.11
    - MSE: 360.98
    - RMSE: 19.00
    - R^2 Score: 0.98
- Ridge Regression:
    - MAE: 13.17
    - MSE: 564.16
    - RMSE: 23.75
    - R^2 Score: 0.97
- Support Vector Regression (SVR):
    - MAE: 87.42
    - MSE: 27231.03
    - RMSE: 165.02
    - R^2 Score: -0.60
- Optimized SVR:
    - MAE: 11.60
    - MSE: 458.61
    - RMSE: 21.42
    - R^2 Score: 0.97

6. Visualizations

The following plots were generated to compare actual vs. predicted stock prices for each model:

- Linear Regression

- Ridge Regression
- Support Vector Regression (SVR)
- Optimized SVR

Each plot displayed the actual stock prices against the predicted values, allowing for visual assessment of model performance.

7. Conclusion

The analysis revealed that the Linear Regression model performed the best among the tested methods, achieving a high R^2 score of 0.98 and the lowest MAE. The Optimized SVR also performed well, indicating its potential for stock price prediction with further tuning.

Part-2:Email Spam Detection Report

1. Introduction

This report presents a machine learning-based approach to classify emails as **spam** or **not spam** (also called **ham**). Using text-based features from email content, we trained and evaluated three different classification models: **Decision Tree**, **Random Forest**, and **Logistic Regression**. The primary goal is to develop a model that can accurately distinguish between spam and ham emails, which is critical for building robust email filtering systems.

2. Dataset Overview

The dataset used is `spam_ham_dataset.csv`, which contains a mix of spam and ham emails.

- **Features:**
    - `text`: The actual email content.
    - `label_num`: Binary label — `1` for spam, `0` for ham.

3. Preprocessing and Feature Extraction

- **Train-Test Split:**
  The dataset is split into 80% training and 20% testing data to ensure fair model evaluation. The training set is used to fit the models, and the testing set checks performance on unseen data.

- **Text Vectorization (TF-IDF):**
  Since machine learning models work with numerical data, we convert email text into numerical features using **TF-IDF (Term Frequency - Inverse Document Frequency)**. This technique helps highlight important words in an email while reducing the weight of commonly used words like "the", "is", etc.

4. Models Used

- **Decision Tree Classifier:**
  A tree-based model that makes decisions by splitting data based on feature values. It is simple and interpretable.

- **Random Forest Classifier:**
  An ensemble of decision trees that improves performance by averaging the results of multiple trees, reducing overfitting.

- **Logistic Regression:**
  A statistical model used for binary classification problems. It predicts the probability of a class based on input features.

5. Evaluation Metrics

- **Accuracy:** Percentage of correctly classified emails.

- **Precision:** How many predicted spam emails were actually spam.
- **Recall:** How many actual spam emails were correctly predicted.
- **F1-Score:** Harmonic mean of precision and recall, giving a balanced measure.
- **Confusion Matrix:** Shows the number of true positives, true negatives, false positives, and false negatives.

6. Results Summary

Decision Tree Classifier:

- Accuracy: **95%**
- Precision (Spam): 90%
- Recall (Spam): 92%
- F1-Score (Spam): 91%

Random Forest Classifier:

- Accuracy: **98%**
- Precision (Spam): 95%
- Recall (Spam): 98%
- F1-Score (Spam): 97%

Logistic Regression:

- Accuracy: **99%**
- Precision (Spam): 98%
- Recall (Spam): 99%

- F1-Score (Spam): 98%

7. Conclusion

Among the models tested, **Logistic Regression** performed the best in terms of accuracy, precision, and recall. **Random Forest** also delivered excellent performance and was more robust to overfitting compared to the Decision Tree.

This classification task highlights the effectiveness of using **TF-IDF for text feature extraction** and demonstrates the power of simple yet efficient models like Logistic Regression in natural language processing (NLP) tasks.

References

1. Kaggle – *Stock Market Dataset*
   https://www.kaggle.com/datasets/jacksoncrow/stock-market-dataset

2. Scikit-learn Documentation – *Machine Learning in Python*
   https://scikit-learn.org/stable/documentation.html

3. Wikipedia – *TF-IDF (Term Frequency–Inverse Document Frequency)*
   https://en.wikipedia.org/wiki/Tf–idf

4. Kaggle – *Spam Mails Dataset*
   https://www.kaggle.com/datasets/venky73/spam-mails-dataset

5. Towards Data Science – *Logistic Regression Explained*
   https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc