****

***“Artificial Intelligence”***

***“Project”***

***“Stock market Analysis”***

***Submitted by:***

*Ayesha Kanwal 80*

*Nisa Abbas 103*

*Tehreem Aziz 97*

***Program:*** *BS Computer Science*

***Semester:*** *5th*

**Project Drive link:**[*https://colab.research.google.com/drive/1rKaCS9i\_zR9Oc9anA-CbA8gRUwNPCkmz?usp=sharing*](https://colab.research.google.com/drive/1rKaCS9i_zR9Oc9anA-CbA8gRUwNPCkmz?usp=sharing)

***Dataset:***

/content/AAPL.csv

***Project description:****This code* ***performs stock price prediction*** *using two different approaches:*

***Part 1: LSTM-based Time Series Forecasting***

*This part uses a Long Short-Term Memory (LSTM) network, a type of recurrent neural network (RNN), to predict the closing price of a stock (AAPL in this case) based on historical price data.*

1. ***Data Loading and Preprocessing:***
   * *Installs necessary libraries (numpy, pandas, matplotlib, scikit-learn, tensorflow).*
   * *Loads stock data from a CSV file (presumably containing historical price data).*
   * *Converts the 'Date' column to datetime objects and sorts the data by date.*
   * *Focuses on the 'Close' price as the target variable.*
   * *Normalizes the data using MinMaxScaler to scale the prices between 0 and 1. This is crucial for LSTM networks.*
2. ***Data Splitting and Sequence Creation:***
   * *Splits the data into training and testing sets (80% train, 20% test).*
   * *Creates sequences of 60 days of historical data as input features (X) and the subsequent day's closing price as the target (y). This is how LSTMs learn temporal dependencies.*
3. ***LSTM Model Building and Training:***
   * *Builds a sequential LSTM model with two LSTM layers and two dense layers.*
   * *The first LSTM layer returns sequences, allowing the second LSTM layer to receive a sequence as input. The second LSTM layer does not return sequences.*
   * *The dense layers are fully connected layers. The final dense layer has a single output neuron, as we are predicting a single value (closing price).*
   * *Compiles the model using the Adam optimizer and mean squared error (MSE) as the loss function.*
   * *Trains the model on the training data for 10 epochs with a batch size of 32, using the test data as validation data.*
4. ***Prediction and Evaluation:***
   * *Uses the trained model to predict closing prices on the test data.*
   * *Inverse transforms the predictions and actual test values back to their original scale using the MinMaxScaler.*
   * *Plots the actual and predicted prices to visualize the model's performance.*

***Part 2: Machine Learning Classification for Stock Movement Prediction***

*This part uses several machine learning classification algorithms to predict whether the stock price will go up or down the next day.*

1. ***Data Loading and Feature Engineering:***
   * *Downloads stock data using the yfinance library.*
   * *Calculates technical indicators:* 
     + *'Price Change': Percentage change in closing price.*
     + *'5 Day MA': 5-day moving average of the closing price.*
     + *'10 Day MA': 10-day moving average of the closing price.*
   * *Removes rows with NaN values resulting from the moving average calculations.*
   * *Creates a 'Target' variable: 1 if the next day's closing price is higher than the current day's, 0 otherwise.*
   * *Drops the last row because it has no target value.*
2. ***Feature and Target Selection, Preprocessing:***
   * *Selects 'Price Change', '5 Day MA', and '10 Day MA' as features (X).*
   * *Selects the 'Target' variable as the target (y).*
   * *Scales the features using StandardScaler.*
3. ***Model Training and Evaluation:***
   * *Splits the data into training and testing sets.*
   * *Trains four classification models:* 
     + *Gaussian Naive Bayes*
     + *K-Nearest Neighbors (KNN)*
     + *Decision Tree*
     + *Random Forest*
   * *Evaluates each model using accuracy, confusion matrix, and classification report.*
4. ***Feature Importance Plot:***
   * *Plots the feature importance for the Random Forest model, showing which features have the greatest impact on the predictions.*