Problem Definition and Design thinking
STOCK PRICE PREDICTION

PROBLEM DEFINITION:

Predicting stock prices is a complex and challenging problem in applied data science and finance. The goal is to develop models that can accurately forecast the future prices of stocks or other financial assets. The task is to develop predictive models to forecast future stock prices for a given set of financial assets based on historical stock price data, contextual information, and relevant features from the years 1986 to 2020.

The Kaggle dataset context provides insights on accurate stock price predictions that are essential for investors, traders, and financial professionals to make informed decisions about buying or selling financial assets. This problem leverages data science techniques to create predictive solutions that assist in portfolio optimization, risk management, and profit maximization within the historical time frame of 1986-2020.

Data: The primary data sources include historical stock price data, trading volumes, open price, close price, economic indicators and dates, and news sentiment scores covering the period from 1986 to 2020.

Task: Creating machine learning or predictive algorithms that provide reliable estimates of future stock prices within the specified 1986–2020-time frame, with the objective of minimizing prediction errors.

Evaluation Metrics: The performance of stock price prediction models can be assessed using various metrics. This has been referred as the methods that would predict it with certain accuracy.

Mean Absolute Error (MAE): This metric measures the average absolute difference between the predicted and actual stock prices.

Mean Squared Error (MSE): This metric quantifies the average squared difference between predicted and actual prices, giving more weight to larger errors.

Root Mean Squared Error (RMSE): RMSE is the square root of the MSE, providing a measure of the average magnitude of prediction errors in the same unit as the target variable.

R-squared (R^2) Score: R-squared measures the proportion of the variance in the stock price that is explained by the model. A higher R^2 score indicates a better fit.

Feature engineering can help transform data to be more stationary, making it easier for models to capture trends and patterns because raw stock price data alone may not be sufficient to make accurate predictions.

Challenges specific to this time frame include dealing with long-term historical data, changes in market over time, and the need to account for economic events that occurred within the 1986-2020 period.

Stock price prediction is a complex problem in applied data science, with numerous challenges and approaches. It requires a combination of domain knowledge, data analysis, feature engineering, and machine learning techniques to build accurate and effective predictive models.

DESIGN THINKING:

Applied Data science plays a pivotal role in stock price prediction by providing the tools and techniques to analyse, model, and make decisions based on historical data and relevant information. Here, we have designed an approach to make this prediction on stock price accurately using the Kaggle dataset covering the years 1986 to 2020.

https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dataset

We have provided a step-by-step guide based on our knowledge that we obtained from this course so far and the provided dataset.

Step 1: Data Acquisition (provided dataset)

The dataset obtained from Kaggle, includes historical stock price data, dates, trading volumes, open price, close price and economic indicators and make sure the dataset covers the period from 1986 to 2020.

Step 2: Data Exploration and Understanding

To Load the dataset into a data analysis tool (e.g., Python with Pandas, Google Colab or visual studio code) and explore its structure and contents. We will use google colab and visual studio code based on the dependency of our project needs and preferences.

Google Colab is an excellent choice for data science and machine learning tasks that require GPU acceleration and collaborative work in Jupyter notebooks, especially when working on cloud-based usages.

Visual Studio Code, on the other hand, is a versatile code editor suitable for a wide range of programming tasks, offering extensive customization and offline development capabilities.

So, our choice varies based on our preferences.

Check for **missing values, outliers, and data types**.

Depending on the nature of the data and the analysis, we will perform data imputation to replace missing values with meaningful estimates, ensuring that valuable information is not lost. Some of them include:

- **Mean/Median Imputation**
- **Interpolation Methods (Time Series Data)**
- **K-Nearest Neighbours (KNN) Imputation**
- **Forward Fill and Backward Fill (Time Series Data)**

The choice of imputation method for handling missing values depends on several factors, like the nature of the dataset and the problem definition of our project.

There can be exceptionally high or low values that deviate from the general pattern or distribution of the data and hence identifying outliers is important. Outliers can distort data visualizations, making it challenging to interpret plots and graphs accurately, so we will use visualization techniques like box plots, scatter plots, and histograms to visually identify potential outliers and then Transform Data and then convert date columns to datetime objects for time series analysis.

Step 4: Splitting the Data

We will divide the dataset into training and testing sets. For time series data, we'll use a time-based split ensuring that the training set contains data from earlier years and the testing set contains data from later years within the dataset.

Step 5: Feature Engineering

To improve the performance of machine learning models (LSTM) by Normalizing or scaling numerical features. Using One-hot encoding converts categorical variables into binary columns to represent each category.

Step 6: Model Selection and Building

We will gradually explore complex models like time series models (e.g., ARIMA for the historical dataset) and machine learning models (LSTM) Long Short-Term Memory networks.

Step 7: Model Evaluation

To Evaluate model performance on the testing data using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2) Score and create visualizations to compare predicted prices with actual prices to see accuracy.

Step 8: Continuous Learning and Monitoring

To implement mechanisms for continuous learning, updating models with new data, and monitoring model performance over time.

Step 8: Hyperparameter Tuning

As we are using machine learning models, performing hyperparameter tuning techniques like grid search will be used. To generate evaluation reports, visualizations, and confusion matrices is used to provide insights into the model's strengths and weaknesses.

This step-by-step detail provides a comprehensive approach to stock price prediction using the Kaggle dataset spanning from 1986 to 2020 based on our understanding to solving the problem statement.