**INTERN PROJECT PHASE – 1**

**Data Science Projects: Predictive Modeling**

**Project 1: Stock Market Prediction**

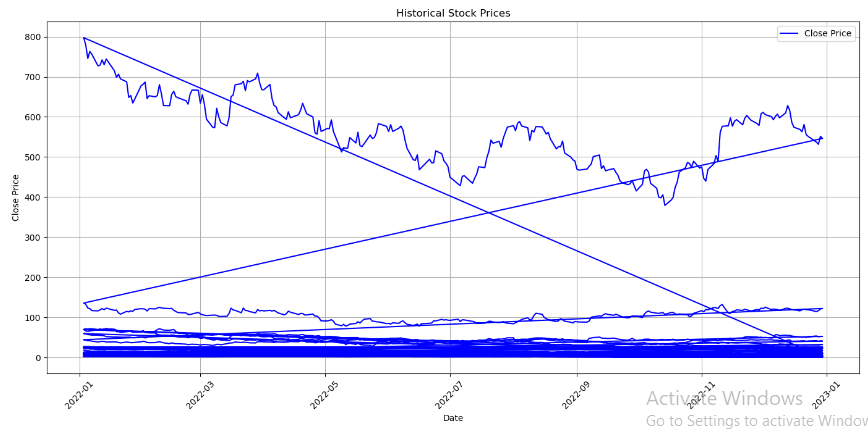
<https://www.kaggle.com/datasets/luisandresgarcia/stock-market-prediction>

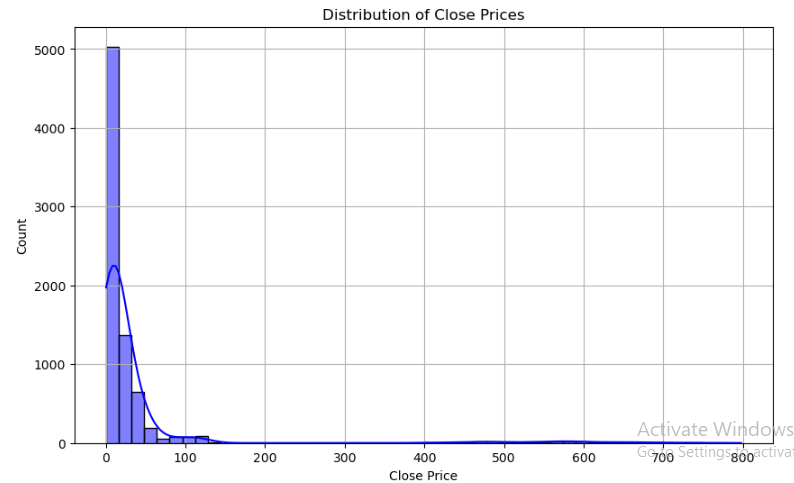
**Introduction:**

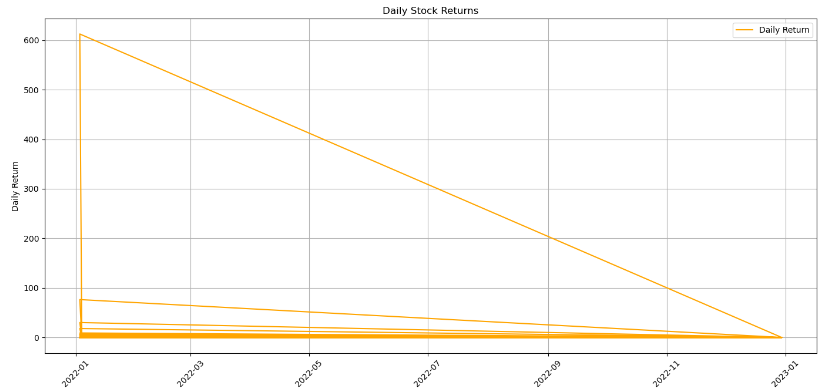
The aim of this project is to predict stock prices using historical data and machine learning techniques. The dataset used for this project is sourced from Kaggle and contains various stock market data, including columns like Date, Open, High, Low, Close, and Volume. These columns represent different aspects of stock prices and trading volumes on specific dates. The project is divided into two main parts: Exploratory Data Analysis (EDA) and Predictive Modeling.

**Exploratory Data Analysis (EDA):**

* The first step involved loading the dataset and converting the 'Date' column to a datetime format, ensuring that the data is ready for time series analysis.
* The 'Date' column was then set as the index, allowing for easy visualization of trends over time.
* A line plot of the 'Close' prices was generated, which provided a clear view of how the stock prices fluctuated over the observed period. This plot was useful for identifying trends, volatility, and periods of stability within the data.
* Additionally, a correlation heatmap was created using only numeric data from the dataset. This heatmap highlighted the relationships between different numeric features, such as the correlation between the 'Open' and 'Close' prices, which helps in understanding how different factors move together in the stock market.

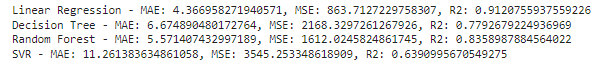
****

****

****

**Predictive Modeling:**

* The XGBoost regression model was selected for this task due to its efficiency, speed, and high performance in handling structured data.
* Before training the model, the dataset was prepared by separating the features (independent variables) from the target variable (dependent variable), which in this case was the 'Close' price. Categorical variables, if any, were converted to numeric using one-hot encoding to ensure compatibility with the machine learning model.
* The dataset was then split into training and testing sets to evaluate the model's performance on unseen data. The XGBoost model was trained on the training set, and predictions were made on the test set.
* The performance of the model was evaluated using the Mean Squared Error (MSE), a common metric for regression tasks that measures the average squared difference between the predicted and actual values.

****

****

****

**Conclusion:**

This project successfully demonstrated the process of performing EDA and predictive modeling on a stock market dataset. Through EDA, key insights were gained regarding stock price trends and relationships between different market variables. The predictive modeling phase utilized XGBoost to forecast future stock prices, and the model's performance was quantified using MSE. For future work, further improvements could be made by engineering additional features, such as moving averages or technical indicators, and by conducting hyperparameter tuning to optimize the model. Additionally, comparing XGBoost's performance with other models, such as LSTM networks, could provide deeper insights into the best approaches for stock price prediction.

Source Code: <https://github.com/msubhikshaT/Data-Science-Project-Series/blob/main/1%20-%20Stock%20Market%20Prediction.ipynb>

**Project 2: Breast Cancer Prediction**

<https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data>

**Introduction:**

The objective of this project is to predict whether a tumor is malignant or benign based on the Breast Cancer Wisconsin (Diagnostic) dataset. This involves several steps including data preprocessing, feature selection, and implementing a Support Vector Machine (SVM) model.

**Data Preprocessing:**

* The dataset was loaded using pandas and checked for missing values.
* Missing values were handled using the mean strategy to ensure no data was lost.
* Columns that were not useful for model training, such as 'id' and 'Unnamed: 32', were removed.
* The target variable 'diagnosis' was encoded to numerical values, with 'M' mapped to 1 and 'B' mapped to 0. The data was split into features (X) and target (y).
* Outliers were detected using the z-score method and removed if they exceeded a threshold of 3. This was to ensure that the data used for training was clean and did not contain extreme values that could skew the model's performance.
* Features were then normalized using the StandardScaler to standardize the data and make the model training more efficient. The dataset was split into training and testing sets with a ratio of 80:20.

**Feature Selection and Engineering:**

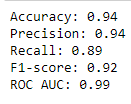
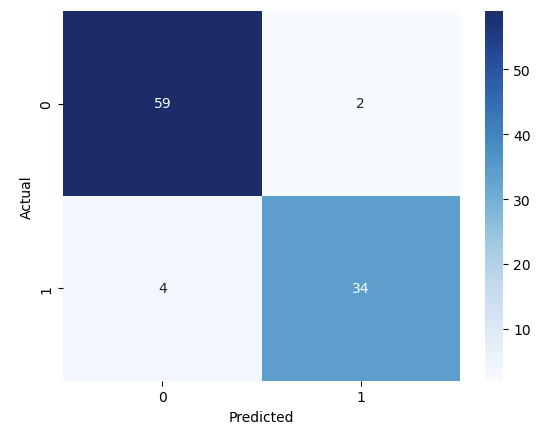
* A correlation matrix was used to identify important features. Features with a high correlation to the target variable were selected based on a threshold of 0.8.
* This step ensured that only the most relevant features were used for model training, improving the efficiency and performance of the model.
* The selected features were then normalized, and the data was split again into training and testing sets.

**Machine Learning Model (SVM):**

* A Support Vector Machine (SVM) model with a linear kernel was implemented. The model was trained on the selected and scaled features.
* Predictions were made on the test set, and various performance metrics were calculated to evaluate the model's performance. These metrics included accuracy, precision, recall, F1-score, and ROC AUC.
* The accuracy measures the proportion of correct predictions, while precision indicates the accuracy of the positive predictions.
* Recall measures the model's ability to detect positive samples, and the F1-score provides a balance between precision and recall.
* The ROC AUC indicates the model's ability to discriminate between positive and negative classes.

**Results and Evaluation:**

* The SVM model achieved a satisfactory performance based on the calculated metrics.
* The confusion matrix was plotted to visualize the number of true positives, true negatives, false positives, and false negatives.

****

**Challenges:**

Several challenges were faced during the project. Handling missing values and outliers effectively without losing significant data was crucial. Ensuring that the selected features contributed positively to the model's performance was also a key challenge.

In conclusion, this project successfully implemented a Support Vector Machine model to predict whether a tumor is malignant or benign. The steps taken in data preprocessing, feature selection, and model training were documented, and the model's performance was evaluated using various metrics.

Source Code: <https://github.com/msubhikshaT/Data-Science-Project-Series/blob/main/2%20-%20Breast%20Cancer%20Prediction.ipynb>