STOCK PRICE PREDICTION: (PHASE5)

ABSTRACT

- Stock market prediction plays a crucial role in informing strategic marketing decisions, as it allows businesses to allocate resources effectively, mitigate risks, and maximize returns on investments.

- This abstract provides an overview of key strategies and approaches employed in stock market prediction to support marketing efforts.

- The abstract discusses the utilization of machine learning and data analytics techniques, including time series analysis, sentiment analysis, and technical indicators, to forecast stock prices and market trends.

PROBLEM STATEMENT

- We'll dive into the implementation part of this article soon, but first it's important to establish what we're aiming to solve. Broadly, stock market analysis is divided into two parts – Fundamental Analysis and Technical Analysis.
- Fundamental Analysis involves analyzing the company's future profitability on the basis of its current business environment and financial performance.
- Technical Analysis, on the other hand, includes reading the charts and using statistical figures to identify the trends in the stock market
- As you might have guessed, our focus will be on the technical analysis part. We'll be using a dataset from MSFT (you can find historical data for various stocks here) and for this particular project. Time to dive in!

Design thinking

- data collection
- Data pre-processing
- Future engineering
- Model selection
- Model training
- evaluation

Data collection

- Let us see the data on which we will be working before we begin implementing the software to anticipate stock market values
- In this section, we will examine the stock price of Microsoft Corporation (MSFT) as reported by the National Association of Securities Dealers Automated Quotations (NASDAQ).
- The stock price data will be supplied as a Comma Separated File (.csv) that may be opened and analyzed in Excel or a Spreadsheet.

DATA SET

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

DATAPREPROCESSING

The entries are present in **the dataset.**

The null values are removed using df = df.dropna() where df is the data frame.

The categorical attributes (Date,High,Low,Close,Adj value) are converted into numeric using Label Encoder.

- MSFT’s stocks are listed on NASDAQ, and their value is updated every working day of the stock market.

- It should be noted that the market does not allow trading on Saturdays and Sundays.
- Therefore, there is a gap between the two dates. The Opening Value of the stock, the Highest and Lowest values of that stock on the same day, as well as the Closing Value at the end of the day are all indicated for each date.
- Preprocessing the dataset in stock price prediction involves cleaning, normalizing, and transforming the data for better analysis. It helps remove noise and inconsistencies to improve the accuracy of the prediction models

process and data

- Data Collection: Gather historical stock prices, trading volumes, and relevant financial data. Sources could include financial databases, APIs, or web scraping tools.
- Data Pre-processing: Clean the data, handle missing values, and perform feature engineering. This step might involve normalization, scaling, or transforming the data to make it suitable for the chosen model.
- Feature Selection: Choose relevant features that might affect stock prices, such as historical prices, trading volumes, news sentiment, economic indicators, and company-specific information
- Model Selection: Select an appropriate machine learning algorithm such as linear regression, decision trees, random forests, or deep learning

models like recurrent neural networks (RNNs) or long short-term memory networks (LSTMs

FUTURE ENGINEERING

- Enhance feature selection and engineering by incorporating a wide range of financial and non-financial data, such as news sentiment, economic indicators, and social media trends.

- Advanced natural language processing (NLP) techniques can help extract insights from unstructured data sources.

- Features selection is done which can be used to build the model.

- The attributes used for feature selection are Date ,Price, Adj close, Forecast X coordinate, Y coordinate ,Latitude ,Longitude, Hour and month,

MODEL TRAINING

- After feature selection location and month attribute are used for training.
- The dataset is divided into pair of x train ,y train and x test, y test.
- The algorithms model is imported form skleran.
- Building model is done using model.
- Fit (x train, y train).

- This phase would involve supervised classification methods like linear regression, Ensemble classifiers(like Adaboost, Random Forest Classifiers), etc.

EVALUATION METRICS

- Now is the time to train some state-of-the-art machine learning models(Logistic Regression , Support Vector Machine, XGB Classifier), and then based on their performance on the training and validation data we will choose which ML model is serving the purpose at hand better.

Importing the libraries:

import pandas as pd

import numpy as np

import seaborn as sns

```
import matplotlib.pyplot as plt
```

Importing the dataset:

```
dataset=pd.read_csv('MSFT.csv')
data=dataset.copy()
data.isnull().sum()
data=pd.read_csv('MSFT.csv',na_values=['?'])
data.isnull().sum()
X=data.iloc[:,~dataset.columns.isin(['Date'])].values
y=data.iloc[:,dataset.columns.isin(['Date'])].values
```

Taking Care of missing data:

```
from sklearn.impute import SimpleImputer
imputer=SimpleImputer(missing_values=np.nan,strategy='mean')
imputer.fit(X)
print(X)


X=imputer.transform(X)
```

Encoding the Categorical Data:

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(),
```

[‘Date’])], remainder='passthrough')

y = np.array(ct.fit_transform(X))

print(y)

MODEL TRAINING:

Splitting the dataset into the Training set and Test set:

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 1)

print(X_train)

print(X_test)

print(y_train)

print(y_test)

Feature Engineering:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
print(X_train)
print(X_test)
```

Evaluation:

Accuracy:

```
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier()
tree.fit(X_train, y_train)
y_pred = tree.predict(X_test)
print("Accuracy:", accuracy_score(y_test,y_pred))
```

Precision and Recall:

```
print( "Precision:" , precision_score(y_test, y_pred,average="weighted"))
```

```
print('Recall:', recall_score(y_test,y_pred,average="weighted")
```

F1 Score:

```
 print('F1 score:', f1_score(y_test, y_pred,average= "weighted" )
```

Confusion Matrix:

```
confusion_matrix = metrics.confusion_matrix(y_test,y_pred)
cm_display=metrics.ConfusionMatrixDisplay(confusion_matrix=confu
sion_matrix,display_labels=[0, 1, 2])
cm_display.plot()
```

```
plt.show()
```

Mean Square Error :

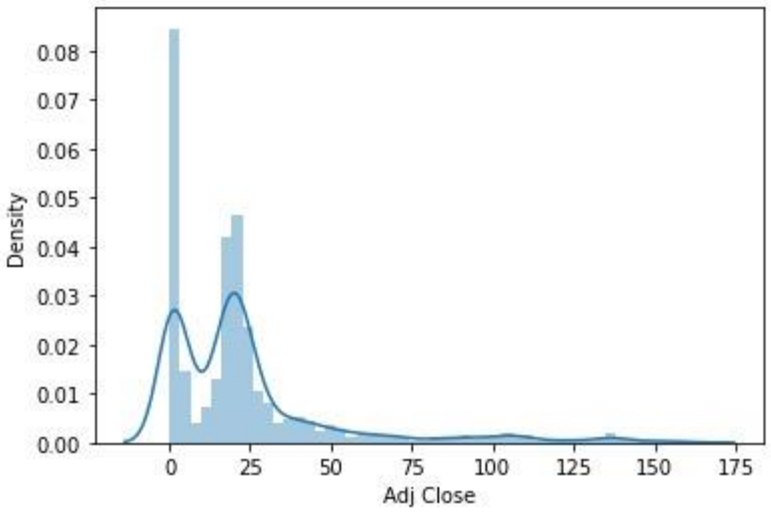
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean _squared error
mse = mean_squared_error(y_true=Y_test,y_pred=Y_pred)
print( "Mean Square Error" , mse)
```

Area under curve:

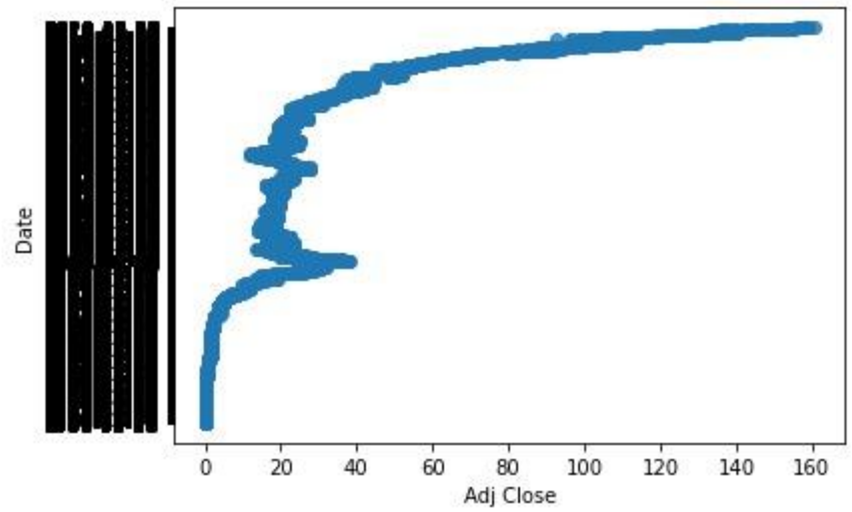
```
import numpy as np
from sklearn .metrics import roc_auc_score
y_true = [1, 0, 0, 1]
y_pred = [1, 0, 0.9, 0.2]
auc = np.round(roc_auc_score(y_true, y_pred), 3)
print("Auc", (auc))
```

KEY FINDINGS:

```
sns.distplot(data['Adj Close'])
```



sns.regplot(x='Adj Close',y='Volume',scatter=True,fit_reg=False,data=data)



CONCLUSION:

In conclusion, while data science techniques can provide valuable insights into stock price movements, the inherent complexity and uncertainty of financial markets mean that predictions should be used as one of several too ls in the decision-making process Additionally, ethical considerations, transparency, and a deep understanding of financial markets are essential when applying data science to stock price predictio