STOCK PRICE PREDICTOR

Submitted By:
Saima Siddiqui
Shreya

Project Overview
Investment firms, hedge funds and even individuals have been using financial models to better understand market behavior and make profitable investments and trades. A wealth of information is available in the form of historical stock prices and company performance data, suitable for machine learning algorithms to process.

Can we actually predict stock prices with machine learning?
Investors make educated guesses by analyzing data. They'll read the news, study the company history, industry trends and other lots of data points that go into making a prediction. The prevailing theories is that stock prices are totally random and unpredictable.

Problem Statement
The challenge of this project is to accurately predict the future closing value of a given stock across a given period of time in the future. For this project I will use a **Linear Regression and Random Forest Regression** to predict the closing [1] price using a dataset of past prices

GOALS
1. Explore stock prices.
2. Implement basic model using linear regression and Random Forest Regression.
3. Compare the results and submit the report.

Metrics
For this project measure of performance will be using score.

Data Exploration
My goal was to predict the closing price for any given date after training. The prediction has to be made for Closing (Adjusted closing) price of the data.

Code
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
dataset=pd.read_csv('Stock_Market.csv')
dataset.index=dataset['Date']
dataset.shape
```

Out: (4392, 7)

```
X=dataset.iloc[:,[1,2,3,5,6]]
Y=dataset.iloc[:,4]
```

```
X.head()
```

Out:

Date	Open	High	Low	Adj_Close	Volume
2000/3/27	3.812500	4.156250	3.812500	4.125000	3675600
2000/3/28	4.125000	4.125000	4.000000	4.015625	1077600
2000/3/29	4.000000	4.031250	3.953125	4.000000	437200

2000/3/30	4.000000	4.000000	3.843750	3.843750	1883600
2000/3/31	3.734375	3.734375	3.390625	3.390625	7931600

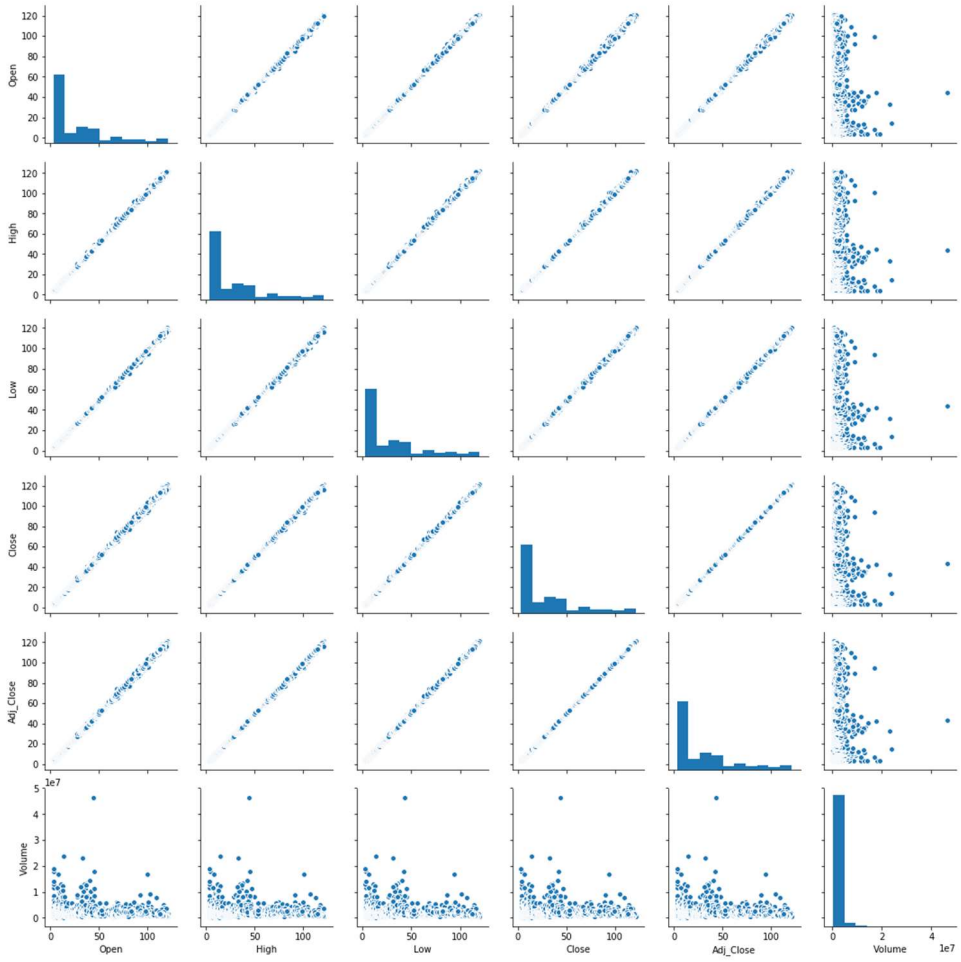
```
d=dataset.drop(["Date"],axis=1)
d.head()
```

Out:

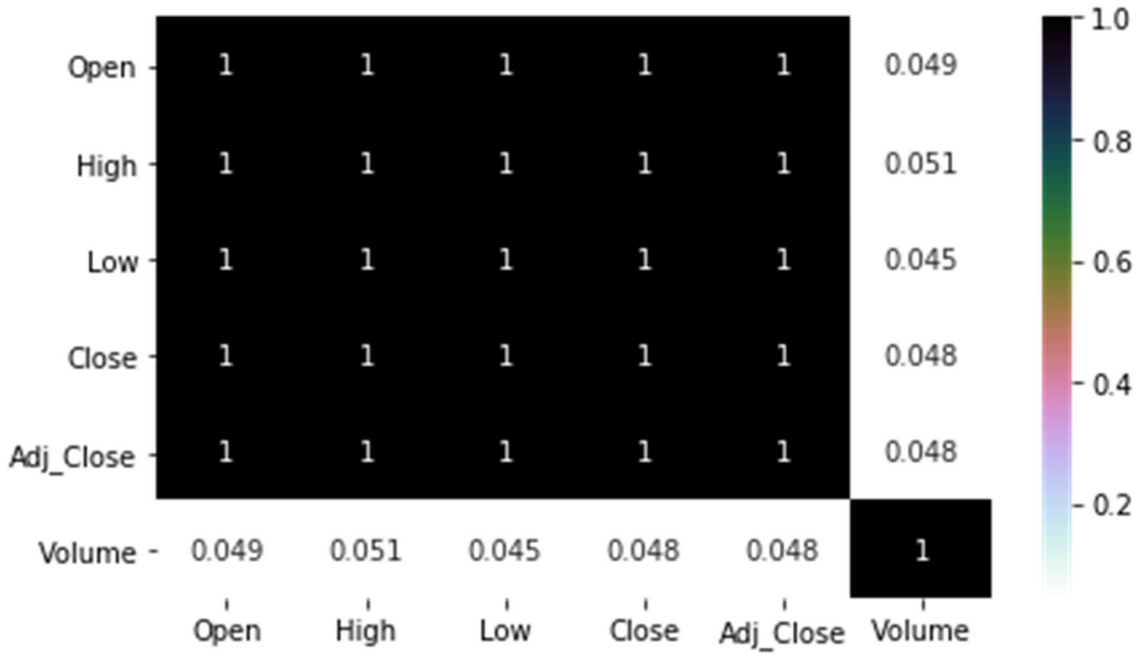
Date	Open	High	Low	Close	Adj_Close	Volume
2000/3/27	3.812500	4.156250	3.812500	4.125000	4.125000	3675600
2000/3/28	4.125000	4.125000	4.000000	4.015625	4.015625	1077600
2000/3/29	4.000000	4.031250	3.953125	4.000000	4.000000	437200
2000/3/30	4.000000	4.000000	3.843750	3.843750	3.843750	1883600
2000/3/31	3.734375	3.734375	3.390625	3.390625	3.390625	7931600

```
import seaborn as sns
sns.pairplot(d)
```

<seaborn.axisgrid.PairGrid at 0x1898ae46390>



```
#correlation of features with each other
plt.figure(figsize=(7,4))
sns.heatmap(d.corr(),annot=True,cmap='cubehelix_r')
```




```
Y.head()
```

```
Date
2000/3/27    4.125000
2000/3/28    4.015625
2000/3/29    4.000000
2000/3/30    3.843750
2000/3/31    3.390625
Name: Close, dtype: float64
```

```
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=0)

from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(X_train,Y_train)
Y_pred=regressor.predict(X_test)

regressor.score(X_test,Y_test)
```

```
1.0
```

```
X_new=[[3.45,3.98,3.76,4.43,2163600]]
d=pd.DataFrame(X_new)

result=regressor.predict(X_new)
result
```

```
array([4.43])
```

```
X_new=[[3.8125,4.15625,3.8125,4.125,3675600]]
d=pd.DataFrame(X_new)
result=regressor.predict(X_new)
result
```

```
array([4.125])
```

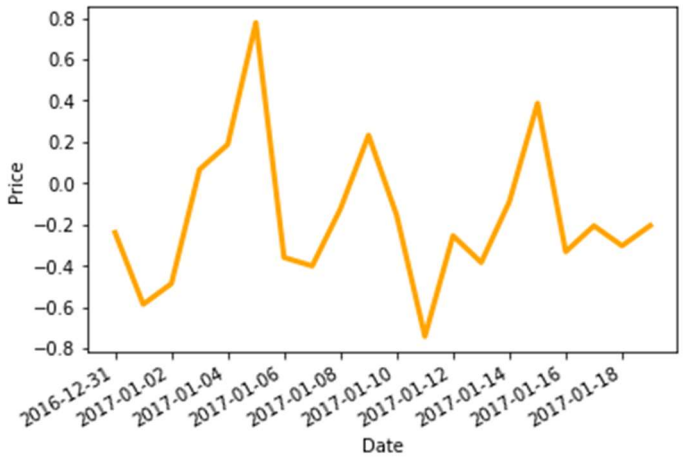
```
Y_test.head()
```

```
Date
2016/5/2     106.860001
2005/4/13     10.630000
2003/3/4       6.630000
2007/4/19     13.000000
2007/5/17     12.297500
Name: Close, dtype: float64
```

```
from sklearn.ensemble import RandomForestRegressor
regressor1=RandomForestRegressor(n_estimators=10,random_state=0)
regressor1.fit(X_train,Y_train)
Y_pred=regressor1.predict(X_test)
regressor1.score(X_test,Y_test)
```

```
0.9999902087205729
```

```
dataset['Close'].plot(color='orange', linewidth=3)
plt.xlabel('Date')
plt.ylabel('Price')
plt.show()
```



Conclusion

We predicted the close value of stock using linear regression and random forest regression and calculated the accuracy for used model.
The close value should be nearly equal to the adjusted close.

Linear Model Accuracy: 100%
Random Forest Accuracy: 99%