Stock Price Prediction Using Machine Learning

By

21781A33C4 (R Pavithra)

21781A33D0 (S Soniya)

21781A33D1 (Sanjana K)

21781A33E9 (V Naga Shanthi)

Guided by

Prof. Ms Raasha

A Dissertation Submitted to

SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY, An Autonomous Institution affiliated to 'JNTU Ananthapur' in Partial Fulfilment of the Bachelor of Technology (***Computer Engineering***) with Specialization in ***Artificial Intelligence and Machine Learning.***

May 2024

SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY

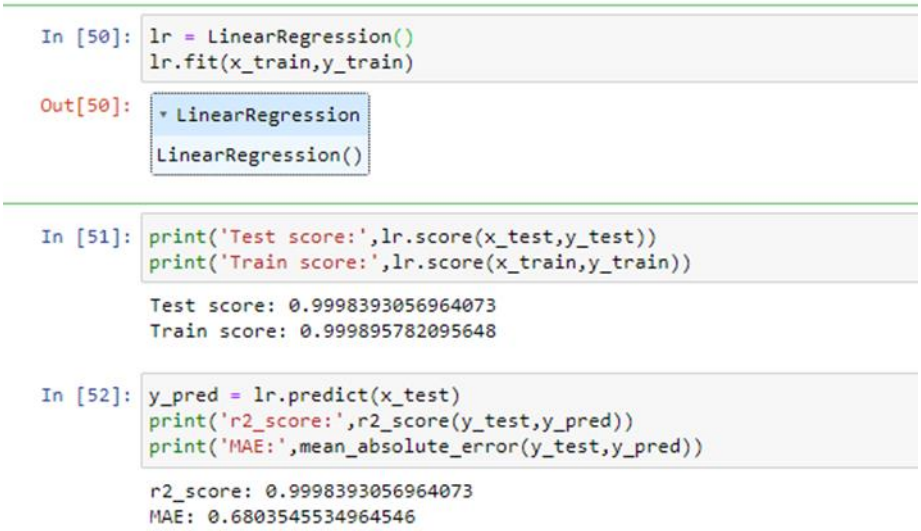
R.V.S. Nagar Tirupathi Road, Andhra Pradesh– 517127

Model Building

Now our data is cleaned and it's time to build the model. We can train our data on different algorithms. For this project we are applying seven classification algorithms. The best model is saved based on its performance.

Linear Regression Model

First we are going to initialise the LinearRegression() model and training data is passed to the model with.fit() function. Test data is forecasted/predicted with predict() function and saved in a new variable. For evaluating the model, train score, test score, r2_score and MAE scores are used.



Decision Tree Regressor

First we are going to initialise the DecisionTreeRegressor () model and training data is passed to the model with.fit() function. Test data is forecasted/predicted with predict() function and saved in a new variable. For evaluating the model, t

Decision Treee

```
In [38]: dt = DecisionTreeRegressor()
         dt.fit(x_train,y_train)

Out[38]: DecisionTreeRegressor
         DecisionTreeRegressor()

In [39]: print('Test score:',dt.score(x_test,y_test))
         print('Train score:',dt.score(x_train,y_train))

         Test score: 0.9995536298964616
         Train score: 1.0

In [40]: y_pred = dt.predict(x_test)
         print('r2_score:',r2_score(y_test,y_pred))
         print('MAE:',mean_absolute_error(y_test,y_pred))

         r2_score: 0.9995536298964616
         MAE: 1.0284022981651377
```

Extra Trees Regressor

First we are going to initialise the ExtraTreeRegressor () model and training data is passed to the model with.fit() function. Test data is forecasted/predicted with predict() function and saved in a new variable. For evaluating the model, train score, test score, r2_score and MAE scores are used.

Extra Trees Regression

```
In [44]: etr = ExtraTreeRegressor()
         etr.fit(x_train,y_train)

Out[44]: ExtraTreeRegressor
         ExtraTreeRegressor()

In [45]: print('Test score:',etr.score(x_test,y_test))
         print('Train score:',etr.score(x_train,y_train))

         Test score: 0.9994142211412407
         Train score: 1.0

In [46]: y_pred = etr.predict(x_test)
         print('r2_score:',r2_score(y_test,y_pred))
         print('MAE:',mean_absolute_error(y_test,y_pred))

         r2_score: 0.9994142211412407
         MAE: 1.1654668512742097
```

Random Forest Regressor

First we are going to initialise the RandomForestRegressor () model and training data is passed to the model with.fit() function. Test data is forecasted/predicted with predict() function and saved in a new variable. For evaluating the model, train score, test score, r2_score and MAE scores are used.

Random Forest Regression

```
In [128]:
rf = RandomForestRegressor()
rf.fit(x_train,y_train)

Out[128]:
RandomForestRegressor
RandomForestRegressor()

In [129]:
print('Test score:',rf.score(x_test,y_test))
print('Train score:',rf.score(x_train,y_train))

Test score: 0.9998013249033134
Train score: 0.9999740655897109

In [130]:
y_pred = rf.predict(x_test)
print('r2_score:',r2_score(y_test,y_pred))
print('MAE:',mean_absolute_error(y_test,y_pred))

r2_score: 0.9998013249033134
MAE: 0.712384695570204
```

Model Comparison And Evaluating Best Model

From the observed r2 scores and Mena absolute errors we can clearly see that the linear regression model has least Mean absolute error among others. So we are going to select Linear Regression model for final Flask application deployment.

To plot the predictions over the original values of all the companies we can use the following code.

First, we are going to store the dates, original closing prices of stocks and predicted closing prices of stocks as shown below. We are going to access company specific rows of test_data and x_test variables using the “Company” column.

```
amd_dates = test_data[test_data['Company']==0]['Date']
amd_pred = lr.predict(x_test[x_test['Company']==0])
amd_orig = test_data[test_data['Company']==0]['Close']

asus_dates = test_data[test_data['Company']==1]['Date']
asus_pred = lr.predict(x_test[x_test['Company']==1])
asus_orig = test_data[test_data['Company']==1]['Close']

intel_dates = test_data[test_data['Company']==2]['Date']
intel_pred = lr.predict(x_test[x_test['Company']==2])
intel_orig = test_data[test_data['Company']==2]['Close']

msi_dates = test_data[test_data['Company']==3]['Date']
msi_pred = lr.predict(x_test[x_test['Company']==3])
msi_orig = test_data[test_data['Company']==3]['Close']

nvidia_dates = test_data[test_data['Company']==4]['Date']
nvidia_pred = lr.predict(x_test[x_test['Company']==4])
nvidia_orig = test_data[test_data['Company']==4]['Close']
```

Now using these columns we are going to plot the data as shown

As we can see the lines plotted by the predicted and original data are almost perfectly overlapping.

```
# Plot predictions and actual values
plt.figure(figsize=(15,8))


sns.lineplot(amd_dates,amd_orig)
sns.lineplot(amd_dates,amd_pred)

sns.lineplot(asus_dates,asus_orig)
sns.lineplot(asus_dates,asus_pred)

sns.lineplot(intel_dates,intel_orig)
sns.lineplot(intel_dates,intel_pred)

sns.lineplot(msi_dates,msi_orig)
sns.lineplot(msi_dates,msi_pred)

sns.lineplot(nvidia_dates,nvidia_orig)
sns.lineplot(nvidia_dates,nvidia_pred)

plt.legend(['AMD Original','AMD Predicted','ASUS Original','ASUS Predicted','INTEL Original',
            'INTEL Predicted','MSI Original','MSI Predicted','NVIDIA Original','NVIDIA Predicted'])
plt.show()
```

The model which is used for this project is Linear Regression as it is evaluated as the best suitable model for estimating the stock price for the top 5 GPU companies according to the results after comparison of the models.