Project Development Phase – Sprint 4

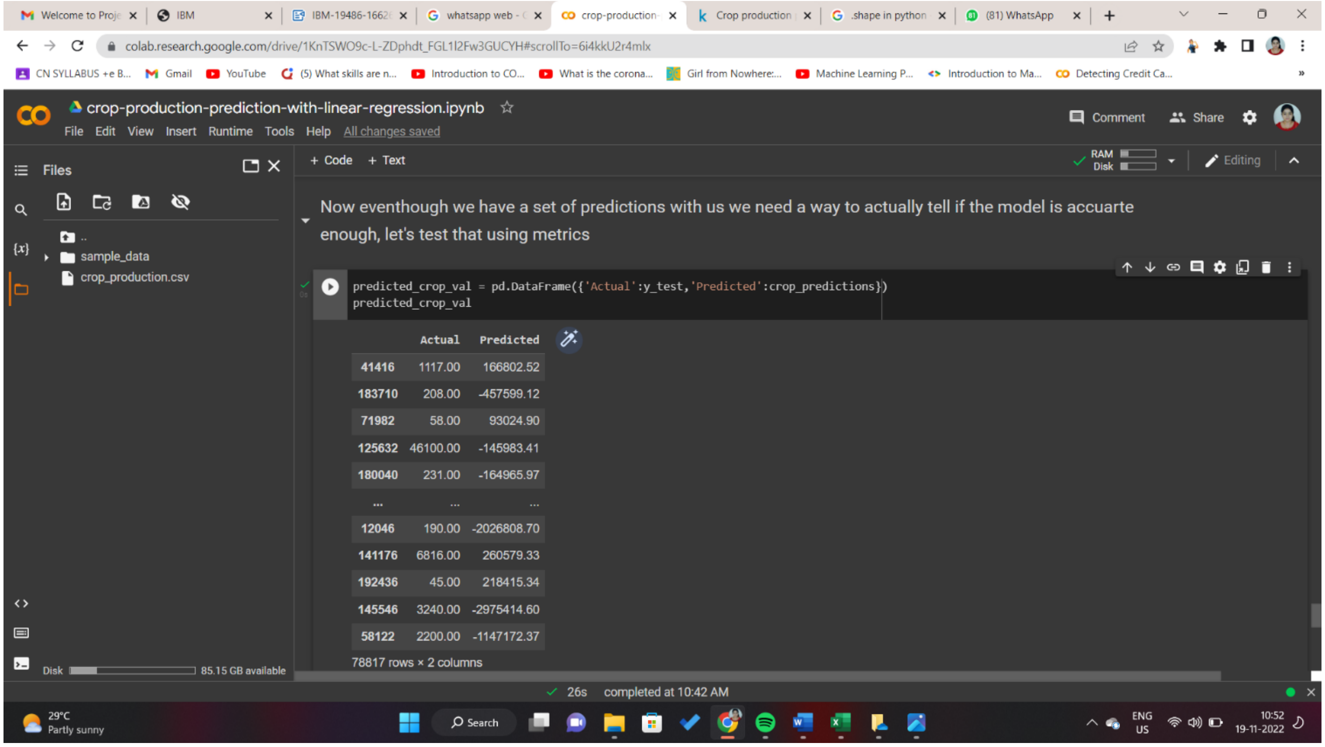
DASHBOARD

Date	19 November 2022
Team ID	PNT2022TMID27968
Project Name	Estimate The Crop Yield Using Data Analytics

TESTING THE ML MODEL AND OUTPUTS

Using python prediction analysis we have tested the accuracy of the predicted data by verifying the test and train data.

Testing the data:-

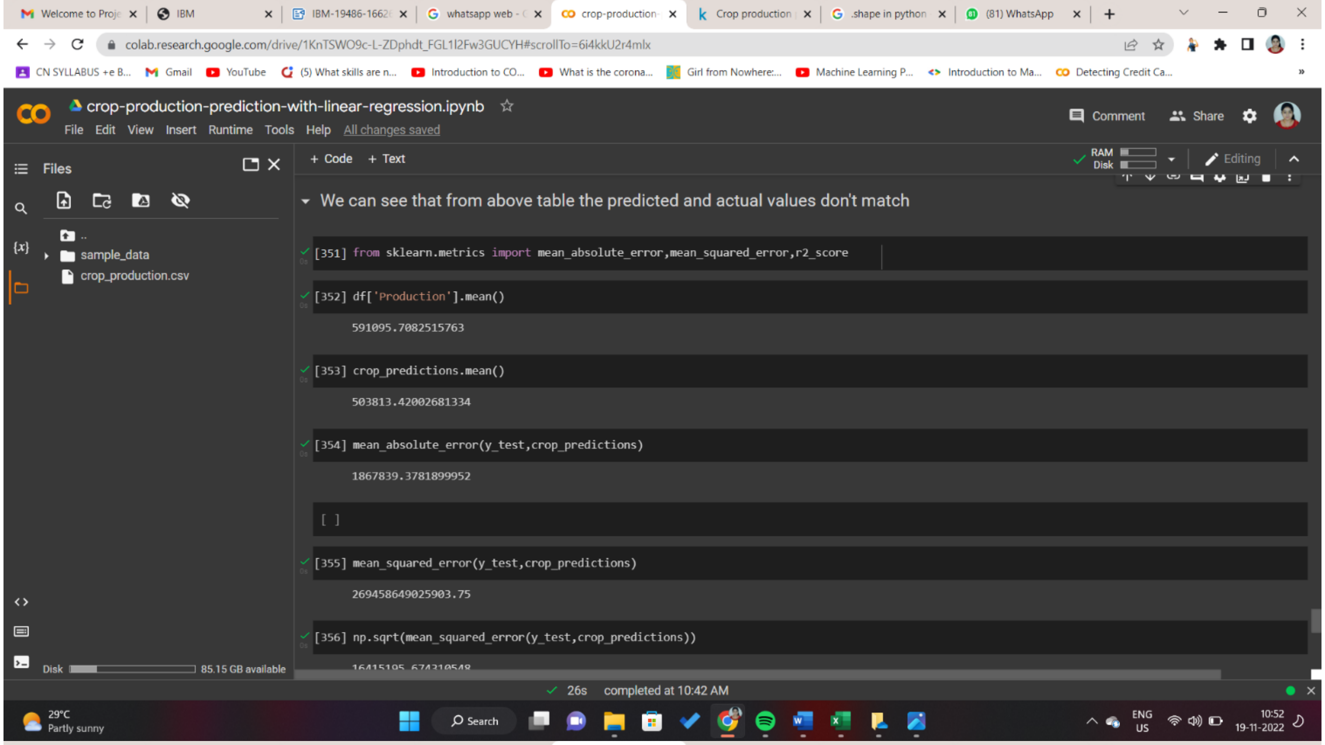


Now eventhough we have a set of predictions with us we need a way to actually tell if the model is accuarte enough, let's test that using metrics

```
predicted_crop_val = pd.DataFrame({'Actual':y_test,'Predicted':crop_predictions})
predicted_crop_val
```

```
          Actual    Predicted
41416    1117.00    166802.52
183710    208.00   -457599.12
71982      58.00     93024.90
125632  46100.00   -145983.41
180040    231.00   -164965.97
...          ...          ...
12046     190.00  -2026808.70
141176   6816.00    260579.33
192436     45.00    218415.34
145546   3240.00  -2975414.60
58122    2200.00  -1147172.37
78817 rows × 2 columns
```

✓ 26s completed at 10:42 AM


crop-production-prediction-with-linear-regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Files
sample_data
crop_production.csv
Disk 85.15 GB available
26s completed at 10:42 AM

We can see that from above table the predicted and actual values don't match

```
[351] from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
```

```
[352] df['Production'].mean()
```

```
591095.7082515763
```

```
[353] crop_predictions.mean()
```

```
503813.42002681334
```

```
[354] mean_absolute_error(y_test,crop_predictions)
```

```
1867839.3781899952
```

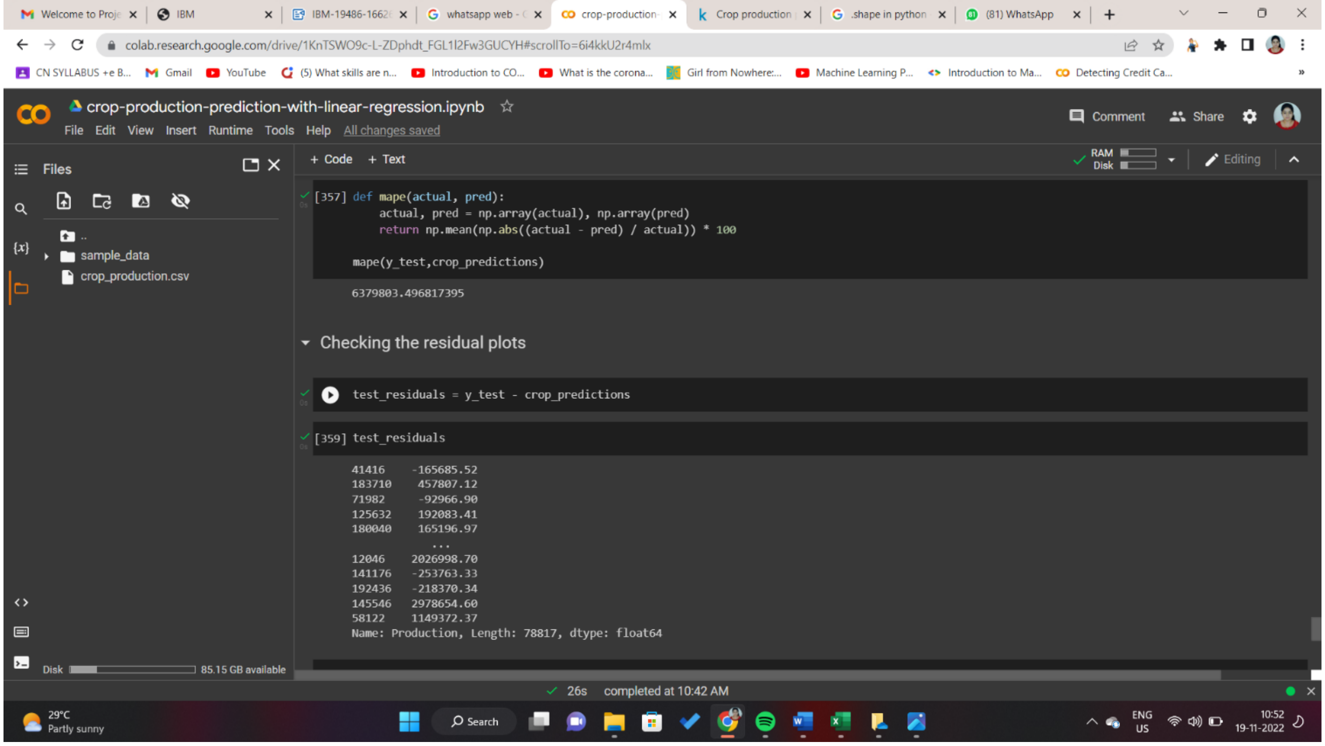
```
[ ]
```

```
[355] mean_squared_error(y_test,crop_predictions)
```

```
269458649025903.75
```

```
[356] np.sqrt(mean_squared_error(y_test,crop_predictions))
```

```
16415195.674310548
```


crop-production-prediction-with-linear-regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Files
sample_data
crop_production.csv
Disk 85.15 GB available
26s completed at 10:42 AM

```
[357] def mape(actual, pred):
          actual, pred = np.array(actual), np.array(pred)
          return np.mean(np.abs((actual - pred) / actual)) * 100

      mape(y_test,crop_predictions)
```

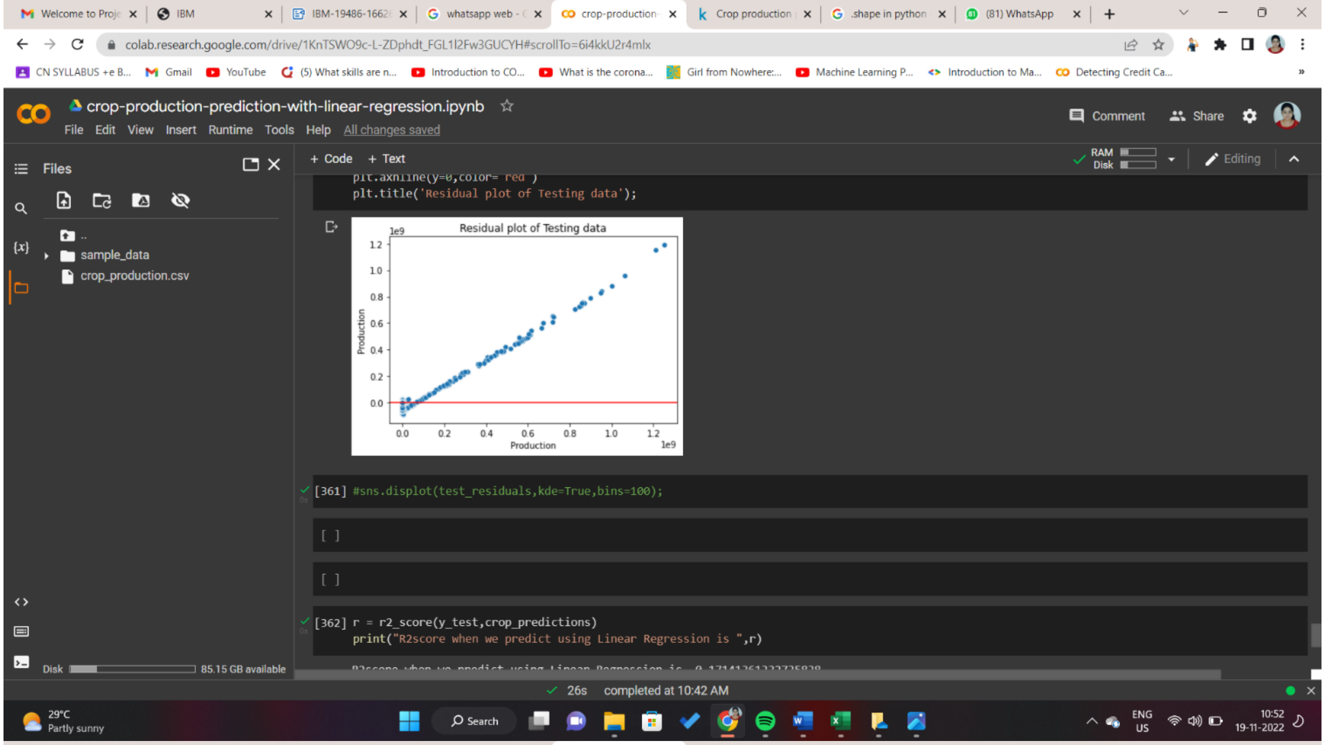
```
6379803.496817395
```

Checking the residual plots

```
test_residuals = y_test - crop_predictions
```

```
[359] test_residuals
```

```
41416     -165685.52
183710     457807.12
71982      -92966.90
125632     192083.41
180040     165196.97
             ...
12046     2026998.70
141176    -253763.33
192436    -218370.34
145546    2978654.60
58122     1149372.37
Name: Production, Length: 78817, dtype: float64
```



crop-production-prediction-with-linear-regression.ipynb

```
plt.axhline(y=0,color='red')
plt.title('Residual plot of Testing data');
```

```
[361] #sns.displot(test_residuals,kde=True,bins=100);
```

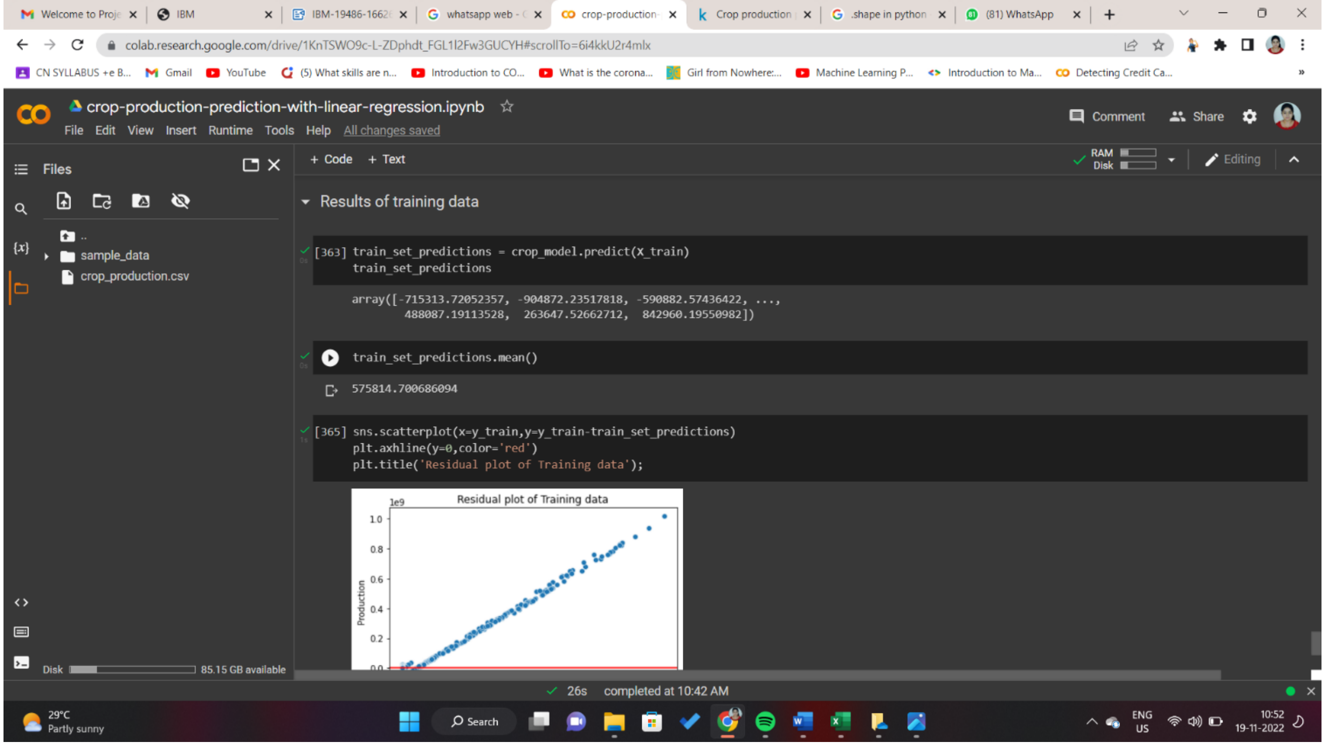
```
[ ]
```

```
[ ]
```

```
[362] r = r2_score(y_test,crop_predictions)
      print("R2score when we predict using Linear Regression is ",r)
```

```
R2score when we predict using Linear Regression is  0.17141261323725828
```

✓ 26s completed at 10:42 AM



crop-production-prediction-with-linear-regression.ipynb

Results of training data

```
[363] train_set_predictions = crop_model.predict(X_train)
      train_set_predictions
```

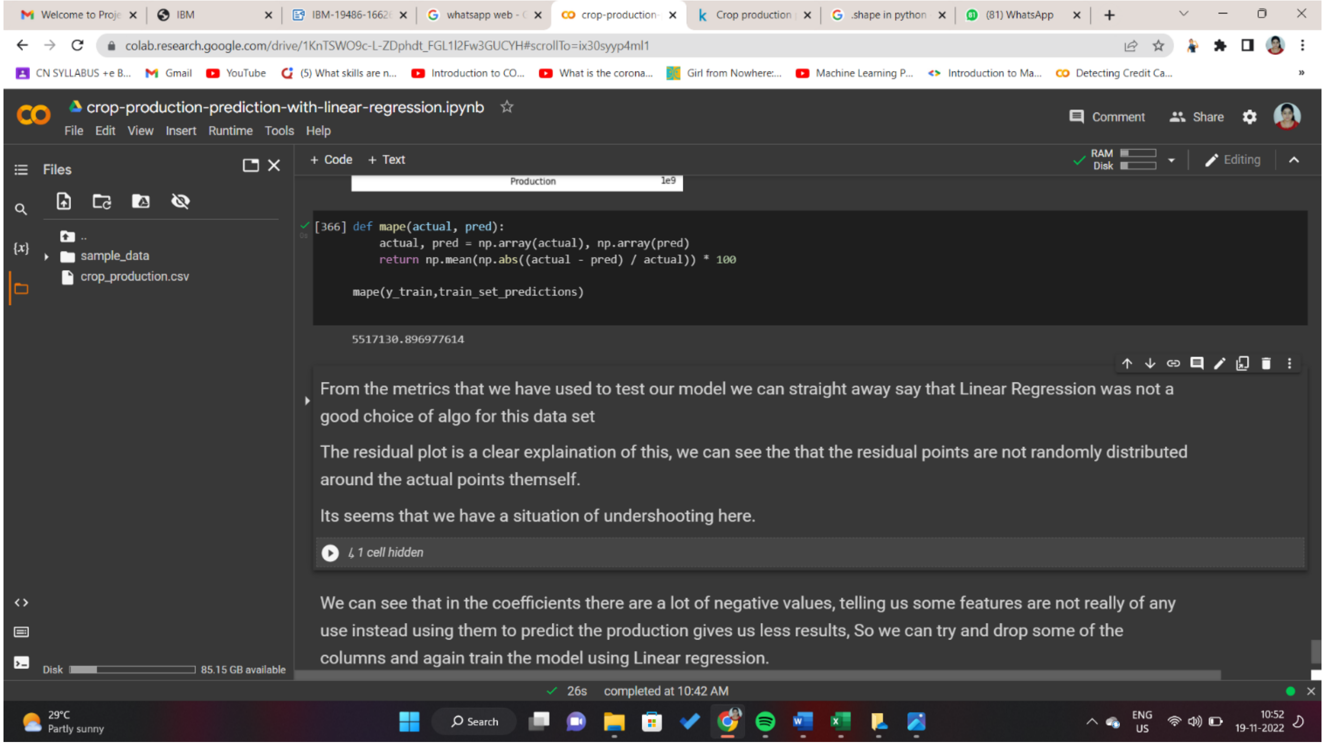
```
array([-715313.72052357, -904872.23517818, -590882.57436422, ...,
        488087.19113528,  263647.52662712,  842960.19550982])
```

```
train_set_predictions.mean()
```

```
575814.700686094
```

```
[365] sns.scatterplot(x=y_train,y=y_train-train_set_predictions)
      plt.axhline(y=0,color='red')
      plt.title('Residual plot of Training data');
```

✓ 26s completed at 10:42 AM


crop-production-prediction-with-linear-regression.ipynb
File Edit View Insert Runtime Tools Help
Files
sample_data
crop_production.csv
+ Code + Text
Comment Share
Editing
Production 1e9
1 cell hidden
26s completed at 10:42 AM

```
[366] def mape(actual, pred):
          actual, pred = np.array(actual), np.array(pred)
          return np.mean(np.abs((actual - pred) / actual)) * 100

      mape(y_train,train_set_predictions)
```

```
5517130.896977614
```

From the metrics that we have used to test our model we can straight away say that Linear Regression was not a good choice of algo for this data set

The residual plot is a clear explaination of this, we can see the that the residual points are not randomly distributed around the actual points themself.

Its seems that we have a situation of undershooting here.

We can see that in the coefficients there are a lot of negative values, telling us some features are not really of any use instead using them to predict the production gives us less results, So we can try and drop some of the columns and again train the model using Linear regression.

Conclusion:

From the metrics that we have used to test our model we can straight away say that Linear Regression was not a good choice of algo for this data set. The residual plot is a clear explanation of this, we can see the that the residual points are not randomly distributed around the actual points themself. Its seems that we have a situation of undershooting here.
Hence linear regression may not be an efficient model for the prediction of crop yield.