MULTI-TARGET REGRESSION

INTRODUCTION

When multiple dependent variables exist in a regression model, this task is called as multi-target regression. regressor is employed to learn the mapping from input features to output variables jointly. In this study, multi- implemented for quality prediction in a mining process to estimate the amount of silica and iron concentrate process.

In this study, two inter-dependent single target regression tasks are transformed into a multiple output regres in a mining process.

In the pervious models have been conducted to estimate silica concentrate with or without taking iron conce aspect, the problem is a single-target regression problem. However, this study that focuses on the estimation concentrates simultaneously as output variables. We compared different multi-target regressors that use Ra ,RIDGE and Decision Tree algorithms separately in the background. Coefficient of determination (R2) metric a predictive performance of the regression methods for the mentioned data.

METHODS TO IMPLEMENT MTR

Problem transformation methods

1. These methods are mainly based on transforming the multi-output regression problem into single-targe for each target, and finally concatenating all the d predictions. The main drawback of these methods is targets are ignored, and the targets are predicted independently, which may affect the overall quality of
2. **Regressor chains (RC) method**

   It is inspired by the recent multi-label chain classifiers 31. RC is another problem transformation metho single-target models. The training of RC consists of selecting a random chain (i.e., permutation) of the s building a separate regression model for each target following the order of the selected chain.
3. **Single traget model**

   output variables are estimated independently and potential relations between them cannot be exploitec

RELATED WORKS

https://ieeexplore.ieee.org/abstract/document/8907120Y

The paper focus on inherent multiregressor models and concluded to it is best to predict silica and iron conc

NEW METHODS

https://machinelearningmastery.com/multi-output-regression-models-with-python/

My work focus on following implementation:

1. To see whether the %silica concentrate can be predicted without iron concentrate and result showed us
   concentrate withou iron concentrate . Hence, to solve the problem we can implement the multitarget re
   target variables at same time.
2. To try diffrent models which is not inherent multitarget regression models like Randomforest,Ridge,Xgb
3. To finalize the best model with R2 as well as MSE metric.

LIBARY

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.metrics import r2_score
from sklearn.ensemble import AdaBoostRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.multioutput import MultiOutputRegressor
from sklearn.metrics import mean_squared_error
import joblib
import sklearn.preprocessing import StandardScalar
```

FUNCTION-1

PRE-PROCESSING

```
1. MISSING VALUE
2. NULL VALUE
3. CHANGING INTO CORRECT FORMAT
4. SCALING
```

CALCULATION

ONE HR = 3600 SECS

SAMPLES AT 20 SECONDS

LET US HAVE ONE RECORD AT END OF 20 SECS

SO, 3600/20 = 180

WHICH MEANS WE GET 180 RECORDS FOR ONE HOUR SO WE NEED TO WE FIND THE NUMBER OF RECORD
LUCKY TO COCNLUDE THERE IS NO MISSING VALUE ELSE WE NEED TO FILL THOSE MISSIN VALUES

WE DONT TREAT THE CORRELATION FEATURES SINCE OUR TASK
CORRELATION ON FEATURES WE DONT ELIMINATE THEM SIMPLY

WE TREAT THEM WHILE DOING MODELING OF DATA

FEATURE ENGINEERING

Rounding

Often when dealing with continuous numeric attributes like proportions or percentages, we may not need the of precision. Hence it often makes sense to round off these high precision percentages into numeric integers proper percentages

2017-03-10 01:00:00	55,2	16,98	3019,53
2017-03-10 01:00:00	55,2	16,98	3024,41
2017-03-10 01:00:00	55,2	16,98	3043,46
2017-03-10 01:00:00	55,2	16,98	3047,36
2017-03-10 01:00:00	55,2	16,98	3033,69

THE RAW DATA IS SHOWN ABOVE

THE ROUND OFF DATA IS SHOWN BELOW

	index	datetime hours	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	Flotation Column 03 Air Flow	Flotation Column 04 Air Flow	Flotation Column 05 Air Flow	Flota Colum Air
0	2017-03-10 01:02:00	2017-03-10 01:00:00	55.2	16.98	3019.53	557.434	395.713	10.0664	1.74	249.214	253.235	250.576	295.096	306.4	25
1	2017-03-10 01:02:20	2017-03-10 01:00:00	55.2	16.98	3024.41	563.965	397.383	10.0672	1.74	249.719	250.532	250.862	295.096	306.4	25

REFERENCES

1. assignment donors dataset-preprocessing
2. https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/

   https://www.kaggle.com/juejuewang/handle-missing-values-in-time-series-for-beginners
3. https://machinelearningmastery.com/multi-output-regression-models-with-python/


```
def final_fun_1(X):
  d1=[]
  if X.shape==(25,):
     X1=X.reshape(-1,1)
     x=np.frompyfunc(lambda x: x.replace(',','.'),1,1)(X1[2:25]).astype(float)

     unique, counts = np.unique(x, return_counts=True)
     d1.append(unique)
     unique, counts = np.unique(d1, return_counts=True)
     d=dict(zip(unique, counts))
     for key, value in d.items():
         if value==180 and  np.isnan(x):
```

```
            x.remove(x)
      features_x = scale_features_std.fit_transform(x[:21].reshape(1,-1))
      from_joblib = joblib.load('/content/drive/My Drive/ADABOOST-MTR.pkl')
      y_pred=from_joblib.predict(features_x)
      return y_pred
   else:

      x=np.frompyfunc(lambda x: x.replace(',','.'),1,1)(X[:,2:23]).astype(float)
      for i in range(len(x)):
            unique, counts = np.unique(x[i][0], return_counts=True)
            d1.append(unique)
      unique, counts = np.unique(d1, return_counts=True)
      d=dict(zip(unique, counts))
      for key, value in d.items():
          if value==180 and  np.isnan(x[i]):
            x.remove(x[i])
      features_x = scale_features_std.fit_transform(x[:,:21])
      from_joblib = joblib.load('/content/drive/My Drive/ADABOOST-MTR.pkl')
      y_pred=from_joblib.predict(features_x)
      return y_pred
```

FUNTION-2

Finding R2_METRIC AND MSE

```
def final_fun_2(X,y):
  d1=[]
  if X.shape==(25,):
     X1=X.reshape(-1,1)
     x=np.frompyfunc(lambda x: x.replace(',','.'),1,1)(X1[2:25]).astype(float)
     y=np.frompyfunc(lambda x: x.replace(',','.'),1,1)(X1[23:25]).astype(float)
     unique, counts = np.unique(x, return_counts=True)
     d1.append(unique)
     unique, counts = np.unique(d1, return_counts=True)
     d=dict(zip(unique, counts))
     for key, value in d.items():
         if value==180 and  np.isnan(x):
           x.remove(x)
     features_x = scale_features_std.fit_transform(x[:21].reshape(1,-1))
     from_joblib = joblib.load('/content/drive/My Drive/ADABOOST-MTR.pkl')
     y_pred=from_joblib.predict(features_x)
     y_pred=y_pred.reshape(-1,1)
     r2_metric=r2_score(y,y_pred)
     mse=mean_squared_error(y,y_pred)
     return r2_metric,mse
  else:
     y=np.frompyfunc(lambda x: x.replace(',','.'),1,1)(y).astype(float)
     x=np.frompyfunc(lambda x: x.replace(',','.'),1,1)(X[:,2:23]).astype(float)
     for i in range(len(x)):
           unique, counts = np.unique(x[i][0], return_counts=True)
           d1.append(unique)
     unique, counts = np.unique(d1, return_counts=True)
     d=dict(zip(unique, counts))
     for key, value in d.items():
         if value==180 and  np.isnan(x[i]):
           x.remove(x[i])
```

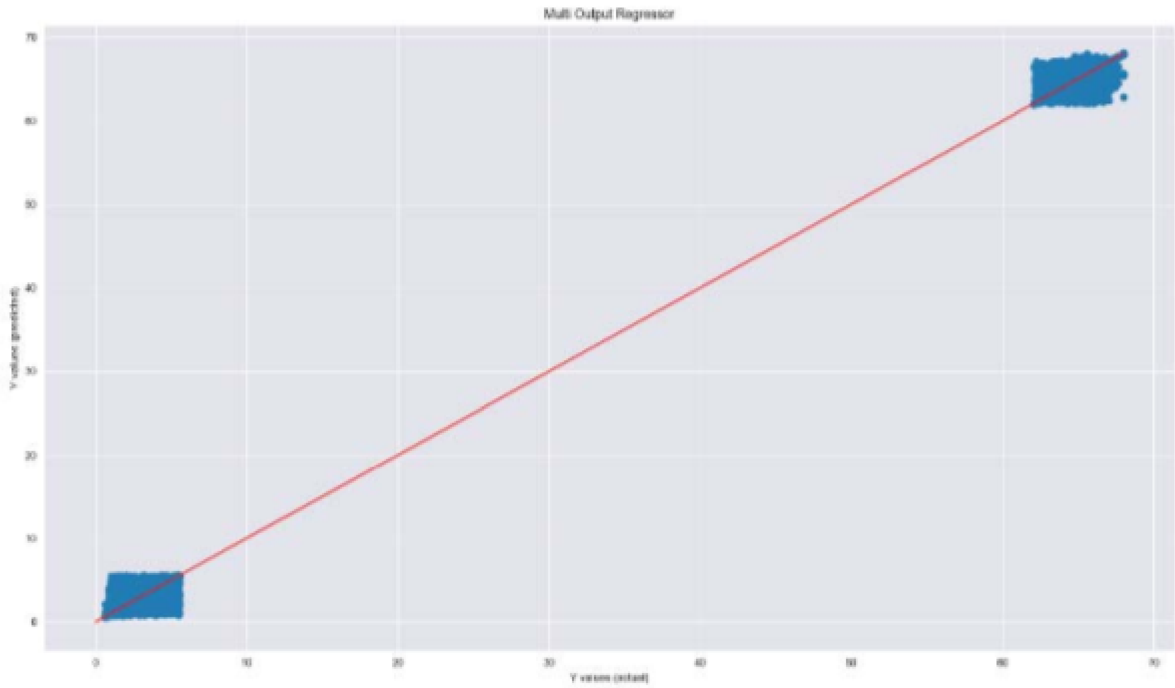
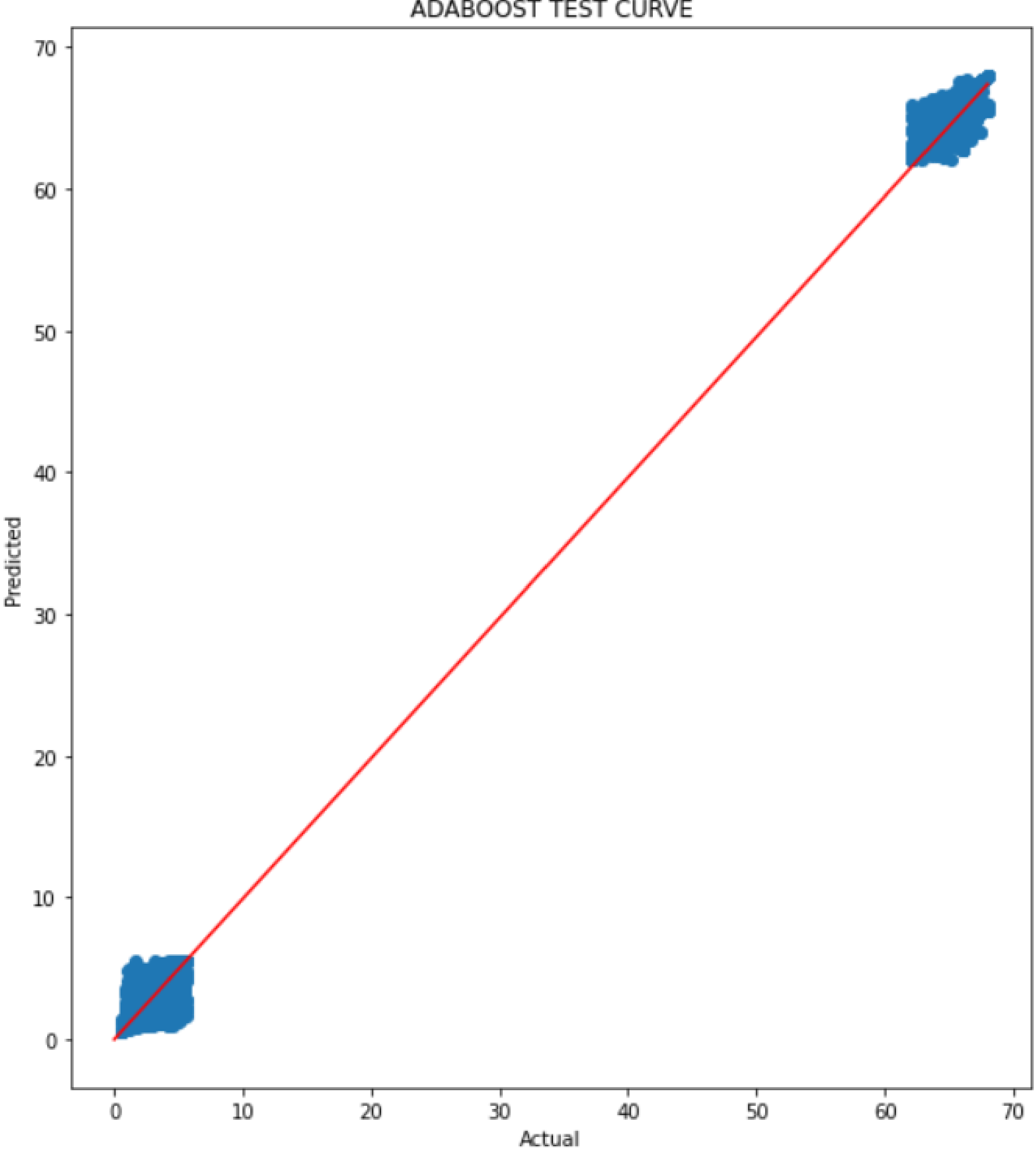


Fig. 4. Scatter plot of the model that predicts two target variables: silica and iron concentrates by multi-output regressor

PLOTS OBTAINED



SUMMARY

The experimental results show that AdaBoost regressor clearly provided higher coefficient of determination v
probably because AdaBoost is an ensemble method, which generally provides better accuracies than an indiv
decisions of several predictors. In addition, AdaBoost is an iterative algorithm, each time reweighting the inst
next classifier on incorrectly classified ones. By this way, it constructs a strong classifier from a combination

Our Results match with research paper results . In reerach paper also ADABOOST is the best model likewise i
ADABOOST.

The experimental results demonstrate the superiority of AdaBoost .

In this study, a multi-target regression problem is handled to predict quality in a mining process. The aim is to
simultaneously estimates the amount of silica and iron concentrates in the ore. Several approaches are imple

```
    features_x = scale_features_std.fit_transform(X[:,:21])
    from_joblib = joblib.load('/content/drive/My Drive/ADABOOST-MTR.pkl')
    y_pred=from_joblib.predict(features_x)
    r2_metric=r2_score(y,y_pred)
    mse=mean_squared_error(y,y_pred)
    return r2_metric,mse
```

```
df=pd.read_csv('/content/drive/My Drive/prep_time')
X=df.values
```

PREDICTION

```
y_pred=final_fun_1(X)
```

EVALUATION-METRIC

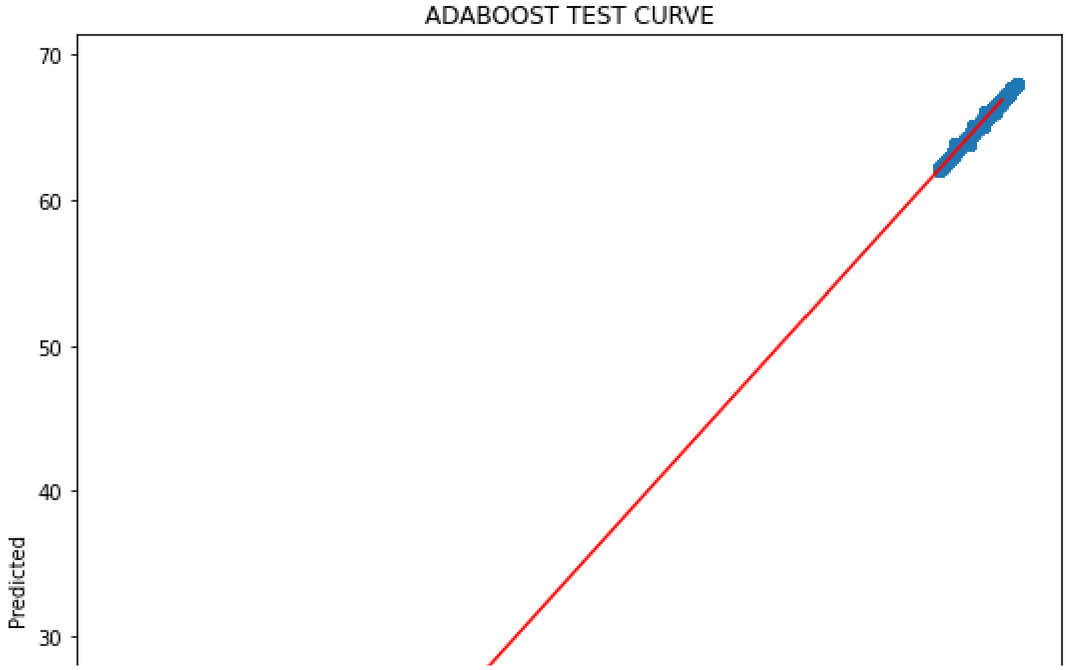
```
r2_score,mse=final_fun_2(X,X[:,23:25])
```

```
print("R2 metric is",r2_score)
print("*"*100)
print("MSE value is",mse)
```

```
R2 metric is 0.970523270885995
***********************************************************************************************
MSE value is 0.03709564508317323
```

PLOTS

```
#https://www.kaggle.com/plbescond/quality-prediction-r-0-81-mse-0-12
fig = plt.figure(figsize=(30, 10))
ax = fig.add_subplot(131)
ax.set(title="ADABOOST TEST CURVE", xlabel="Actual", ylabel="Predicted")
ax.scatter(y1,a)
ax.plot([0,max(y1[0])], [0,max(a[0])], color='r')
fig.show()
```



METRIC ANALYSIS

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.
highlight=r2#sklearn.metrics.r2_score

https://scikit-
learn.org/stable/modules/generated/sklearn.metrics.mean_square

R2 SCORE

Acoording to literature, the r2 score is good when it is closer to 1 and it can be negative (because the model c
constant model that always predicts the expected value of y, disregarding the input features, would get a R^2

TRAIN R2 is so closer to 0 and TEST r2 is also to 0 and hence inorder to get better result , we must try other r

MSE VALUE

A non-negative floating point value (the best value is 0.0), or an array of floating point values, one for each inc

The MSE value and R2 score value is better for the model.

PLOTS ANALYSIS

1. MTR MODEL

   In curve, the points tend to overlay on the line and in both iron and silica the points are overfalling on reg

   In curve, the models are able to predict iron concetrate and are able to predict the silica better in the sar

CONCLUSION

to handle more than one target variable. We tried to observe the performance of a multi target regression app
highly correlated. At the end, it is noticed that this approach can also be efficient in manufacturing data when
the algorithm as an input parameter. Instead, that feature can also be evaluated as an output variable by bein
feature. We have observed that this alteration did not create an adverse effect on the regression performance

AdaBoost model can be used for quality prediction. It shows the scatter plot of the model that predicts two ta

OVERALL COMPARSION

```
#https://pypi.org/project/PrettyTable/-- for representation of data
#https://stackoverflow.com/questions/36423259/how-to-use-pretty-table-in-python-to-print-out-data-fr
from prettytable import PrettyTable
x = PrettyTable(border=True, header=True, padding_width=15)
x.field_names = ["MODEL1", "R2","MSE"]
x.add_row(["ADABOOST",970523270885995,0.03203487819099303])
print(x)
```

```
+------------------------------------+------------------------------------------+---------
|               MODEL1               |                    R2                    |
+------------------------------------+------------------------------------------+---------
|              ADABOOST              |             970523270885995              |
+------------------------------------+------------------------------------------+---------
```

RESULTS IN RESERACH PAPER VS OUR PAPER

AdaBoost Regressor	0.98

OUR RESULTS

```
+-----------------------------------+-----------------------------------------+------------------
|               MODEL1              |                   R2                    |
+-----------------------------------+-----------------------------------------+------------------
|              ADABOOST             |             970523270885995             |              0.0
+-----------------------------------+-----------------------------------------+------------------
```

PLOTS IN PAPER VS OUR RESULTS(ADABOOST ALONE)

```
            **PLOTS IIN PAPER**
```