algorithms_underperformed

December 12, 2021

[ ]:

```
#importing core libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

#importing essential libraries
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn import metrics
#importing Machine learning libraries
from sklearn.svm import SVR
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import SGDRegressor
```

[ ]:

```
#importing and reading dataset
data = pd.read_csv('student-perf.csv')
data
```

[ ]:

```
      sex  age  address  famsize  medu  fedu  mjob  fjob  guardian  \
0       2   23        1        1     4     4     1     4         1
1       2   22        1        1     1     1     1     5         2
2       2   20        1        2     1     1     1     5         1
3       2   20        1        1     4     2     2     3         1
4       2   21        1        1     3     3     5     5         2
…   …  …      …      …   …   …   …   …       …
7699    2   24        2        1     2     3     3     5         1
7700    2   23        1        2     3     1     4     3         1
7701    2   23        1        1     1     1     5     5         1
7702    1   22        1        2     3     1     3     3         1
7703    1   23        2        2     3     2     3     5         1

      traveltime  …  paid  activities  internet  romantic  famrel  freetime  \
0              2  …     2           2         2         2       4         3
1              1  …     2           2         1         2       5         3
2              1  …     2           2         1         2       4         3
3              1  …     2           1         1         1       3         2
```

```
4              1  …     2           2         2         2       4         3
…          …  …   …         …       …       …     …       …
7699           1  …     2           1         1         2       5         4
7700           1  …     2           2         1         2       4         3
7701           2  …     2           1         2         2       1         1
7702           2  …     2           2         1         2       2         4
7703           3  …     2           2         1         2       4         4

      goout  health  absences  GPA
0         4       3         4   11
1         3       3         2   11
2         2       3         6   13
3         2       5         0   14
4         2       5         0   13
…     …     …       …  …
7699      2       5         4   11
7700      4       1         4   15
7701      1       5         6   12
7702      5       2         6   10
7703      1       5         4   11

[7704 rows x 24 columns]
```

[ ]:

```
#Declaration of dependent variable
X=data[['sex', 'age', 'address', 'famsize', 'medu', 'fedu', 'mjob', 'fjob',␣
 ↪'guardian',
        'traveltime', 'studytime', 'failures', 'schoolsup', 'famsup', 'paid',␣
 ↪'activities',
        'internet', 'romantic', 'famrel', 'freetime', 'goout', 'health',␣
 ↪'absences']]

#Declaration of independent variable
data['GPA'] = data['GPA'] / 5
y=data['GPA']
```

[ ]:

```
#spliting data
X_train, X_test, y_train, y_test = train_test_split(X,y, train_size=0.8,␣
 ↪test_size=0.2, random_state=42)
```

[ ]:

```
#training the model using linear regression as classifier
regressor = LinearRegression()

regressor.fit(X_train, y_train)
```

[ ]:

```
LinearRegression()
```

[ ]:

```
#prediction score
r_score = regressor.score(X_test,y_test)

#printing output
print("Regression = ",r_score*100)
```

```
Regression =  27.49181537583937
```

[ ]:

```
#performance evaluation using metrics of MAE, MSE and RMSE
x_pred = regressor.predict(X_test)

#performance on Mean Absolute Error
print('Mean Absolute Error')
print('Regression :',metrics.mean_absolute_error(y_test,x_pred))
```

```
Mean Absolute Error
Regression : 0.3565001735492551
```

[ ]:

```
#performance on Mean Squared Error
print('Mean Squared Error')
print('Regression :',metrics.mean_squared_error(y_test,x_pred))
```

```
Mean Squared Error
Regression : 0.20600457310908657
```

[ ]:

```
#performance evaluation using nearest prediction accuracy
x_pred = regressor.predict(X_test)

df = pd.DataFrame({'Actual':y_test,'linearR_Prediction':x_pred})
df
```

[ ]:

```
      Actual  linearR_Prediction
2225     2.4            2.147643
1839     2.0            1.892826
7648     2.0            2.555333
2572     2.6            2.604945
2226     3.0            2.706028
…       …                  …
7498     1.4            1.790410
7494     1.2            1.960695
188      2.8            2.593144
6862     1.8            2.455708
1201     1.8            1.973579

[1541 rows x 2 columns]
```