LEMBAR JAWABAN BABAK FINAL

SEBELAS MARET STATISTICS OLYMPIAD

Gamau Kalah

SSO2023035

Pemodelan Machine Learning: Prediksi Keputusan Mahasiswa Melanjutkan Studi Menggunakan Metode *Supervised Learning Classifier*

Rancangan Penyelesaian Masalah

Berdasarkan metode yang telah dijabarkan pada latar belakang, maka digunakan metode *Supervised Learning Regression* terbaik untuk menentukan model prediktif. Adapun data yang digunakan terdiri dari 4424 baris dan 47 kolom dengan rincian variabel yang akan dianalisis disajikan pada tabel 1 berikut.

Tabel 1. Variabel Penelitian

Variabel	Keterangan
Y	*Target*
X	*Marital status*
	Application order
	Daytime/evening attendance
	Previous qualification (grade)
	Admission grade
	Displaced
	Educational special needs
	Debtor
	Tuition fees up to date
	Gender
	Scholarship holder
	Age at enrollment
	International
	Curricular units 1st sem (credited)
	Curricular units 1st sem (enrolled)
	Curricular units 1st sem (evaluations)
	Curricular units 1st sem (approved)
	Curricular units 1st sem (grade)
	Curricular units 1st sem (without evaluations)
	Curricular units 2nd sem (credited)
	Curricular units 2nd sem (enrolled)
	Curricular units 2nd sem (evaluations)
	Curricular units 2nd sem (approved)
	Curricular units 2nd sem (grade)
	Curricular units 2nd sem (without evaluations)
	Unemployment rate
	Inflation rate
	GDP

Metode regresi digunakan untuk melihat pengaruh variabel X terhadap variabel Y sehingga dapat diidentifikasi faktor-faktor apa saja yang berkontribusi pada keputusan mahasiswa. Tahapan langkah analisis yang dilakukan adalah sebagai berikut.

1. *Processing* data.
   a. Mengkategorikan data pada variabel Target (Y) dengan angka.

b. Melakukan evaluasi data dengan menghapus variabel yang memiliki kasus *missing value*.
c. Mengeksplorasi data untuk mengetahui dugaan korelasi antar variabel penelitian.
d. Memeriksa *outlier* pada data numerik.
e. Melakukan *scalling* data untuk meyeragamkan data menggunakan metode transformasi *robust scaller*, *minmax scaller*, dan *standart scaller*.

2. Melakukan pemodelan prediksi dengan *Supervised Learning Regression.*
   a. Membagi data menjadi training dan testing dengan proporsi 80% dan 20% serta 90% dan 10%.
   b. Melakukan pemodelan menggunakan metode *Linear Regression, Decision Tree Regression, Random Forest Regression, K-Nearest Neighbor Regression,* serta *Support Machine Vector Regression.*
   c. Melakukan evaluasi model dengan memilih nilai *R-square* tertinggi dan nilai galat paling rendah.
3. Melakukan pemeriksaan nilai *Importance* variabel X terhadap variabel Y pada model yang terpilih.

Hasil dan Pembahasan

1. Hasil *Processing* Data

Analisis data diawali dengan mengubah data pada variabel Target (Y) menjadi data kategorik dengan rincian: *Dropout* = 0, *Enrolled* = 1, dan 2 = *Graduated*. Selanjutnya, memeriksa persentase *missing value* pada setiap variabel yang digunakan guna mengevaluasi data. Hasil pemeriksaan *missing value* disajikan pada tabel 2 berikut.

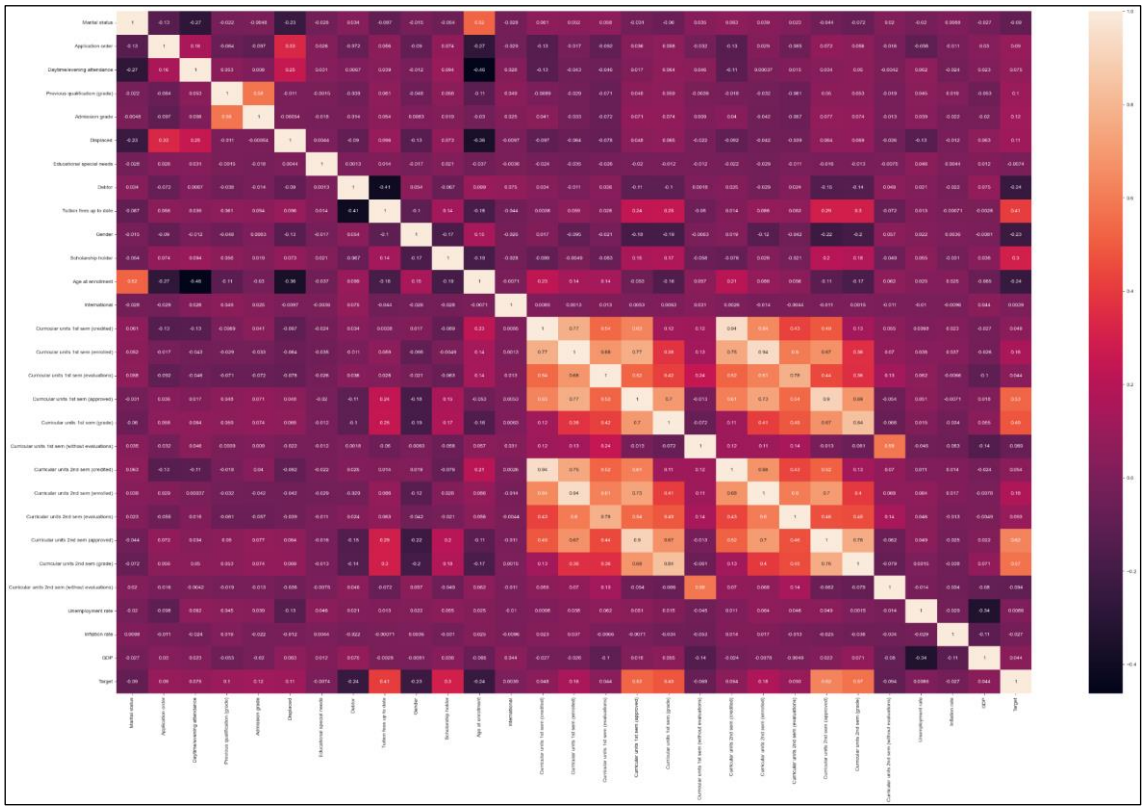
Tabel 2. Persentase *Missing Value*

Variabel	Persentase
Target	0%
Marital status	0%
Application order	0%
Daytime/evening attendance	0%
Previous qualification (grade)	0%
Admission grade	0%
Displaced	0%
Educational special needs	0%
Debtor	0%
Tuition fees up to date	0%
Gender	0%
Scholarship holder	0%

Tabel 2. Persentase *Missing Value*

Age at enrollment	0%
International	0%
Curricular units 1st sem (credited)	0%
Curricular units 1st sem (enrolled)	0%
Curricular units 1st sem (evaluations)	0%
Curricular units 1st sem (approved)	0%
Curricular units 1st sem (grade)	0%
Curricular units 1st sem (without evaluations)	0%
Curricular units 2nd sem (credited)	0%
Curricular units 2nd sem (enrolled)	0%
Curricular units 2nd sem (evaluations)	0%
Curricular units 2nd sem (approved)	0%
Curricular units 2nd sem (grade)	0%
Curricular units 2nd sem (without evaluations)	0%
Unemployment rate	0%
Inflation rate	0%
GDP	0%

Tabel 2 menunjukkan bahwa persentase *missing value* pada seluruh data yang digunakan bernilai 0%, maka tidak perlu melakukan penghapusan variabel sehingga analisis dapat dilanjutkan dengan mengeksplorasi data. Hasil eksplorasi data hubungan antar variabel disajikan pada gambar 1 berikut.



Gambar 1. Plot Korelasi Antar Variabel

Gambar 1 menunjukkan bahwa antar variabel X memiliki korelasi yang negatif hingga positif sehingga saling mempengaruhi dan dapat diindikasikan terdapat kasus multikolinearitas. Namun, pada analisis ini tidak perlu dilakukan penanganan pada kasus multikolinearitas. Selain itu, gambar 1 menunjukkan visualisasi hubungan antar variabel X terhadap variabel Y yang masing-masing disajikan pada tabel 3 berikut.

Tabel 3. Hubungan Variabel Target dengan Setiap Variabel X

Keterangan	Hubungan
Marital status	Negatif
Application order	Positif
Daytime/evening attendance	Positif
Previous qualification (grade)	Positif
Admission grade	Positif
Displaced	Positif
Educational special needs	Negatif
Debtor	Negatif
Tuition fees up to date	Positif
Gender	Negatif
Scholarship holder	Positif
Age at enrollment	Negatif
International	Positif
Curricular units 1st sem (credited)	Positif
Curricular units 1st sem (enrolled)	Positif
Curricular units 1st sem (evaluations)	Positif
Curricular units 1st sem (approved)	Positif
Curricular units 1st sem (grade)	Positif
Curricular units 1st sem (without evaluations)	Negatif
Curricular units 2nd sem (credited)	Positif
Curricular units 2nd sem (enrolled)	Positif
Curricular units 2nd sem (evaluations)	Positif
Curricular units 2nd sem (approved)	Positif
Curricular units 2nd sem (grade)	Positif
Curricular units 2nd sem (without evaluations)	Negatif
Unemployment rate	Positif
Inflation rate	Negatif
GDP	Positif

Selanjutnya dilakukan pemeriksaan *outlier* pada variabel dengan data yang bersifat numerik dan hasilnya disajikan pada tabel 4 berikut.

Tabel 4. Jumlah *Outlier* Data Variabel Numerik

Keterangan	Hubungan
Previous qualification (grade)	179
Admission grade	86
Age at enrollment	441
Curricular units 1st sem (credited)	577
Curricular units 1st sem (enrolled)	424
Curricular units 1st sem (evaluations)	158
Curricular units 1st sem (approved)	180

Tabel 5. Jumlah *Outlier* Data Variabel Numerik

Curricular units 1st sem (grade)	726
Curricular units 1st sem (without evaluations)	294
Curricular units 2nd sem (credited)	530
Curricular units 2nd sem (enrolled)	369
Curricular units 2nd sem (evaluations)	109
Curricular units 2nd sem (approved)	44
Curricular units 2nd sem (grade)	877
Curricular units 2nd sem (without evaluations)	282
Unemployment rate	0
Inflation rate	0
GDP	0

Tabel 4 menunjukkan bahwa di antara sebelas variabel dengan data numerik yang diperiksa, hanya terdapat tiga variabel yang tidak memiliki data *outlier*, yakni *Unemployee rate*, *Inflation rate*, serta GDP. Kemudian, dengan metode *robust scaller*, *minmax scaller*, dan *standart scaller* dilakukan scalling data yang hasilnya disajikan pada gambar 2 berikut.

	Marital status	Application order	Daytime/evening attendance\t	Previous qualification (grade)	Admission grade	Displaced	Educational special needs	Debtor	Tuition fees up to date	Gender	...	Curricular units 2nd sem (credited) Robust Scl	Curricular units 2nd sem (enrolled) Robust Scl	Curricular units 2nd sem (evaluations) Robust Scl	Curricular units 2nd sem (approved) Robust Scl	Curric units (gr Rol
0	1	5	1	122.0	127.3	1	0	0	1	1	...	0.0	-3.0	-2.00	-1.25	-4.72
1	1	1	1	160.0	142.5	1	0	0	0	1	...	0.0	0.0	-0.50	0.25	0.56
2	1	5	1	122.0	124.8	1	0	0	0	1	...	0.0	0.0	-2.00	-1.25	-4.72
3	1	2	1	122.0	119.6	1	0	0	1	0	...	0.0	0.0	0.50	0.00	0.07
4	2	1	0	100.0	141.5	0	0	0	1	0	...	0.0	0.0	-0.50	0.25	0.30
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4419	1	6	1	125.0	122.2	0	0	0	1	1	...	0.0	0.0	0.00	0.00	0.18
4420	1	2	1	120.0	119.0	1	0	1	0	0	...	0.0	0.0	-0.50	-0.75	-0.46
4421	1	1	1	154.0	149.5	1	0	0	1	0	...	0.0	1.0	0.25	-1.00	0.50
4422	1	1	1	180.0	153.8	1	0	0	1	0	...	0.0	-0.5	-0.50	0.00	-0.07
4423	1	1	1	152.0	152.0	1	0	0	1	0	...	0.0	0.0	-0.50	0.25	0.30

(a) *Robust Scaller*

	Marital status	Application order	Daytime/evening attendance\t	Previous qualification (grade)	Admission grade	Displaced	Educational special needs	Debtor	Tuition fees up to date	Gender	...	Curricular units 2nd sem (credited) MinMax Scl	Curricular units 2nd sem (enrolled) MinMax Scl	Curricular units 2nd sem (evaluations) MinMax Scl	Curricular units 2nd sem (approved) MinMax Scl	Curric units (gr Mini
0	1	5	1	122.0	127.3	1	0	0	1	1	...	0.0	0.000000	0.000000	0.00	0.00
1	1	1	1	160.0	142.5	1	0	0	0	1	...	0.0	0.260870	0.181818	0.30	0.73
2	1	5	1	122.0	124.8	1	0	0	0	1	...	0.0	0.260870	0.000000	0.00	0.00
3	1	2	1	122.0	119.6	1	0	0	1	0	...	0.0	0.260870	0.303030	0.25	0.66
4	2	1	0	100.0	141.5	0	0	0	1	0	...	0.0	0.260870	0.181818	0.30	0.70
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4419	1	6	1	125.0	122.2	0	0	0	1	1	...	0.0	0.260870	0.242424	0.25	0.68
4420	1	2	1	120.0	119.0	1	0	1	0	0	...	0.0	0.260870	0.181818	0.10	0.59
4421	1	1	1	154.0	149.5	1	0	0	1	0	...	0.0	0.347826	0.272727	0.05	0.72
4422	1	1	1	180.0	153.8	1	0	0	1	0	...	0.0	0.217391	0.181818	0.25	0.64
4423	1	1	1	152.0	152.0	1	0	0	1	0	...	0.0	0.260870	0.181818	0.30	0.70

(b) *MinMax Scaller*

	Marital status	Application order	Daytime/evening attendance\t	Previous qualification (grade)	Admission grade	Displaced	Educational special needs	Debtor	Tuition fees up to date	Gender	...	Curricular units 2nd sem (credited) Standard Scl	Curricular units 2nd sem (enrolled) Standard Scl	Curricular units 2nd sem (evaluations) Standard Scl	Curricular units 2nd sem (approved) Standard Scl	Curric units (gr Stan
0	1	5	1	122.0	127.3	1	0	0	1	1	...	-0.282442	-2.838337	-2.042630	-1.471527	-1.96
1	1	1	1	160.0	142.5	1	0	0	0	1	...	-0.282442	-0.105726	-0.522682	0.518904	0.65
2	1	5	1	122.0	124.8	1	0	0	0	1	...	-0.282442	-0.105726	-2.042630	-1.471527	-1.96
3	1	2	1	122.0	119.6	1	0	0	1	0	...	-0.282442	-0.105726	0.490616	0.187165	0.41
4	2	1	0	100.0	141.5	0	0	0	1	0	...	-0.282442	-0.105726	-0.522682	0.518904	0.53
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4419	1	6	1	125.0	122.2	0	0	0	1	1	...	-0.282442	-0.105726	-0.016033	0.187165	0.46
4420	1	2	1	120.0	119.0	1	0	1	0	0	...	-0.282442	-0.105726	-0.522682	-0.808050	0.14
4421	1	1	1	154.0	149.5	1	0	0	1	0	...	-0.282442	0.805144	0.237291	-1.139788	0.62
4422	1	1	1	180.0	153.8	1	0	0	1	0	...	-0.282442	-0.561161	-0.522682	0.187165	0.33
4423	1	1	1	152.0	152.0	1	0	0	1	0	...	-0.282442	-0.105726	-0.522682	0.518904	0.53

(c) *Standard Scaller*

Gambar 2. Hasil Transformasi Data

2. Hasil Pemodelan Prediksi

Untuk melihat model prediksi terbaik akan diuji dengan dua skenario *splitting* data *training* dan *testing* yaitu sebesar 80 20 dan 90 10. Hasil uji kebaikan model untuk dua skenario dari masing-masing metode ditunjukkan oleh tabel 6.

Tabel 6. Kebaikan Model Hasil Prediksi

Model Regresi	Training Testing: 80% 20%				
	R-Square	MAE	MSE	RMSE	Accuracy
Robust Scaller					
Logistic regression	0.481	0.290	0.412	0.642	0.770
KNN	0.377	0.348	0.495	0.704	0.730
Decission Tree	0.243	0.420	0.601	0.775	0.670
Random Forest	0.536	0.264	0.368	0.607	0.790
SVM	0.475	0.297	0.417	0.646	0.760
Min Max Scaller					
Logistic regression	0.485	0.287	0.409	0.640	0.770
KNN	0.191	0.428	0.643	0.802	0.680
Decission Tree	0.223	0.427	0.617	0.785	0.670
Random Forest	0.548	0.258	0.359	0.599	0.790
SVM	0.401	0.327	0.476	0.690	0.750
Standard Scaller					
Logistic regression	0.474	0.294	0.418	0.647	0.770
KNN	0.388	0.346	0.486	0.697	0.720
Decission Tree	0.210	0.433	0.627	0.792	0.660
Random Forest	0.555	0.252	0.354	0.595	0.800
SVM	0.461	0.304	0.428	0.654	0.760

Berdasarkan hasil pemodelan yang ditunjukkan oleh Tabel x, model prediksi terbaik adalah *Decision Tree Regression* dengan transformasi data *MinMax Scaller* dan *splitting* data sebesar 90% dan 10%. Hal ini ditunjukkan dengan tingkat akurasi sebesar 81% dan nilai R-*Square* sebesar 61.8% serta nilai RMSE sebesar 0.552.

3. Hasil Analisis *Importance*

Model prediksi yang digunakan adalah *Decision Tree Regression* dengan transformasi data *MinMax Scaller* dan *splitting* data sebesar 90% dan 10%. Kemudian dilanjutkan analisis variabel penting yang berpengaruh terhadap variabel Y (Target).

Tabel 7. Skor Importance Variabel X

Rank	Index	*Features*	*Importances*
1	22	Curricular units 2nd sem (approved) MinMax Scl	0.1648
2	23	Curricular units 2nd sem (grade) MinMax Scl	0.1070
3	16	Curricular units 1st sem (approved) MinMax Scl	0.0969
4	17	Curricular units 1st sem (grade) MinMax Scl	0.0911
5	11	Admission grade MinMax Scl	0.0658
6	10	Previous qualification (grade) MinMax Scl	0.0599
7	21	Curricular units 2nd sem (evaluations) MinMax ...	0.0463
8	15	Curricular units 1st sem (evaluations) MinMax ...	0.0451
9	6	Tuition fees up to date	0.0424
10	27	GDP MinMax Scl	0.0313
11	25	Unemployment rate MinMax Scl	0.0301
12	14	Curricular units 1st sem (enrolled) MinMax Scl	0.0297
13	26	Inflation rate MinMax Scl	0.0292
14	20	Curricular units 2nd sem (enrolled) MinMax Scl	0.0275
15	1	Application order	0.0223
16	8	Scholarship holder	0.0177
17	5	Debtor	0.0156
18	7	Gender	0.0144
19	19	Curricular units 2nd sem (credited) MinMax Scl	0.0102
20	12	Displaced MinMax Scl	0.0099
21	3	Displaced	0.0089
22	0	Marital status	0.0074
23	18	Curricular units 1st sem (without evaluations)...	0.0072
24	24	Curricular units 2nd sem (without evaluations)...	0.0068
25	2	Daytime/evening attendance\t	0.0060
26	13	International MinMax Scl	0.0025
27	9	International	0.0024
28	4	Educational special needs	0.0017

Tabel 7 menunjukkan bahwa variabel yang paling mempengaruhi keputusan mahasiswa akan dropot, terdaftar, atau lulus adalah *curricular units 2nd sem (approved)*. Kemudian, beberapa variabel penting lainnya adalah *curricular unit 2nd sem (grades)*, *curricular unit 1st sem (approved)*, dan *admission grade.*

Kesimpulan

Berdasarkan hasil analisis yang telah dijabarkan dapat ditarik kesimpulan bahwa seluruh variabel yang digunakan tidak terdapat *missing value*, meski terindikasi terdapat kasus multikolinearitas. Setelah dilakukan pemodelan prediktif didapatkan model terbaik yang dipilih adalah dengan metode *Decision Tree Regression* dengan transformasi data *MinMax Scaller* dan *splitting* data sebesar 90% dan 10%. Sedangkan hasil analisis *Importance* didapatkan asumsi faktor-faktor yang mempengaruhi keputusan mahasiswa untuk melanjutkan kuliahnya adalah variabel *curricular units 2nd sem (approved)*, *curricular unit 2nd sem (grades)*, *curricular unit 1st sem (approved)*, dan *admission grade.*

Lampiran

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, MinMaxScaler,
MaxAbsScaler, RobustScaler, Normalizer, Binarizer

#input data
data=pd.read_csv("DataSSO2023
(1).csv",encoding='unicode_escape')

#Encoding
from sklearn.preprocessing import LabelEncoder
lE = LabelEncoder()
data['Target'] = lE.fit_transform(data['Target'])

#Missing Value
round(data.isnull().sum()/data.shape[0]*100, 2)
#variabel kategori
colnames=data.columns[[0,1,2,5,6,7,8,9,10,12,28]]
list(colnames)

#cek outlier
def find_outlier(yourlist):
    outlier = ''
```

```
    for i in range(len(yourlist)):
        if yourlist[i] > batas_atas:
            outlier += str(yourlist[i]) + ', '
        if yourlist[i] < batas_bawah:
            outlier += str(yourlist[i]) + ', '
    return outlier
for j in data.columns.drop(list(colnames)):
    Q1 = data[j].describe()['25%']
    Q3 = data[j].describe()['75%']
    IQR = Q3 - Q1
    batas_bawah= Q1-(1.5*IQR)
    batas_atas = Q3+(1.5*IQR)
    if find_outlier(data[j]) == '':
        print('Feature {} has no Outlier'.format(j))
    else:
        print('Feature {} has {} Outlier which is:'.format(j,
(len(find_outlier(data[j]).split(',')))-1))
        print(find_outlier(data[j]))

#Scalling
##MinMax
for                              i                                  in
data.columns.drop(list(data.columns[[0,1,2,6,7,8,9,10,11,13]])):
    data['{}        MinMax        Scl        '.format(i)]        =
MinMaxScaler().fit_transform(data[[i]])
##Robust
for                              i                                  in
data.columns.drop(list(data.columns[[0,1,2,6,7,8,9,10,11,13]])):
    data['{}        Robust        Scl        '.format(i)]        =
RobustScaler().fit_transform(data[[i]])
##Standard
for                              i                                  in
data.columns.drop(list(data.columns[[0,1,2,6,7,8,9,10,11,13]])):
    data['{}       Standard       Scl       '.format(i)]        =
StandardScaler().fit_transform(data[[i]])
#modeling
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import plot_confusion_matrix
from  sklearn.model_selection  import  train_test_split  #untuk
splitting data
from  sklearn.linear_model  import  LinearRegression  #untuk  buat
prediksi linearRegression
from sklearn import metrics #untuk mengukur performa model
from sklearn import svm
from sklearn.svm import SVC

dtfix.columns[[0,1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,20
,21,22,23,24,25,26,27,28]]
```

```
list(dtfix.columns[[0,1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,
19,20,21,22,23,24,25,26,27,28]])

##splitting data X dan Y
X=
dtfix[list(dtfix.columns[[0,1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,
17,18,19,20,21,22,23,24,25,26,27,28]])].values
y = dtfix[['Target']].values
##80 20
X_train,  X_test,  y_train,  y_test  =  train_test_split(X,  y,
test_size=0.2, random_state=0)
## 90 10
X_train,  X_test,  y_train,  y_test  =  train_test_split(X,  y,
test_size=0.1, random_state=0)
#logistic regression
model_lr = LogisticRegression()
model_lr.fit(X_train, y_train)
y_pred = model_lr.predict(X_test)
print(classification_report(y_test, y_pred))
print('Mean          Absolute          Error(MAE):          ',
metrics.mean_absolute_error(y_test, y_pred))
print('Mean           Squared           Error(MSE):          ',
metrics.mean_squared_error(y_test, y_pred))
print('Root       Mean       Squared       Error(RMSE):       ',
np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
print('R2 Score:', round(metrics.r2_score(y_test, y_pred), 4))
##KNN
model_knn = KNeighborsClassifier()
model_knn.fit(X_train, y_train)
y_pred_knn = model_knn.predict(X_test)
print(classification_report(y_test, y_pred_knn))
print('Mean          Absolute          Error(MAE):          ',
metrics.mean_absolute_error(y_test, y_pred_knn))
print('Mean           Squared           Error(MSE):          ',
metrics.mean_squared_error(y_test, y_pred_knn))
print('Root       Mean       Squared       Error(RMSE):       ',
np.sqrt(metrics.mean_squared_error(y_test, y_pred_knn)))
print('R2  Score:',  round(metrics.r2_score(y_test,  y_pred_knn),
4))
##Decision tree
model_tree = DecisionTreeClassifier()
model_tree.fit(X_train, y_train)
y_pred_tree = model_tree.predict(X_test)
print(classification_report(y_test, y_pred_tree))
model_tree = DecisionTreeClassifier()
model_tree.fit(X_train, y_train)
y_pred_tree = model_tree.predict(X_test)
print(classification_report(y_test, y_pred_tree))
## Random Forest
randomforest = RandomForestClassifier()
randomforest.fit(X_train, y_train)
y_pred_rand = randomforest.predict(X_test)
print(classification_report(y_test, y_pred_rand))
```

```
print('Mean Absolute Error(MAE): ',
metrics.mean_absolute_error(y_test, y_pred_rand))
print('Mean Squared Error(MSE): ',
metrics.mean_squared_error(y_test, y_pred_rand))
print('Root Mean Squared Error(RMSE): ',
np.sqrt(metrics.mean_squared_error(y_test, y_pred_rand)))
print('R2 Score:', round(metrics.r2_score(y_test, y_pred_rand),
4))
##SVM
model_svm = svm.SVC()
model_svm.fit(X_train, y_train)
y_pred_svm = model_svm.predict(X_test)
print(classification_report(y_test, y_pred_svm))
print('Mean Absolute Error(MAE): ',
metrics.mean_absolute_error(y_test, y_pred_svm))
print('Mean Squared Error(MSE): ',
metrics.mean_squared_error(y_test, y_pred_svm))
print('Root Mean Squared Error(RMSE): ',
np.sqrt(metrics.mean_squared_error(y_test, y_pred_svm)))
print('R2 Score:', round(metrics.r2_score(y_test, y_pred_svm),
4))

#Importance Skor
feature_importances = randomforest.feature_importances_
m=dtfix.columns[[0,1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,
20,21,22,23,24,25,26,27,28]]
feature_importance_df = pd.DataFrame({'Feature': m, 'Importance':
feature_importances})
feature_importance_df =
feature_importance_df.sort_values(by='Importance',
ascending=False)
print(feature_importance_df)
feature_importance_df =
feature_importance_df.sort_values(by='Importance',
ascending=True)
plt.figure(figsize=(8, 6))
plt.barh(feature_importance_df['Feature'],
feature_importance_df['Importance'])
plt.xlabel('Feature Importance')
plt.ylabel('Feature Name')
plt.title('Decision Tree Regression Feature Importance')
plt.show()
```