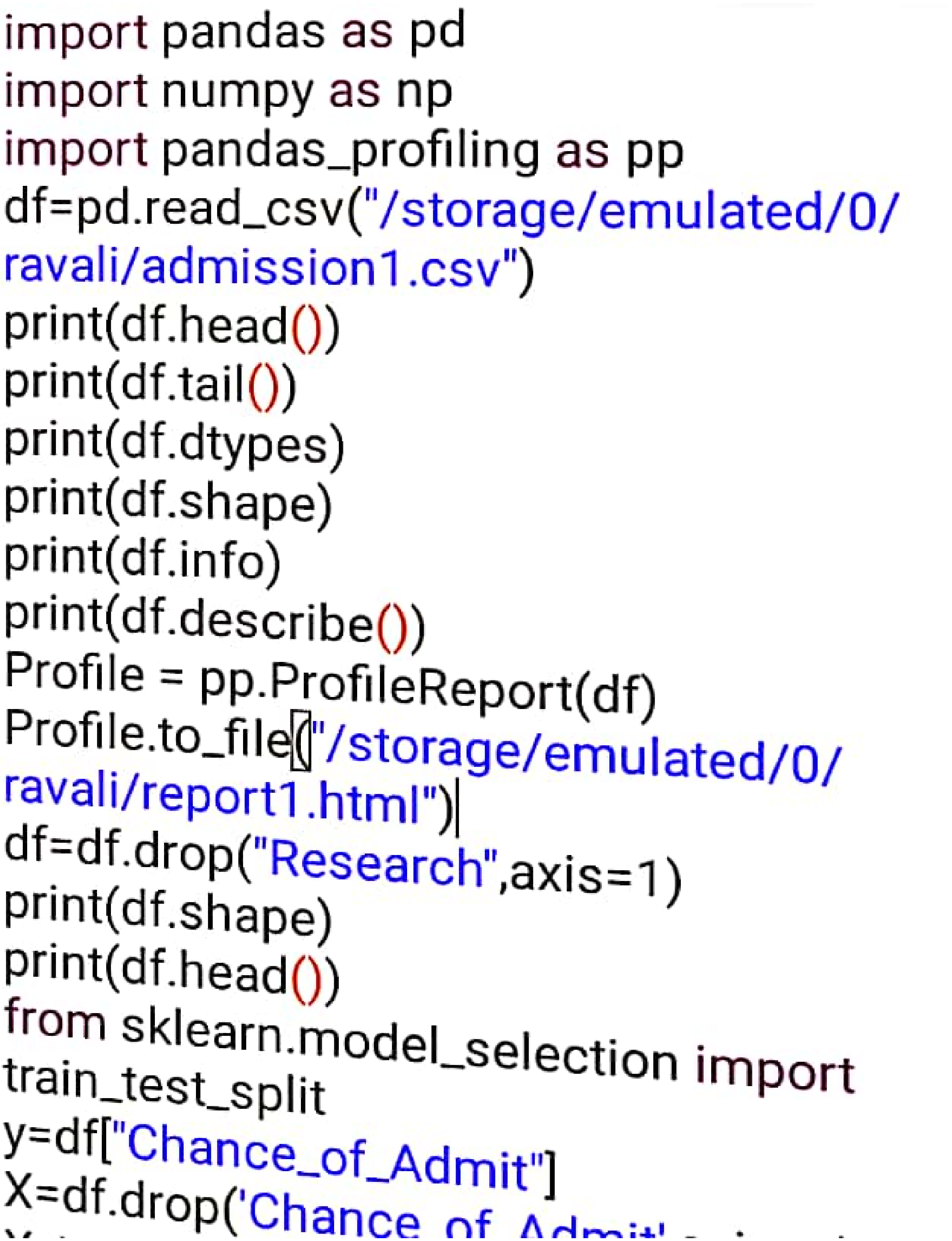
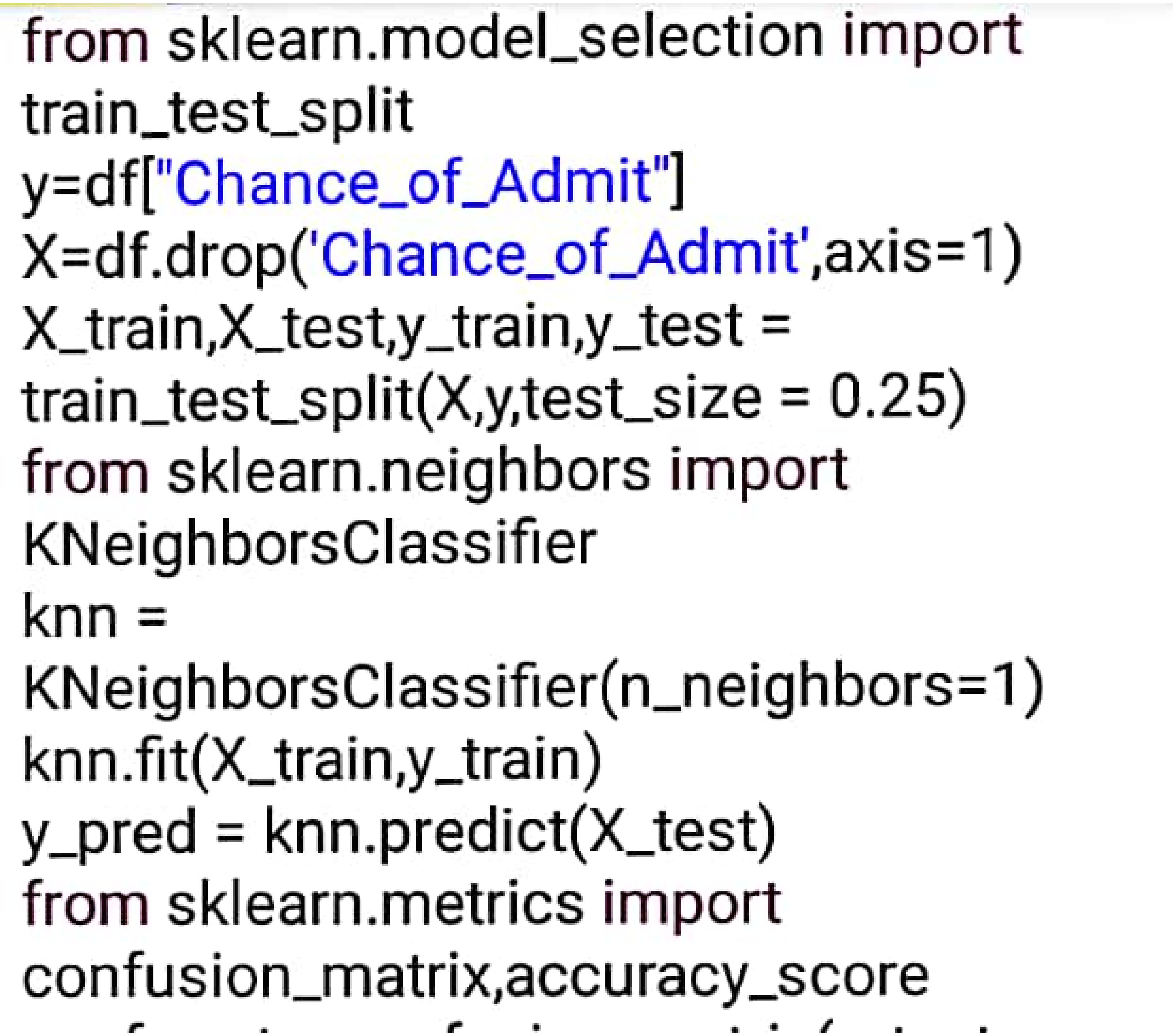
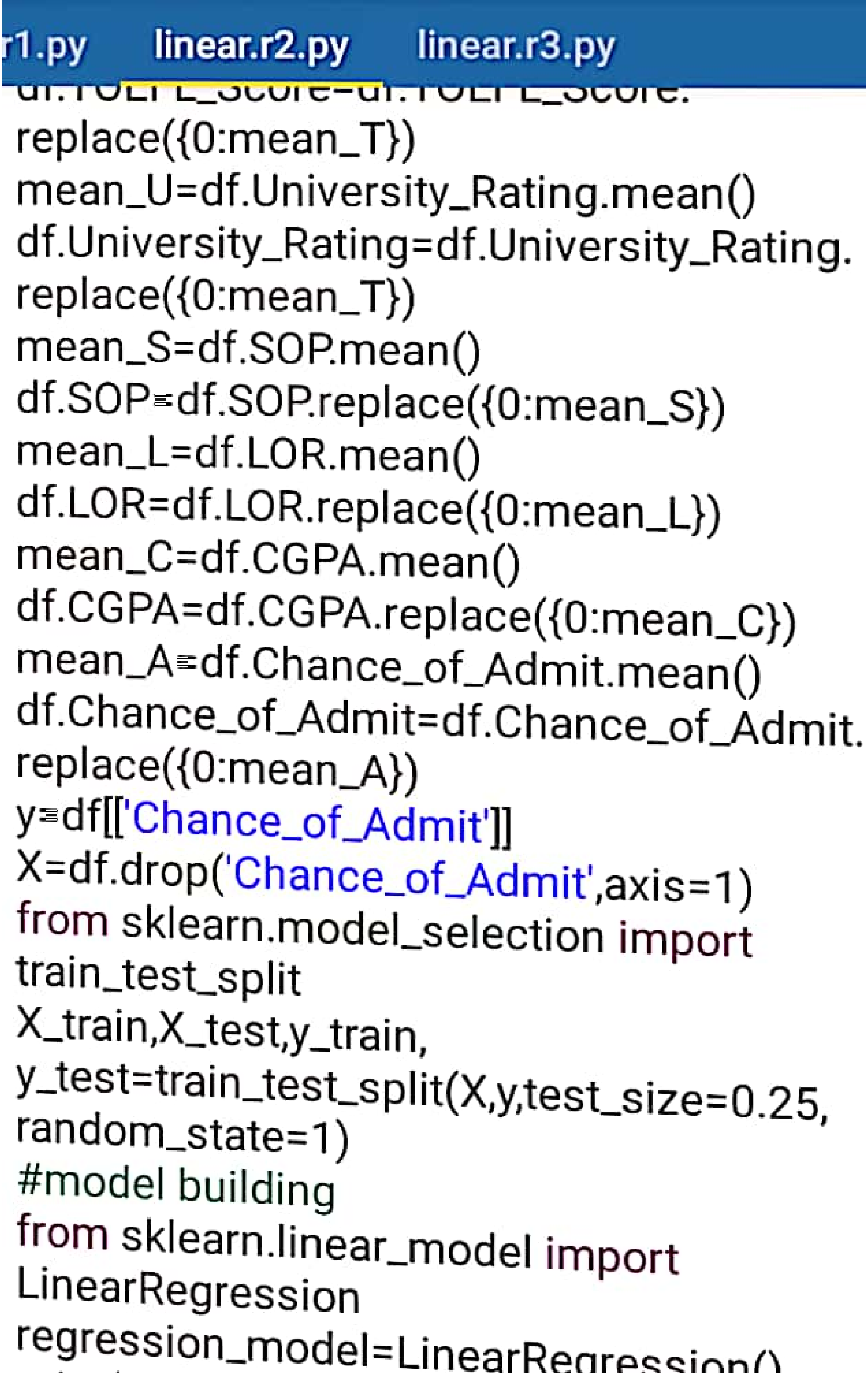


```
import pandas as pd
import numpy as np
import pandas_profiling as pp
df=pd.read_csv("/storage/emulated/0/
ravali/admission1.csv")
print(df.head())
print(df.tail())
print(df.dtypes)
print(df.shape)
print(df.info)
print(df.describe())
Profile = pp.ProfileReport(df)
Profile.to_file("/storage/emulated/0/
ravali/report1.html")
df=df.drop("Research",axis=1)
print(df.shape)
print(df.head())
from sklearn.model_selection import
train_test_split
y=df["Chance_of_Admit"]
X=df.drop('Chance_of_Admit'
#Linear Regression
import pandas as pd
import numpy as np
import pandas_profiling as pp
df=pd.read_csv("/storage/emulated/0/
ravali/admission1.csv")
print(df.head())
print(df.tail())
print(df.dtypes)
print(df.shape)
print(df.info)
print(df.describe())
Profile = pp.ProfileReport(df)
Profile.to_file("/storage/emulated/0/
ravali/report1.html")
df=df.drop("Research",axis=1)
print(df.shape)
print(df.head())
mean_G=df.GRE_Score.mean()
df.GRE_Score=df.GRE_Score.
replace({0:mean_G})
mean_T=df.TOEFL_Score.mean()
df.TOEFL_Score=df.TOEFL_Score
replace({0:mean_T})
```

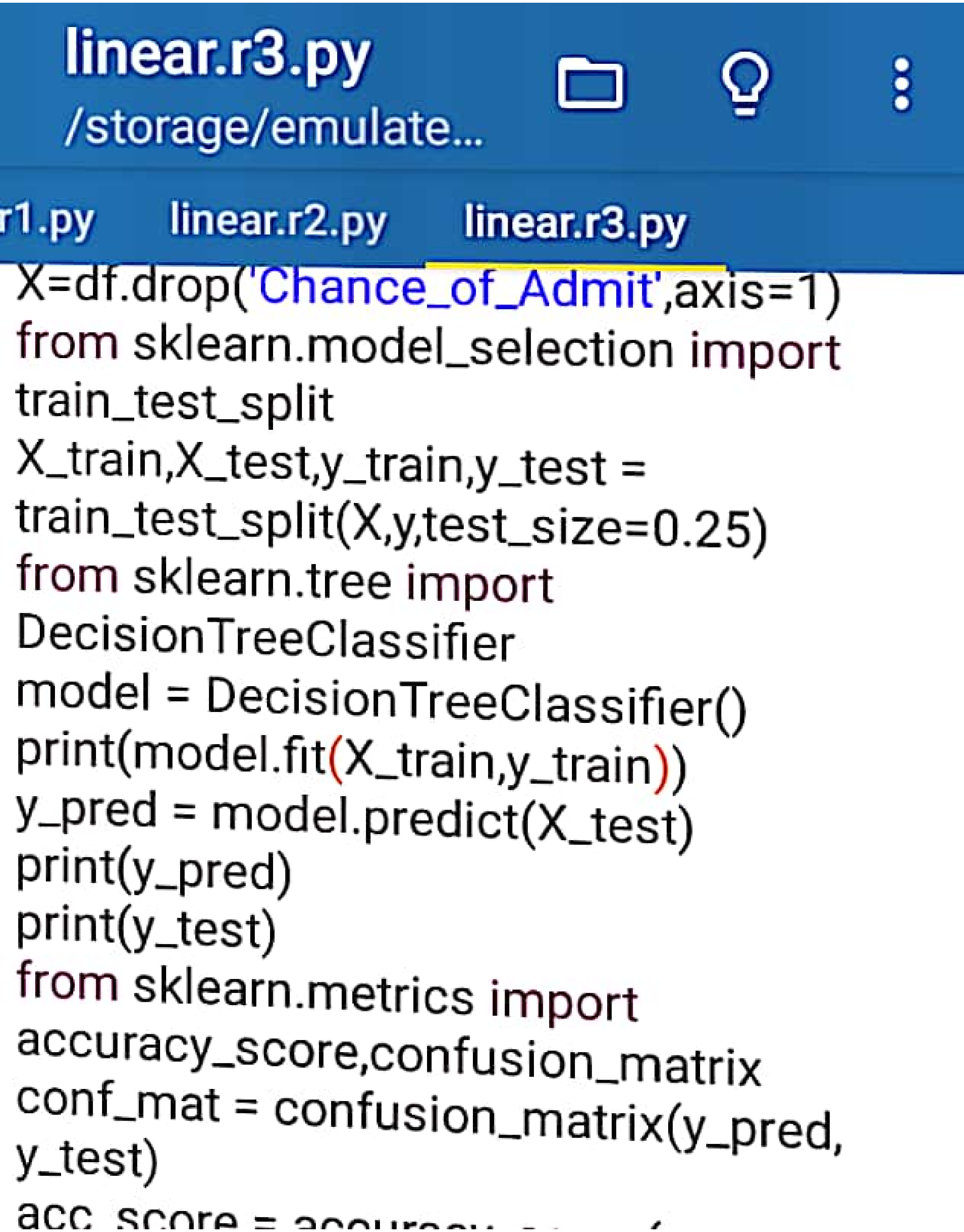


```
from sklearn.model_selection import
train_test_split
y=df["Chance_of_Admit"]
X=df.drop('Chance_of_Admit',axis=1)
X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size = 0.25)
from sklearn.neighbors import
KNeighborsClassifier
knn =
KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train,y_train)
y_pred = knn.predict(X_test)
from sklearn.metrics import
confusion_matrix,accuracy_score
```

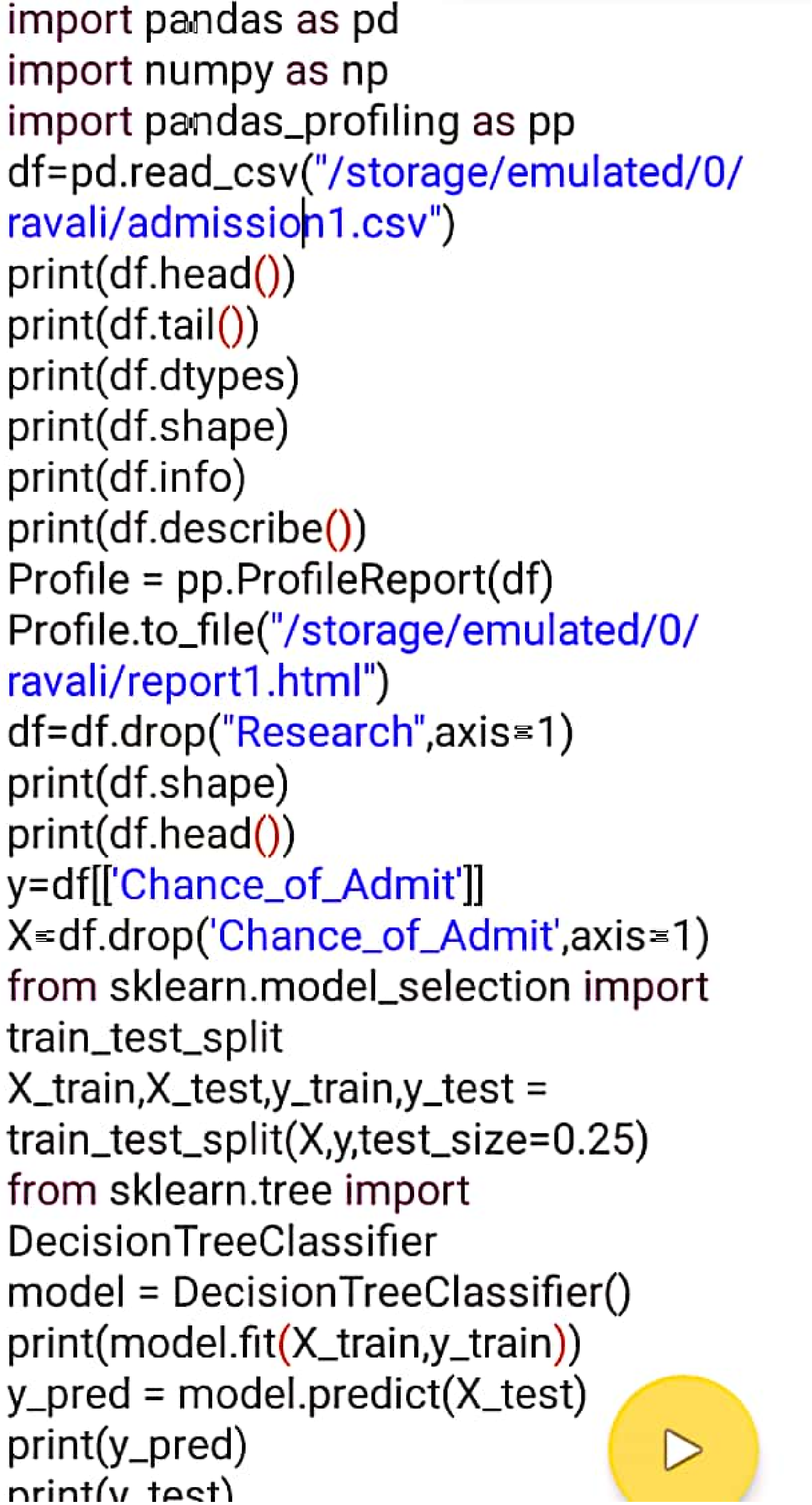
```
print(regression_model.fit(X_train,y_train))
intercept=regression_model.intercept_[0]
print(intercept)
for idx,col_name in enumerate(X_train.
columns):
   print("The co-efficient for {} is {}".
format(col_name,regression_model.
coef_[0][idx]))
#Evaluation metrics
from sklearn.metrics import
mean_squared_error
y_pred=regression_model.predict(X_test)
print(y_pred)
regression_model_mse=mean_squared_er
ror(y_pred,y_test)
print(regression_model_mse)
import math
```



```
df.TOEFL_Score=df.TOEFL_Score.
replace({0:mean_T})
mean_U=df.University_Rating.mean()
df.University_Rating=df.University_Rating.
replace({0:mean_T})
mean_S=df.SOP.mean()
df.SOP=df.SOP.replace({0:mean_S})
mean_L=df.LOR.mean()
df.LOR=df.LOR.replace({0:mean_L})
mean_C=df.CGPA.mean()
df.CGPA=df.CGPA.replace({0:mean_C})
mean_A=df.Chance_of_Admit.mean()
df.Chance_of_Admit=df.Chance_of_Admit.
replace({0:mean_A})
y=df[['Chance_of_Admit']]
X=df.drop('Chance_of_Admit',axis=1)
from sklearn.model_selection import
train_test_split
X_train,X_test,y_train,
y_test=train_test_split(X,y,test_size=0.25,
random_state=1)
#model building
from sklearn.linear_model import
LinearRegression
regression_model=LinearRegression()
```



```
X=df.drop('Chance_of_Admit',axis=1)
from sklearn.model_selection import
train_test_split
X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.25)
from sklearn.tree import
DecisionTreeClassifier
model = DecisionTreeClassifier()
print(model.fit(X_train,y_train))
y_pred = model.predict(X_test)
print(y_pred)
print(y_test)
from sklearn.metrics import
accuracy_score,confusion_matrix
conf_mat = confusion_matrix(y_pred,
y_test)
acc_score = accuracy
```



```
import pandas as pd
import numpy as np
import pandas_profiling as pp
df=pd.read_csv("/storage/emulated/0/
ravali/admission1.csv")
print(df.head())
print(df.tail())
print(df.dtypes)
print(df.shape)
print(df.info)
print(df.describe())
Profile = pp.ProfileReport(df)
Profile.to_file("/storage/emulated/0/
ravali/report1.html")
df=df.drop("Research",axis=1)
print(df.shape)
print(df.head())
y=df[['Chance_of_Admit']]
X=df.drop('Chance_of_Admit',axis=1)
from sklearn.model_selection import
train_test_split
X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.25)
from sklearn.tree import
DecisionTreeClassifier
model = DecisionTreeClassifier()
print(model.fit(X_train,y_train))
y_pred = model.predict(X_test)
print(y_pred)
print(y_test
```