

splitting data into features and target column

```
x = data.drop(columns=['actual_productivity'], axis=1)
y = data['actual_productivity']
```

splitting data into train test split

```
#import train_test_split dependency
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

Scaling the data

```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
```

```
x_train_scaled = scaler.fit_transform(x_train)
```

```
x_test_scaled = scaler.transform(x_test)
```

```
x_train_scaled=x_train
```

```
x_test_scaled=x_test
```

Model building

#importing the metrics libraries

```
from sklearn.metrics import r2_score
```

```
from sklearn.metrics import mean_squared_error
```

```
from sklearn.metrics import mean_absolute_error
```

Linear regression

```
#model building
```

```
#importing linear regression dependency
```

```
from sklearn.linear_model import LinearRegression
```

```
linear=LinearRegression()
```

```
linear.fit(x_train,y_train)
```

```
#linear model mean squared error
```

```
score_train=linear.predict(x_train)
```

```
mse_train=mean_squared_error(y_train,score_train)
```

```
print("mean squared error in training data in linear regression is:",mse_train)
```

```
score_test=linear.predict(x_test)
```

```
mse_test=mean_squared_error(y_test,score_test)
```

```
print("mean squared error in testing data in linear regression is:",mse_test)
```

```
#linear model r2_score
```

```
score_train=linear.predict(x_train)
```

```
mse_train=r2_score(y_train,score_train)
```

```
print("r2_score in training data in linear regression is:",mse_train)
```

```
score_test=linear.predict(x_test)
```

```
mse_test=r2_score(y_test,score_test)
```

```
print("r2_score in test data in linear regression is:",mse_test)
```

```
#linear model mean_absolute_error
```

```
score_train=linear.predict(x_train)
```

```
mse_train=mean_absolute_error(y_train,score_train)
```

```
print("mean_absolute_error in training data in linear regression is:",mse_train)
```

```
score_test=linear.predict(x_test)
mse_test=mean_absolute_error(y_test,score_test)
print("mean_absolute_error in testing data in linear regression is:",mse_test)
```

Random forest model

```
#Random Forest Regressor

from sklearn.ensemble import RandomForestRegressor

RandomForest = RandomForestRegressor()

RandomForest.fit(x_train, y_train)


#Random Forest Regressor mean squared error

score_train=RandomForest.predict(x_train)
mse_train=mean_squared_error(y_train,score_train)
print("mean squared error in training data in Random Forest Regressor is:",mse_train)


score_test=RandomForest.predict(x_test)
mse_test=mean_squared_error(y_test,score_test)
print("mean squared error in testing data in Random Forest Regressor is:",mse_test)


#Random Forest Regressor r2_score

score_train=RandomForest.predict(x_train)
mse_train=r2_score(y_train,score_train)
print("r2_score in training data in Random Forest Regressor is:",mse_train)


score_test=RandomForest.predict(x_test)
mse_test=r2_score(y_test,score_test)
print("r2_score in test data in Random Forest Regressor is:",mse_test)


#Random Forest Regressor mean_absolute_error

score_train=linear.predict(x_train)
```

```
mse_train=mean_absolute_error(y_train,score_train)
print("mean_absolute_error in training data in Random Forest Regressor is:",mse_train)
```

```
score_test=linear.predict(x_test)
mse_test=mean_absolute_error(y_test,score_test)
print("mean_absolute_error in testing data in Random Forest Regressor is:",mse_test)
```

Xgboost

```
#Xgboost regression
```

```
import xgboost as xgb
```

```
model_xgb=xgb.XGBRegressor(n_estimators=200,max_depth=5,learning_rate=0.1)
```

```
model_xgb.fit(x_train,y_train)
```

```
#Xgboost mean squared error
```

```
score_train=model_xgb.predict(x_train)
```

```
mse_train=mean_squared_error(y_train,score_train)
```

```
print("mean squared error in training data in Xgboost regression is:",mse_train)
```

```
score_test=model_xgb.predict(x_test)
```

```
mse_test=mean_squared_error(y_test,score_test)
```

```
print("mean squared error in testing data in Xgboost regressionr is:",mse_test)
```

```
#Xgboost Regressor r2_score
```

```
score_train=model_xgb.predict(x_train)
```

```
mse_train=r2_score(y_train,score_train)
```

```
print("r2_score in training data in Xgboost regression is:",mse_train)
```

```
score_test=model_xgb.predict(x_test)
```

```
mse_test=r2_score(y_test,score_test)
```

```
print("r2_score in test data in Random Xgboost regressionr is:",mse_test)
```

```
#Xgboost regression mean_absolute_error
```

```
score_train=linear.predict(x_train)
```

```
mse_train=mean_absolute_error(y_train,score_train)
```

```
print("mean_absolute_error in training data in Xgboost regression is:",mse_train)
```

```
score_test=linear.predict(x_test)
```

```
mse_test=mean_absolute_error(y_test,score_test)
```

```
print("mean_absolute_error in testing data in Xgboost regression is:",mse_test)
```