

## ML Assignment – 7

**Title:** Implement Regression Technique and evaluate its performance

**Aim:** To perform Regression using different regression techniques and compare performance using Python

**Objectives:** To Implement Regression Technique and evaluate its performance

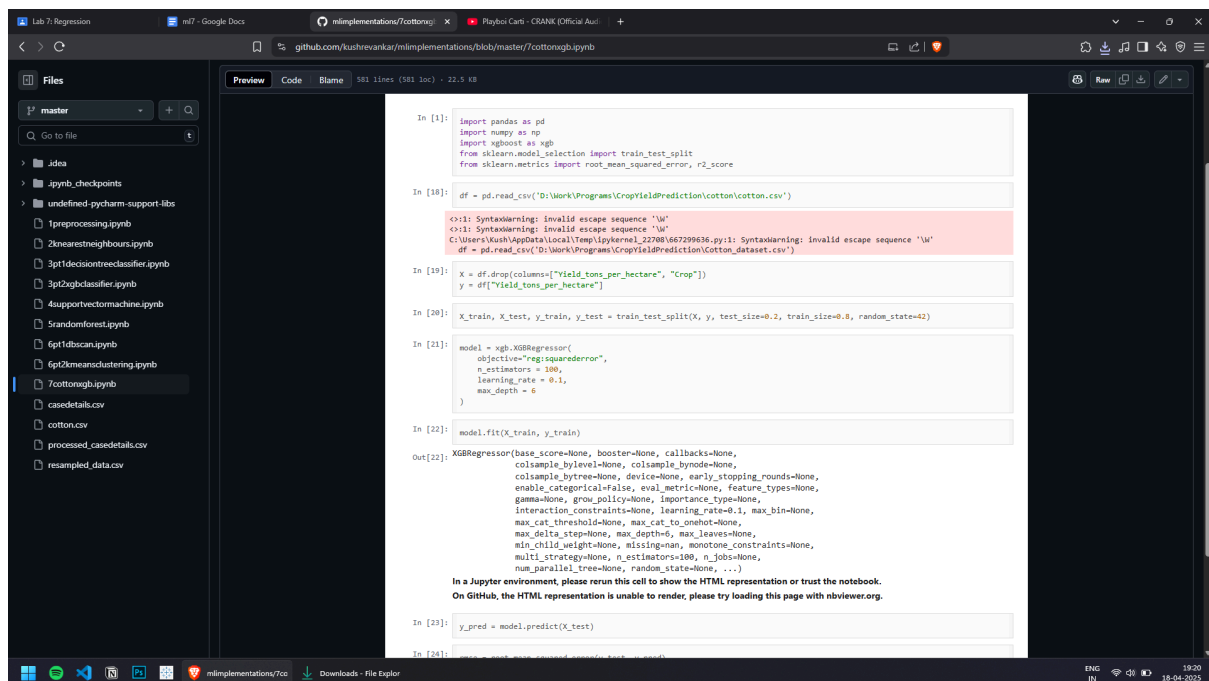
**Problem statement:** Implement Regression Technique and evaluate its performance

**Input: Dataset**

FAQ's

1. What is Regression?
2. List different Regression Methods in Machine Learning ?
3. Which different evaluation metrics are used for regression? List them and explain

Code & Output:



```
In [1]: import pandas as pd
import numpy as np
import xgboost as xgb
from sklearn.model_selection import train_test_split
from sklearn.metrics import root_mean_squared_error, r2_score

In [18]: df = pd.read_csv('D:\Work\Programs\CropYieldPrediction\cotton\cotton.csv')

<!!: SyntaxWarning: invalid escape sequence '\M'
<!!: SyntaxWarning: invalid escape sequence '\M'
C:\Users\Kush\AppData\Local\Temp\ipykernel_22780\467299636.py:1: SyntaxWarning: invalid escape sequence '\M'
df = pd.read_csv('D:\Work\Programs\CropYieldPrediction\cotton\dataset.csv')

In [19]: X = df.drop(columns=["Yield_tons_per_hectare", "Crop"])
y = df["Yield_tons_per_hectare"]

In [20]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, train_size=0.8, random_state=42)

In [21]: model = xgb.XGBRegressor(
    objective="reg:squarederror",
    n_estimators=100,
    learning_rate=0.1,
    max_depth=6
)

In [22]: model.fit(X_train, y_train)

Out[22]: XGBRegressor(base_score=None, booster=None, callbacks=None,
    colsample_bylevel=None, colsample_bynode=None,
    colsample_bynode=None, device=None, early_stopping_rounds=None,
    enable_categorical=False, eval_metric=None, feature_types=None,
    gamma=None, grow_policy=None, importance_type=None,
    interaction_constraints=None, learning_rate=0.1, max_bin=None,
    max_cat_threshold=None, max_cat_to_ohot=None,
    max_delta_step=None, max_depth=6, max_leaves=None,
    min_child_weight=None, missing=None, monotone_constraints=None,
    multi_strategy=None, n_estimators=100, n_jobs=None,
    num_parallel_tree=None, random_state=None, ...)

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [23]: y_pred = model.predict(X_test)

In [24]:
```

Lab 7: Regression

ml7 - Google Docs

mlimplementations/7cottonygb.ipynb

Playboi Carti - CRANK (Official Audi

github.com/kushrevankar/mlimplementations/blob/master/7cottonygb.ipynb

Files

master

Go to file

adoc

ipyb\_checkpoints

undefined-pycharm-support-libs

1preprocessing.ipynb

2knearestneighbours.ipynb

3pt1decisiontreeclassifier.ipynb

3pt2xgbclassifier.ipynb

4supportvectormachine.ipynb

5randomforest.ipynb

6pt1dscan.ipynb

6pt2kmeansclustering.ipynb

7cottonygb.ipynb

caseDetails.csv

cotton.csv

processed\_caseDetails.csv

resampled\_data.csv

mlimplementations / 7cottonygb.ipynb

581 lines (581 box) · 22.5 KB

Preview Code Blame

In [19]:  
x = df.drop(columns=["Yield\_tons\_per\_hectare", "Crop"])  
y = df["Yield\_tons\_per\_hectare"]  
  
In [20]:  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, train\_size=0.8, random\_state=42)  
  
In [21]:  
model = xgb.XGBRegressor(  
 objective="reg:squarederror",  
 n\_estimators = 100,  
 learning\_rate = 0.1,  
 max\_depth = 6  
)  
  
In [22]:  
model.fit(X\_train, y\_train)  
  
Out[22]:  
XGBRegressor(base\_score=None, booster=None, callbacks=None,  
 colsample\_bylevel=None, colsample\_bynode=None,  
 colsample\_bynode=None, device=None, early\_stopping\_rounds=None,  
 enable\_categorical=False, eval\_metric=None, feature\_types=None,  
 gamma=None, grow\_policy=None, importance\_type=None,  
 interaction\_constraints=None, learning\_rate=0.1, max\_bin=None,  
 max\_cat\_threshold=None, max\_cat\_to\_onehot=None,  
 max\_delta\_step=None, max\_depth=6, max\_leaves=None,  
 min\_child\_weight=None, missing=nan, monotone\_constraints=None,  
 multi\_strategy=None, n\_estimators=100, n\_jobs=None,  
 num\_parallel\_tree=None, random\_state=None, ...)  
  
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.  
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.  
  
In [23]:  
y\_pred = model.predict(X\_test)  
  
In [24]:  
mse = root\_mean\_squared\_error(y\_test, y\_pred)  
print("Root Mean Squared Error:", mse)  
  
Root Mean Squared Error: 0.5011964917055526  
  
In [25]:  
acc = r2\_score(y\_test, y\_pred)  
print("R2 Score:", acc)  
  
R2 Score: 0.9124594487908881

mlimplementations/7c

Downloads - File Explor

ENG IN

19:20 18-04-2025