Project Name

Dry Bean Predictor

**Problem Statement**

To build a classification methodology to predict the type of forest cover based on the given training

data.

* **Dataset Information**

**1. Data Set Name:** Dry Bean Dataset

**2. Abstract:**

Images of 13,611 grains of 7 different registered dry beans were taken with a high-resolution camera. A total of 16 features; 12 dimensions and 4 shape forms, were obtained from the grains.

**3. Source:** UCI Machine Learning Repository

*Link:* [Index of /ml/machine-learning-databases/00602 (uci.edu)](https://archive.ics.uci.edu/ml/machine-learning-databases/00602/)

**4. Data Type:** Multivariate

**5. Task:** Classification

**6. Attribute Type:** Categorical, Integer, Real

**7. Format Type:** Matrix

**8. Number of Instances (records in your data set**): 13611

**9. Number of Attributes (fields within each record):** 17

**10. Relevant Information:**

Seven different types of dry beans were used in this research, considering the features such as form, shape, type, and structure by the market situation. A computer vision system was developed to distinguish seven different registered varieties of dry beans with similar features in order to obtain uniform seed classification. For the classification model, images of 13,611 grains of 7 different registered dry beans were taken with a high-resolution camera. Bean images obtained by computer vision system were subjected to segmentation and feature extraction stages, and a total of 16 features; 12 dimensions and 4 shape forms, were obtained from the grains.

**11. Attribute Information:**

1.) Area (A): The area of a bean zone and the number of pixels within its boundaries.

2.) Perimeter (P): Bean circumference is defined as the length of its border.

3.) Major axis length (L): The distance between the ends of the longest line that can be drawn from a bean.

4.) Minor axis length (l): The longest line that can be drawn from the bean while standing perpendicular to the main axis.

5.) Aspect ratio (K): Defines the relationship between L and l.

6.) Eccentricity (Ec): Eccentricity of the ellipse having the same moments as the region.

7.) Convex area (C): Number of pixels in the smallest convex polygon that can contain the area of a bean seed.

8.) Equivalent diameter (Ed): The diameter of a circle having the same area as a bean seed area.

9.) Extent (Ex): The ratio of the pixels in the bounding box to the bean area.

10.) Solidity (S): Also known as convexity. The ratio of the pixels in the convex shell to those found in beans.

11.) Roundness (R): Calculated with the following formula: (4piA)/(P^2)

12.) Compactness (CO): Measures the roundness of an object: Ed/L

13.) ShapeFactor1 (SF1)

14.) ShapeFactor2 (SF2)

15.) ShapeFactor3 (SF3)

16.) ShapeFactor4 (SF4)

17.) Class (Seker, Barbunya, Bombay, Cali, Dermosan, Horoz and Sira)

* Data Batches for Training

1. Download the data (Excel sheet data)
2. Read data in jupyter notebook using pandas Library.
3. Splitting data into three batches training data, testing data, validation set.
4. Using **sklearn** library for splitting data into train and test data sets.
5. Shuffled data while splitting or making batches.

* Data Validation

1.from training data splitting data for validation

2.uses **sklearn** library for splitting data in random state = 42

* Data Transformation

1. Read dataset from excel to pandas dataframe

2. Then split that into train, test, validation datasets.

3. converted those datasets into dictionary format using ‘to\_dict’ pandas inbuild method.

* Data Insertion in Database

1. Then Inserted that data into MongoDB database.
2. Used “pymongo” python module.
3. Created one database with name “project-080422”
4. Created three collections “train\_data”, “test\_data”, ”validation\_data”.

* Export data from database in CSV format.

1. After that fetched data form mongoDB using same library pymongo.
2. And save it as CSV format. For data model training.

* Data processing.

1. Reading CSV file data using pandas lib
2. Getting information about data. how may Nan values, data type of each columns. Etc.
3. Uses .info() , .describe() methods for understanding this data.
4. Plotting correlation plot of data features for getting correlation information with each feature with other features. Using matplotlib subplots method.

## Scaling data using StandardScaler (*from sklearn.preprocessing import StandardScaler*)

1. Scaling the train data

## Data Clustering

1. Importing KMeans for clustering (*from sklearn.cluster import KMeans*)
2. Keep n\_clusters = 7 , because we know that we have 7 unique classes in dataset

## Get best model of each cluster

1. Defining function optimal\_k\_means to get best models If we don’t know the clusters numbers confirm. We can try any numbers of clusters in that function.

* Defining model

1. Defining RandomForestClassifier (*from sklearn.ensemble import RandomForestClassifier*).
2. Fitting train data to model
3. Scaling test data before using in model prediction.

* Hyperparameter Tuning

1. Using GridSearchCV for Hyperparameter tuning (*from sklearn.model\_selection import GridSearchCV*)
2. The best parameters are {'criterion': 'entropy','max\_depth': 8,'max\_features': 'sqrt','n\_estimators': 40}
3. Best estimator RandomForestClassifier(criterion='entropy', max\_depth=8, max\_features='sqrt' n\_estimators=40)
4. After tuning with best parameter score is around 91.5656% of model

* Saving model

1. Using pickle or joblib we can save our model in binary format
2. Save model as ‘model.pkl’ file

Cloud Setup:

1. Cloud setup.
   1. Create webpage using Flask and .html file
   2. Created one EC2 instance on AWS account.
   3. Download and key .pem file for further use. To connect to EC2 instance

Through windows.

* 1. Please install Putty and WinSCP in your local machine or laptop.

1. Open WinSCP from your search bar
   1. Provide your Host name, user name and password. To
   2. Host name you can get from AWS EC2 instance. Just select instance and click on connect button on top. Then go to to SSH client tab, there you can get.
   3. Connect to your instance using its Public DNS:

ec2-3-110-121-10.ap-south-1.compute.amazonaws.com

Graphical user interface, text, application, email

Description automatically generated

* 1. You can get username from EC2 instance connect tab which Is shown in below diagram. You can check this.Graphical user interface, text, application, email

     Description automatically generated
  2. For password you and use your .pem file which you got from aws .

Graphical user interface

Description automatically generated

Click on advance and select Authentication key.

And browse your Private key file (.pem) file in search bar.

Click on okay.

Again okay convert to pkt file. And use it .

Graphical user interface, text, application

Description automatically generated

Click on Login and are connected to instance of EC2.

Graphical user interface, text, application

Description automatically generated

* 1. You will get an tab having two partitions on left side and right side .

The left side one consist folder showing files having in your local machine like.html templets app.py or any other file and Right side folder showing files/ folder on AWS .

You just simply right click and click on upload to file which required to deployed on aws folder.

Graphical user interface, text, application

Description automatically generated

Click on Open Session in Putty

You will automatically get open putyy terminal with connection of your ec2 instance . No need to open again open putty separately.

Graphical user interface, application, Word

Description automatically generated

This is your EC2 as shown.

Text

Description automatically generated

* 1. Write some command on this terminal to install python on AWS machine.

After connect to server

$ sudo apt install python3

$ sudo apt-get update && sudo apt-get install python3-pip

$ pip3 install -r requirements.txt (which will install all listed libraries in text file)

$ python3 app.py - to simply run webpage until putty cmd is opened. Once we closed it the webpage will shut down, automatically.

$ screen -R deploy python3 app.py -- to deploy app on web permanently even if you closed the putty the website will run continuously.

1. Now you can simply copy your public DNS and paste in url link search bar and at lastly to that address simply add (:8080)

Eg. ec2-3-110-121-10.ap-south-1.compute.amazonaws.com:8080

Graphical user interface, application, Word

Description automatically generated

1. Great your model is pushed on web. Thanks ,

Hurrrey…………

Now you can give inputs values in respected fields and check the predict.

1. After some modification in HTML page we can seen more beautiful screen.

Please check the link:

http://ec2-3-7-45-153.ap-south-1.compute.amazonaws.com:8080