***GLASS QUALITY PREDICTION ARTICLE***

***By Gaurav Singh mewari***

***Batch: DS2309***

******

*The Phenomenon of reading this article is step wise:*

The article contains the following sub-topics:

1-Problem Definition

2-Data Analysis

3-EDA Concluding Remarks

4-Pre-Processing Pipeline

5-Building Machine Learning Models

6-Concluding Remarks

***Let’s Begin:***

1. ***Problem Statement:***

*Que: Try to predict the type of glass,* *Attribute Information-*

*1.  Id number: 1 to 214*

*2.  RI: refractive index*

*3. Na: Sodium (unit measurement: weight percent in corresponding oxide, as are attributes*

*4-10)*

*4. Mg: Magnesium*

*5. Al: Aluminum*

*6. Si: Silicon*

*7. K: Potassium*

*8. Ca: Calcium*

*9. Ba: Barium*

*10. Fe: Iron*

*11. Type of glass: (class attribute)*

1. ***Data Analysis***
2. Importing Libraries:

Importing pandas for data manipulation and analysis (import pandas as pd), Importing NumPy for the performing complicated mathematical calculation for large data sets (import numpy as np)

And more important libraries like:

Matplotlib

Seaborn

From sklearnmodels base, TransformerMixin

For pre-processing, FunctionTransFormer,StandardScaler

**from** sklearn.model\_selection **import** (train\_test\_split, KFold , StratifiedKFold,

cross\_val\_score, GridSearchCV,

learning\_curve, validation\_curve)

**from** sklearn.metrics **import** accuracy\_score

**from** sklearn.neighbors **import** KNeighborsClassifier

**from** sklearn.tree **import** DecisionTreeClassifier

**from** sklearn.linear\_model **import** LogisticRegression

**from** sklearn.svm **import** SVC

**from** sklearn.base **import** BaseEstimator, TransformerMixin

*# To create a box-cox transformation class*

**from** collections **import** Counter

**import** warnings

warnings**.**filterwarnings('ignore')

*here: Option + Return to fetch the libraries and adding a new row*

And many more important libraries to ease the work on data and most probably like will not get any kind of errors, which we suppose.

**Que: What are Libraries?**

library is a collection of pre-written code that you can use to perform specific tasks. Libraries are often used to reduce the amount of code a programmer needs to write by providing reusable functions or classes that can be called upon as needed.

1. After importing the libraries, we need to forward one more step which is Data Importing:

Firstly, we need raw data in our file manager or we can upload it in our jupyter notebook to fetch

Like we can do this;

Data=pd.read\_csv(‘Glass Identification.csv’,header =none)

Data

(*Option + Return to fetch the libraries and adding a new row)*

*Outcome:*

It is depended on the data, if data have row name or column name, it will show neither not manually you check this in the excel files or raw form but here we are for the data analysis so we have to do this with some code.

***3-EDA Concluding remarks***

After fetching the data, we have to move one step further which is EDA (Exploratory Data Analysis)

We have data which is named as Data ok.

Assuming the null values:

#calculating number of nulls

Data.isnull().sum()

Output:

If data has null value, it will show you neither not.

Let’s Assume the shape of given data

#Shape of Data

Data.shape()

Output:

Which the shape data contains shows here.

after assuming the null values and shapes of data we assume the unique values

#unique values in table

Data(10).unique()

Output:

Unique values are shown here.

Now we are going to check the data types,

#checking Data types,

Data.dtypes

Output:

We get the containing types of data which values can be:

Int,float,objects

Let’s Go for the checking unbalanced Dataset with bargraph

Data[10].value\_counts()

Sns.histplot(x=10,data=data)

Output:

If data is unbalanced it will show you some gaps in graph and if it not then no gaps will be show

In my case data is unbalanced.

Here I use SNS which is Seaborn library ,imported above

***4-Pre-Processing Pipeline***

Attribute information:

The information which is given in the data.now you can do required transformation in the data sets.

Like adding column name, removing unnecessary columns etc.

We can do some Statistics on Datasets:

Data.describe()

Output:

It will show you what is the count, mean, standard deviations, minimum, 25%, 50%,75% and maximum values in dataset.

We have to normalize the data first as they are not in same scale range.

**Ready for Data Visualization;**

#checking skewness: skewness shows the data is inclining or declining.

There is some coding to check the skewness and you can do this in the either whole data or columns.

After checking the skewness in the whole dataset or column we can predict the what data want to describe with the help of skewness.

It will show you Outlier or can say high skewness.

Now using multivariate Plots for visualization.

Multivariate plots will tell you the designed to reveal the relationship among several variables simultaneously. Now let’s connect with the multivariate plots;

X2=pd.DataFrame(x)

Plt.figure(figsize=(8.8))

Sns.pairplot(data=x2)

Plt.show()

Output: here you can see lots of scattering figure and shows the raw.

After showing the scattering features of the data now let’s jump to the Correlation matrix ;

**Que: what is correlation matrix?**

**Ans: Let’s assume you have two variables in the data sets and you want to establish the relation between them in a statistical way, now you will use the correlation matrix.**

You have to clear the table sequence or you can this raw because we cleaned our data further.

Now let’s do some coding.

#seprating class labels and features

Features= [‘\_\_’,’\_\_\_’……………]

Label= [‘\_\_\_\_\_\_’]

X= data [Features]

Y=data [Label]

Now correlation is:

Correlation=X.corr()

Plt.figure(figsize=(15,15))

Sns.heatmap(correlation,cbar=True,Square=True,xticks=features,yticks=label)

Plt.show()

Output:

You will see the whole data is converted into the heatmap, which shows correlation between the data.

Now detect the Outliers

Again the question is

**Que: What is outlier**

**Ans: Outlier is like a above or below thinking value, they are the data points which is significantly different from the observation.**

**Let’s skip the coding of outliers.**

Jump to the Data Treatment:

It is just information of data:

More steps:

Removing Outliers

Normalize The Data

Scaling The Features

**Visualization of Data after Being Pre Processed**

*5-Building Machine Learning Models*

Start with train\_test\_split:

**Que: What is Train Test Split?**

**Ans: This is Machine Learning technique which split the data sets**

**Into two subsets: Training set and Testing set.**

Training models contains:

K-NN

DECISION TREE

REGRESSION MODEL

SVM CLASSIFIER

Let’s Summarize the Training models

KNN (K – Nearest Neighbours): This is based on the idea that observations

Closest to a given data point are the most similar observations in a data set.

DECISION TREE: A decision tree is a decision support hierarchical model that

uses a tree like model of decision and their possible consequences, including

chance event outcomes, resource costs and utility.

REGRESSION MODEL: Regression model can use a line or a plane to represent

The relationship between the variable.

Regression models can show whether changes in the dependent variable are

Associated with changes in the independent variables.

SVM CLASSIFIER: This is a supervised machine learning algorithm used for both

Classification and regression. this is a powerful machine learning algorithm used

For linear or nonlinear classification, regression and even outlier detection tasks.

*6-Concluding Remarks*

Here you get the result according to your algorithms like what appropriate algorithm you are using and also you have to mention all the things which is output by given algorithms.

***Presented by:***

***Gaurav Singh Mewari***

***Batch: DS2309***