**Himanshu Kumar**

**Assignment (Linear Regression & Logistics Regression)**

**Q1-Write a Python program using Scikit-learn to split the iris dataset into 80% train data and 20% test data. Out of total 150 records, the training set will contain 120 records and the test set contains 30 of those records. Train or fit the data into the model and calculate the accuracy of the model using the Logistics Regression Algorithm.**

**ANS:-**

!pip install pandas

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

from sklearn import datasets

from sklearn.preprocessing import StandardScaler

***#Load the data***

iris=datasets.load\_iris()

iris.keys()

x=iris.data

y=iris.target

***#Split arrays or matrices into train and test subsets***

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.20, random\_state=0)

lgr=LogisticRegression(C=100, random\_state=0,tol=0.01)

***#standerdize the data***

sc=StandardScaler()

x\_train\_std=sc.fit\_transform(x\_train)

x\_test\_std=sc.fit\_transform(x\_test)

lgr.fit(x\_train\_std, y\_train)

y\_pred=lgr.predict(x\_test\_std)

***# Calculate the accuracy of the model***

y\_pred=lgr.predict(x\_test\_std)

***output:-***

Percentage accuracy: 96.66666666666667

**We can see that the accuracy is too high so that we can use this model.**

**Q2- Write a Python program using Scikit-learn to split the attached dataset (Income & expenses data) into 80% train and 20% test data. Train or fit the data into the model and calculate the accuracy of the model using the Multiple Linear Regression Model?**

**ANS:-**

**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**%matplotlib inline**

**from sklearn.model\_selection import train\_test\_split**

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

**inc\_exp=pd.read\_csv("Inc\_Exp\_Data.csv - Inc\_Exp\_Data.csv - Inc\_Exp\_Data.csv - Inc\_Exp\_Data.csv.csv")**

**inc\_exp**

***# As Mthly\_HH\_Expense is dependent variable and rest is independent variable***

x=inc\_exp.drop('Mthly\_HH\_Expense', axis=1)

y=inc\_exp['Mthly\_HH\_Expense']

***#Split arrays or matrices into train and test subsets***

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.20, random\_state=0)

lm=LinearRegression()

lm.fit(x\_train,y\_train)

y\_pred =lm.predict(x\_test)

y\_pred

***# importing r2\_score module***

from sklearn.metrics import r2\_score

from sklearn.metrics import mean\_squared\_error

***# predicting the accuracy score***

score=r2\_score(y\_test,y\_pred)

print('r2 score is==' ,score)

print('mean\_sqrd\_error is==',mean\_squared\_error(y\_test,y\_pred))

print('root\_mean\_squared error of is==',np.sqrt(mean\_squared\_error(y\_test,y\_pred)))

***Output:-***

r2 score is== 0.6508611126071502

mean\_sqrd\_error is== 51568686.515142396

root\_mean\_squared error of is== 7181.134068873968

***# Calculate the accuracy of the model***

print("Accuracy of the model:")

print(lm.score(x\_test,y\_test))

***Output:-***

0.6508611126071502

**We can see that the accuracy is high so that we can use this model.**

**Q3- What is the difference in the conventional programming and Machine learning Programming?**

**ANS:-**

**Conventional Programming** is writing a program in a traditional procedural language, such as assembly language or a high-level compiler language (C, C++, Java, JavaScript, Python, etc).

**(ML)** is the [scientific study](https://en.wikipedia.org/wiki/Branches_of_science)of [algorithms](https://en.wikipedia.org/wiki/Algorithm) and [statistical models](https://en.wikipedia.org/wiki/Statistical_model) that [computer systems](https://en.wikipedia.org/wiki/Computer_systems) use to effectively perform a specific task without using **explicit instructions**, relying on **patterns** and inference instead. It is seen as a subset of [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence). Machine learning algorithms build a [**mathematical model**](https://en.wikipedia.org/wiki/Mathematical_model) based on sample data, known as “[**training data**](https://en.wikipedia.org/wiki/Training_data)”, in order to make predictions or decisions without being explicitly programmed to perform the task.

**Conventional Programming**decision making is based on **IF-ELSE**. Many solution can not be modeled with conventional programming because of the variation of the data input variable and complexity of the problem.

**Machine Learning**solves this problem by modeling this data with**train data** and **test data** and then **predict** the result.

**Q4- Explain various steps involved in Machine Learning Process?**

### ANS:-

### The 7 Steps of Machine Learning:-

**1 - Data Collection**

* The quantity & quality of your data dictate how accurate our model is
* The outcome of this step is generally a representation of data (Guo simplifies to specifying a table) which we will use for training
* Using pre-collected data, by way of datasets from Kaggle, UCI, etc., still fits into this step

**2 - Data Preparation**

* Wrangle data and prepare it for training
* Clean that which may require it (remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.)
* Randomize data, which erases the effects of the particular order in which we collected and/or otherwise prepared our data
* Visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis
* Split into training and evaluation sets

**3 - Choose a Model**

* Different algorithms are for different tasks; choose the right one

**4 - Train the Model**

* The goal of training is to answer a question or make a prediction correctly as often as possible
* Linear regression example: algorithm would need to learn values for *m* (or *W*) and *b* (*x* is input, *y* is output)
* Each iteration of process is a training step

**5 - Evaluate the Model**

* Uses some metric or combination of metrics to "measure" objective performance of model
* Test the model against previously unseen data
* This unseen data is meant to be somewhat representative of model performance in the real world, but still helps tune the model (as opposed to test data, which does not)
* Good train/eval split? 80/20, 70/30, or similar, depending on domain, data availability, dataset particulars, etc.
* **6 - Parameter Tuning**
* This step refers to *hyperparameter* tuning, which is an "artform" as opposed to a science
* Tune model parameters for improved performance
* Simple model hyperparameters may include: number of training steps, learning rate, initialization values and distribution, etc.

**7 - Make Predictions**

* Using further (test set) data which have, until this point, been withheld from the model (and for which class labels are known), are used to test the model; a better approximation of how the model will perform in the real world

**Q5- Describe the evaluation techniques used for Linear Regression?**

**ANS:-**

**Evaluation metrics for a linear regression model:-**

Evaluation metrics are a measure of how good a model performs and how well it approximates the relationship. Let us look at**MSE, MAE, R-squared, Adjusted R-squared, and RMSE.**

## Mean Squared Error (MSE)

The most common metric for regression tasks is MSE. It has a convex shape. It is the average of the squared difference between the predicted and actual value. Since it is differentiable and has a convex shape, it is easier to optimize. MSE penalizes large errors.

## Mean Absolute Error (MAE)

This is simply the average of the absolute difference between the target value and the value predicted by the model. Not preferred in cases where outliers are prominent. MAE does not penalize large errors.

## R-squared or Coefficient of Determination

This metric represents the part of the variance of the dependent variable explained by the independent variables of the model. It measures the strength of the relationship between your model and the dependent variable.

## Root Mean Squared Error (RMSE)

This is the square root of the average of the squared difference of the predicted and actual value. RMSE penalizes large errors.

## Adjusted R-squared — selection criterion

The main difference between **adjusted R-squared**and R-square is that **R-squared** describes the amount of variance of the dependent variable represented by every single independent variable, while **adjusted R-squared** measures variation explained by only the independent variables that actually affect the dependent variable.

**Q6- Describe the evaluation techniques for Logistics Regression?**

**ANS:-** Considering logistic regression is used for Binary Classification, you can use the following measures for evaluating your model on stratified validation data:

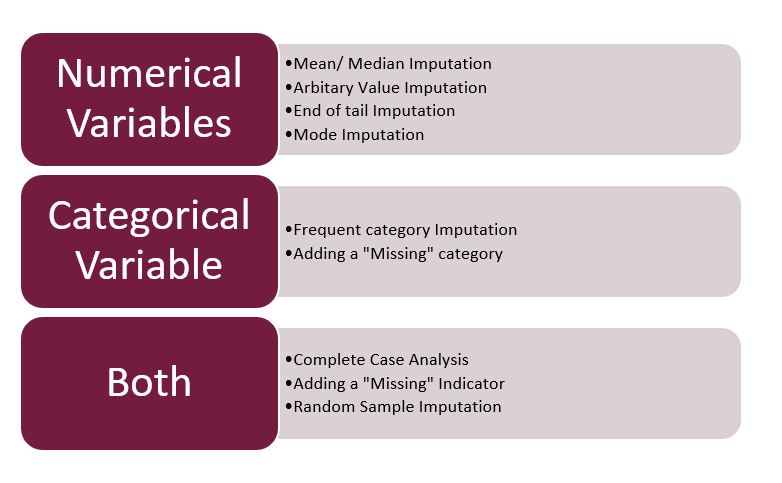
1. Accuracy Score
2. Recall: Recall can be thought of as a measure of a classifiers completeness. A low recall indicates many False Negatives.
3. Precision: Precision can be thought of as a measure of a classifiers exactness. A low precision can also indicate a large number of False Positives.
4. F1 Score, if you are dealing with a imbalanced dataset this is a must. If you are looking to select a model that is balanced between precision and recall F measure should be used.

For Multinomial Logistic Regression I would suggest to use the classification report. You can use the above scores too, precision, F measure with macro.

**Q7- What do you understand by imputation of missing values. Write a note on the methods of Imputation techniques?**

**ANS:-** Imputation is a technique used for replacing the missing data with some substitute value to retain most of the data/information of the dataset.

**Methods of Imputation techniques:-**



**Q8- Discuss some of the ways we can find the normality of the Numerical data?**

ANS:- For normal data, **Q-Q plots** tend to plot data points in almost a straight line. Some sample points with smallest and largest values may stray farther from the line than points between the lower and upper quartiles. Fit to a straight line is usually better for larger samples. Usually, one uses Q-Q plots (also called 'normal probability plots') to judge normality by eye-perhaps without doing a formal test.

A **boxplot** is not really intended as a way to check for normality. However, boxplots do show outliers. Normal distributions extend in theory to ±∞,±∞, even though values beyond μ±kσμ±kσ for k=3k=3 and especially k=4k=4 are quite rare. Consequently, very many extreme outliers in a boxplot may indicate non normality--especially if most of the outliers are in the same tail.

**Q9- What is meant by balanced and unbalanced data?**

ANS:- A balanced dataset is **a dataset where each output class (or target class) is represented by the same number of input samples**. Balancing can be performed by exploiting one of the following techniques: oversampling. undersampling. class weight.

Imbalanced data refers to **those types of datasets where the target class has an uneven distribution of observations**, i.e one class label has a very high number of observations and the other has a very low number of observations.

**Q10- Which of the following statement is TRUE.**

1. Outliers should be identified and removed always from the dataset.
2. Outliers can never be present in the testing dataset.
3. Outlier is a data point which is significantly close to other data points.
4. The nature of our business problem determines how outliers are used.

**ANS:- Option (D) is true.**

**Q11- In Regression modelling we develop a mathematical equation that  
describes how, (Predictor-Independent variable, Response-Dependent variable)**

 (A) one predictor and one or more response variables are related.

 (B) several predictors and several response variables response is related.

 (C) one response and one or more predictors are related.

 (D) All of these are correct.

**ANS:- Option (C) is true**