ADG TASK – DATA SCIENCE

Data Cleaning:

1. First I loaded the CSV given files into the dataframe and set the index to ‘Loan\_ID’
2. Manually replaced all the categorical data to labelled data (label encoding).
3. Removed all the ’NaN’ objects.
4. Separated the Loan payers’ and Non payers data.
5. Saved the updated data into new files.
6. Using ‘MinMaxScaler()’ function of the ‘sklearn.preprocessing’ class, scaled the data to improve the prediction accuracy of the model. That is some data might be very large and mislead the model i.e. add weight towards it’s side. So scaling all the data from large quantity to between 0 to 1 reduces this risk.
7. Set the index back to ‘Loan\_ID’ after scaling.
8. Saved the scaled data.

EDA:

1. Loaded the saved CSV data into the dataframe and set the index to ‘Loan\_ID’.
2. Then printed all the columns present.
3. Printed all the unique values present in each column
4. Used ‘describe()’ method to describe the the data – both train and test. This method describes the data statistically.
5. Described the separated Loan payers’ and nonpayers data.

Model:

1. Loaded all the cleaned data into the dataframe with index ‘Loan\_ID’.
2. Got the labels after One Hot Encoding (OHE) only the categorical feature using ‘OneHotEncoder()’ function of the ‘sklearn.preprocessing’ class.
   1. What OHE does is encodes the data in pattern of present-not-present fashion, i.e. it takes all the unique categorical feature values into the column and encodes each data entry with ‘0’ or ‘1’ indicating the presence or absence of that feature in the data entry.
   2. This is used against Label encoding because, label encoding may confuse the Training model by considering the labels as priorities. This situation is omitted by the OHE Method. By default, the encoder derives the categories based on the unique values in each feature.
   3. Alternatively, you can also specify the categories manually. This encoding is needed for feeding categorical data to many scikit-learn estimators, notably linear models.
3. The first model used is KNN – k Nearest Neighbours Model:
   1. In this Model ‘k’ denotes the number of nearest neighbours to be considered.
   2. The Model calculates the distance between the test data and the fed data and based on the label of k closest neighbours predicts the value of the test data. (mode).
   3. The efficiency of the model increases first and decreases as the ‘k’ value is increased starting, 1.
   4. This Model is implemented using ‘KNearestNeighbours()’ function (structure) of the ‘sklearn.neighbors’ class.
4. Next Model used is Logistic regression Model:
   1. Logistic regression Model is used to determine relation between one binary variable and one or more ratio-level independent variables. It is a linear Model.
   2. The Model is used in situations where the Model is needed to predict a binary value, i.e. a ‘yes’ (1) or a ‘no’ (0). It is useful in our application as we have to predict a ‘Y’ or a ‘N’ – customer may pay or not the Loan.
   3. This Model is implemented using ‘LinearRegression()’ function (structure) of the ‘sklearn.linear\_model’ class.
5. As we can see the application of Scaling the data may improve the prediction accuracy of the model (KNN) or not (LR).
6. A OHE data is also taken into the Model and the solution is got. The accuracy using the OHE is the highest for both the Models. (Because it uses single weight concept).