**ADG task3**

**Data cleaning-**

* Replaced all the NAN(not available) values with the median of the values calculated from the corresponding columns using fillna() function.
* The unwanted rows are removed from the data frame with the dropna() function for which the values can’t be imputed.
* Both the above operations are performed for both train and test datasets.

**Label Encoding-**

* Converted the values under certain columns(Categorical data)to numbers(binarization of values).
* In the data set given gender column’s male and female is encoded to 0s and 1s.

**One hot encoding-**

* Split the column which contains numerical categorical data to different columns depending on the number of categories present in that column. Each column contains 0 or 1 corresponding to which column it has been placed.
* Gender column is split into two columns male and female.
* Dependents column is split into four columns 0,1,2,3+.
* Code-

ttrain = pd.DataFrame( oe.fit\_transform(dfe[['Gender','Married','Dependents','Education','Self\_Employed','Property\_Area']]).toarray(), columns=['Male','Female','unmarried','Married','Independent','1Dependent','2Dependents','3+Dependents','Graduate','Ungraduate','Not\_Self\_Employed','Self\_Employed','Rural','Semiurban','Urban'])

**Test Train Split-**

* Split the training set into two parts. The first part is used to train the model and the second is used to test the accuracy of the model.
* Code-

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.4, random\_state=1)

If there are 200 rows, then 0.4x200 is produced as the test data.

**Building Model for ML-**

* Create the model for machine learning using KNN and Logistic Regression models

**Logistic regression model:**

* from sklearn.linear\_model import LogisticRegression – import from sklearn
* logReg = LogisticRegression() - create an object of model
* logiReg.fit(x\_train, y\_train) – train model

**K Nearest Neighbours model:**

* from sklearn.neighbors import KNeighborsClassifier – import from sklearn
* knn = KNeighborsClassifier(n\_neighbors=3) – create an instance of model
* knn.fit(x\_train, y\_train) – train model

**Predicting output-**

* Now we Predict the output for the second split part of the data set(Test).
* Check accuracy for the created model.