ITM740 Assignment 2

Winter 2002

Objective:

Construct classification models and evaluate the performance with appropriate metrics using loan data. These models include logistic regression, decision tree and random forest etc.

Jupyter Notebook, the web-based interactive computing platform, shall be used for notebooks, code, and data.

Dataset:

loanTrain.csv

Tasks:

1. Import libraries, modules and methods
2. Read data and understand data

   Check the columns, data types shape, and the value counts of “Loan_Status”.
3. Perform missing value imputation
   a. List out feature-wise count of missing values

      Hint: train.isnull().sum()

   b. Fill missing values in Gender, Married, Dependents, Credit_History, Self_Employed and and Loan_Amount_Term features using the mode of the features.

      Hint: train['Gender'].fillna(train['Gender'].mode()[0], inplace=True)

   c. Use the median to fill the null values of the LoanAmount variable.

      Hint: train['LoanAmount'].fillna(train['LoanAmount'].median(), inplace=True)

   d. Check whether all the missing values are filled in the dataset.
4. Treat outlier and generate dummy variables
   a. do the log transformation to 'LoanAmount' to treat outlier and show the histogram

      Hint: train['LoanAmount_log']=np.log(train['LoanAmount'])

   b. Drop the Loan_ID variable as it does not have any effect on the loan status

      Hint: train=train.drop('Loan_ID',axis=1)

   c. Make dummy variables for the categorical variables. The dummy variable turns categorical variables into a series of 0 and 1.
      Hint: train = pd.get_dummies(train)
5. Discover through feature engineering

a. Create a new feature “Total Income” which is the combine the Applicant Income and Co-applicant Income. Take the log transformation to make the distribution close to normal.

b. Create a new feature “EMI” which is the monthly amount to be paid by the applicant to repay the loan.

c. Create a new feature “Balance Income” which is the income left after the EMI has been paid.

d. Drop the variables which we used to create these new features.

Hint: train=train.drop(['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan_Amount_Term'], axis=1)

6. Prepare data before classification
   a. Drop target variable from the training dataset and save it in another dataset.
   b. Divide train dataset into training and validation part.
7. Apply logistic model
   a. import LogisticRegression and accuracy_score from sklearn and fit the logistic regression model
   b. predict the Loan_Status for validation set and calculate its accuracy.
8. Apply Perform decision tree model.
   a. train the model with the help of DecisionTreeClassifier class of sklearn.
   b. get the accuracy score, confusion matrix and classification report:
9. Your findings and conclusion

Submission:

You must use Jupyter Notebook and provide an ipynb file with both markdown and codes. You don’t have to submit the dataset. Also don’t submit compressed files.

Take use of different levels of heading and numbering when using Markdown in Jupyter Notebook.

The name of the files shall be ITM740Assignment2byAliceBrown.ipynb where “AliceBrown” shall be replaced by your first name and last name.