dun & bradstreet

Run Book

Runbook Name	Loan Default Prediction
Runbook Description	Building Predictive Model for Loan Default Assignment for Job Role: Data Scientist
Owner	Dileep Sathyan
Version	v1.0
Version Date	10-Aug-2022
Overview	

Requirements:

Language	Python3.7 or above

Libraries	• Numpy
	• Pandas
	• Matplotlib
	• Seaborn
	• Scikitlearn
	  ○ preprocessing
	  ■ LabelEncoder
	  ○ model_selection
	  ■ train_test_split
	  ■ Cross_val_score
	  ○ ensemble
	  ■ RandomForestClassifier
	  ○ metrics
	  ■ confusion_matrix

Import Libraries & Dataset:

Import libraries		• Import all above mentioned requirements
Import dataset		• Import the dataset using pandas from the file location.

dun & bradstreet

Data Cleaning:

Drop fields & records	Duplicate fields	• Find all NULL fields and drop them
	Duplicate records	• Find duplicated records and drop them • Find the records with NULL 'SANCTION_AMT' and drop.
Renaming the fields	All	• Rename all the fields appropriately.
Datatype corrections	Datefields	• Correct the DATE fields into right datetime formats.
Drop multiple entries	ACCOUNT_ID	• Find multiple records for same ACCOUNT_ID and keep only the latest record.

Handling Missing Data:

Find missing values	All	• Calculate the total missing values in each field
Filling missing values	'REPAYMENT_FREQUENCY'	• For LOAN_TYPE == 'Overdraft', fill in the frequency as 'AtTheFinalDayOfThePeriodOfContract'
	'NUMBER_OF_INSTALLMENTS'	• Calculate the difference in days between "DATE_EXP_LOAN_CLOSURE' & 'DATE_OPEN' and divide it appropriately as per the 'PAYMENT_FREQUENCY'

	'INSTALLMENT_AMT'	• Calculate it by dividing 'NUMBER_OF_INSTALLMENTS' from 'SANCTION_AMT'
	'OVER_DUE_AMT'	• Fill 0 OVER_DUE for Accounts which have LOAN_STATUS' as "TerminatedAccordingTheContract' • Fill 0 for those Accounts with DPD == 0 • For the rest of the missing cases, fill them with their 'INSTALLMENT_AMT'
	'TOT_OUTSTD_BAL'	• For DPD >0 customers, their Oust_Bal must be greater than Over_Due. However, we shall consider the same amount here as we dont have the Total_Paid on date • For Accounts with DPD == 0, fill in the outstanding amount with their 'INSTALLMENT_AMT'
	'LOAN_TYPE'	• Fill in the missing values using the mode of the field.
	'DPD'	• Accounts with valid LATEST_PAYMENT_DATE, if the OVER_DUE exists, find the days_diff from reporting date and fill the DPD • Accounts with 'REPAYMENT_FREQUENCY' = 'AtTheFinalDayOfThePeriodOfContract', fill the 'DPD' as 0 • Accounts with 'REPAYMENT_FREQUENCY' = 'MonthlyInstalments30Days', if the 'OVER_DUE_AMT' exists, check if the 'OVER_DUE_AMT' / 'INSTALLMENT_AMT' is >=3. If yes, fill 'DPD' wit 91, else with 0

Data Manipulations:

Add new field	'LOAN_TENURE'	• Calculate it afresh from subtracting 'DATE_ACCT_OPEN' from 'DATE_EXP_LOAN_CLOSURE'
Convert Amounts to 1 currency	'SANCTION_AMT', 'TOT_OUTSTD_AMT', 'OVER_DUE_AMT', 'INSTALLMENT_AMT'	• Convert all these 4 field values for Accounts with 'CURR_CODE' = 'USD' by multiplying with USD to TZS Exchange rate as 2335
Data Normalization / Log transformation	'SANCTION_AMT', 'TOT_OUTSTD_AMT', 'OVER_DUE_AMT', 'INSTALLMENT_AMT'	• Transform all the Amount fields by taking their log values to enable the model to predict correctly.
Add Dependant Variable	'IS_DEFAULT'	• Calculate this new field from 'DPD' by flagging all Accounts with 'DPD' >= 90
Feature Selection	'REPAYMENT_FREQUENCY', 'NUMBER_OF_INSTALLMENTS', 'SANCTION_AMT_LOG', 'TOT_OUTSTD_BAL_LOG', 'OVER_DUE_AMT_LOG', 'INSTALLMENT_AMT', 'LOAN_TYPE', 'LOAN_STATUS', 'LOAN_TENURE','IS_DEFAULT'	• Subset the dataframe to have only the given attributes and drop the rest.

Label Encoding:

Encode Categorical Variables	'REPAYMENT_FREQUENCY', 'LOAN_TYPE', 'LOAN_STATUS'	• Encode the categorial variable fields using LabelEncoder from sklearn

Train Test Split:

Splitting the dataset for Training & Testing	All	• Specify the input and output variables as X and y from the dataset. • Split 25% of data for testing and the rest to train the model using train_test_split from sklearn.model_selection

Model Building & Prediction:

Model building and prediction	All	• Build a RandomForest Classifier Model from sklearn.ensemble • Fit the model with X and y datasets • Predict the y variables and check the accuracy score.

Validation Of Predicted Results:

Evaluate using Confusion Matrix	All	• Create a confusion matrix using the predicted and test data. • Plot heatmap using seaborn to visluaise the results.