MLC_B1_C4_Assignment_PrashantBhide

Dataset Reference:

The data set has been taken from the following public S3 links:

1. **Credit card application**: This contains the demographic information of the users who are applying for the credit card.
   https://s3.amazonaws.com/sqoop.oozie.ml/application_record.csv

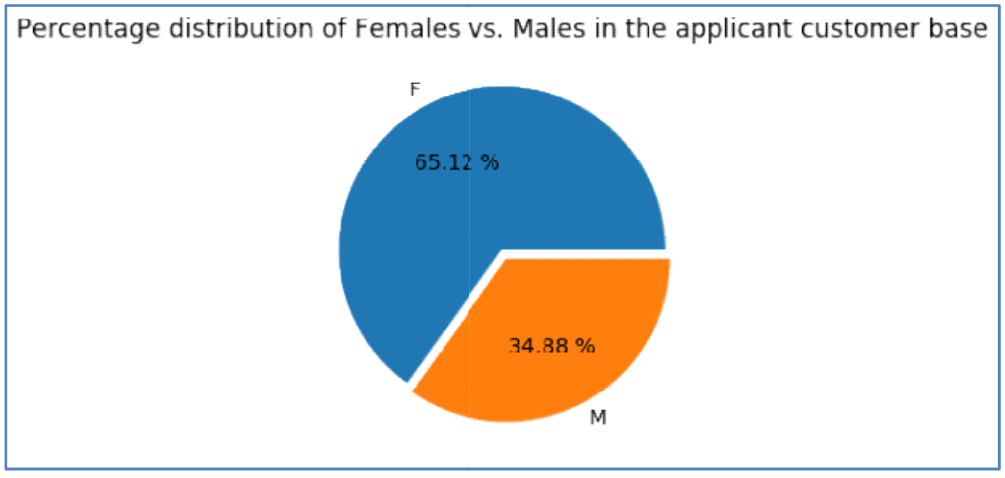
2. **Credit card performance**: This contains the performance of the users after they are issued the credit card.
   https://s3.amazonaws.com/sqoop.oozie.ml/credit_record.csv

Objectives:

1. Perform EDA (Exploratory data analysis) to understand the data set and use well-considered visualisations to unwrap the insights.
2. Perform any required checks such as variable exploration, outlier treatment, missing value imputation, variable transformation and correlation check.
3. Apply the concepts of Weight of Evidence and Information Value to perform variable exploration and variable transformations.
4. Once the data preprocessing is complete, build a credit card application approval model that predicts whether a customer's credit card is delinquent or not, solely based on the customer's application data. Use the credit card performance data to create the target variable, i.e., whether the customer's credit card is delinquent or not. The user ID will be classified as delinquent if the customer has ever delayed their payment by more than 60 days.
5. Fine-tune the model and then evaluate the model by considering various metrics such as precision, recall, F1-score, AUC score and KS statistic.

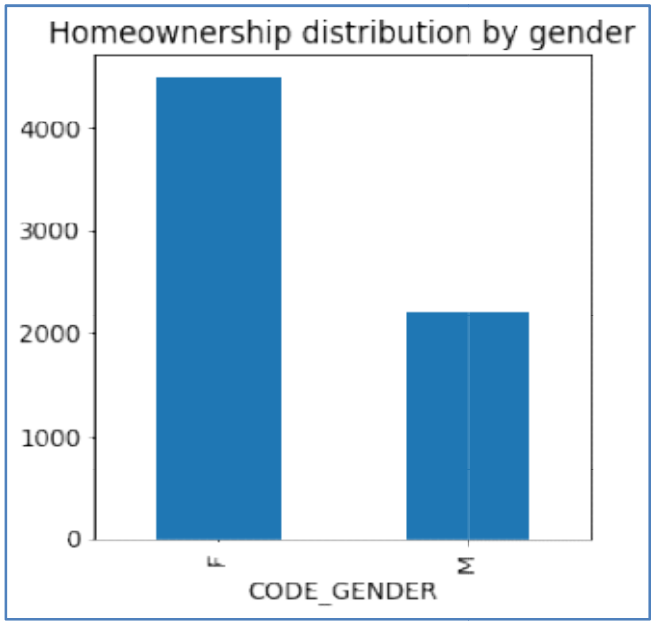
EDA questions:

EDA_1) What is the proportion of females in the applicant customer base?



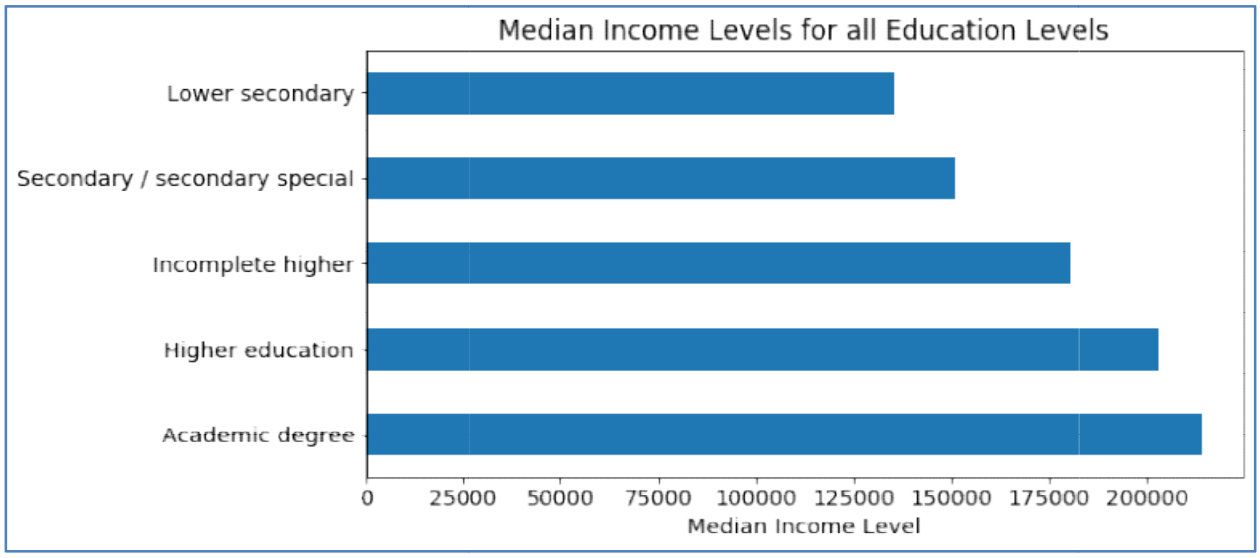
Observation: We can see that Females make up 65.12% of the applicant customer base compared to 34.88% Males

EDA_2) Is homeownership higher among male applicants or female applicants?



Observation: We can see that homeownership is higher for female applicants compared to male applicants

EDA_3) Is there any correlation between the customer's income level and education level?



Observation: We can see a direct correlation between the customer's income level and education level. Higher the education level, higher the median income for the overall applicant base

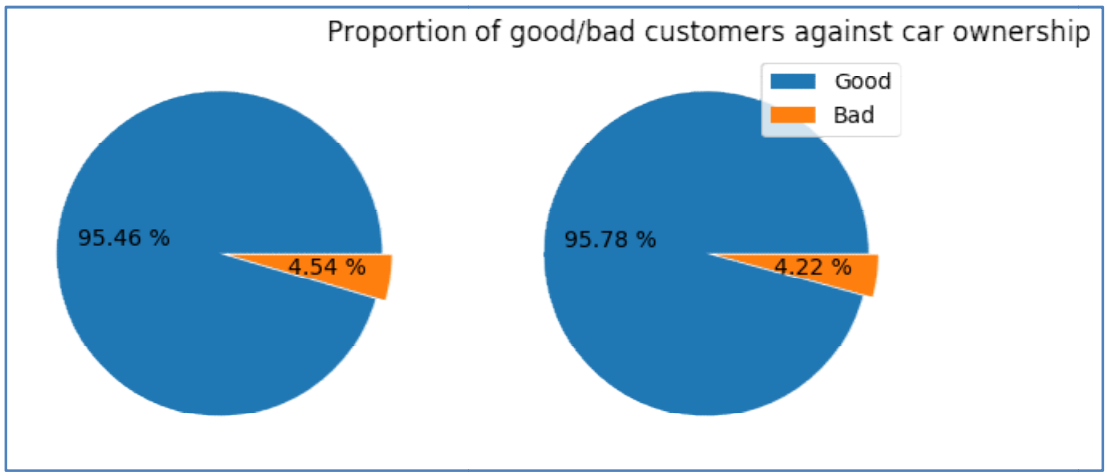
EDA_4) What is the average and median salary of the applicant base?

```
count       9997.0000
```

```
mean      181479.6474
std        99508.1045
min        27000.0000
25%       112500.0000
50%       157500.0000
75%       225000.0000
max      1575000.0000
Name: AMT_INCOME_TOTAL, dtype: float64
```

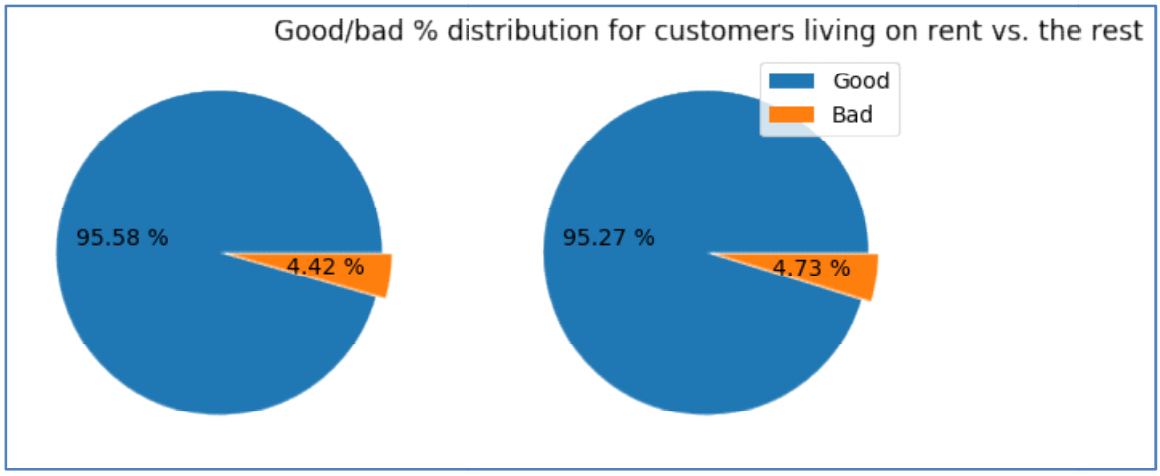
Observation: Based on the Income (salary) statistical distribution, we can see that the average (mean) salary of the applicant base comes to 181479.65, while the median (50th percentile) salary comes out to be 157500.00

EDA_5) Is the proportion of bad customers higher for people who own cars?



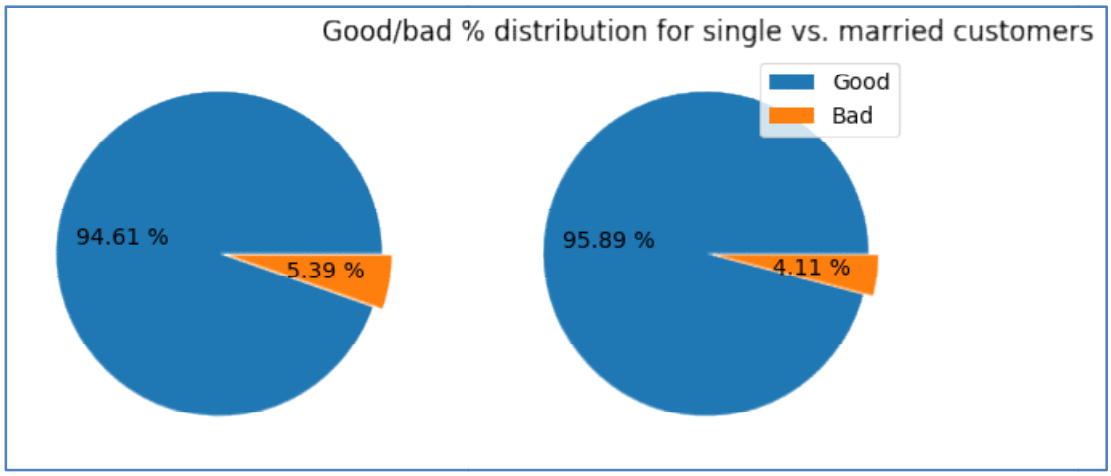
Observation: We can see that the proportion of bad customers is higher for non-car-owners (bad is 4.54%) compared to car-owners (bad is 4.22%)

EDA_6) Is the proportion of bad customers higher for those living on rent than the rest of the population?



Observation: We can see that the proportion of bad customers is higher for renters (bad is 4.73%) compared to non-renters (bad is 4.42%)

EDA_7) Is the proportion of bad customers higher for those who are single than married customers?



Observation: We can see that the proportion of bad customers is higher for single (bad is 5.39%) compared to married (bad is 4.11%) customers

---

Summary of Dataset Preparation, Model Building, and Model Evaluation steps:

1. **Loading and analyzing** the "application_record" dataset

2. **Missing value imputation**: Based on the null value counts, we see that the only feature with missing values is "OCCUPATION_TYPE" with a 30% missing rate. Since this feature can be expected to be closely related to "NAME_INCOME_TYPE", we will go ahead and impute the missing values with the most frequent value within each category as grouped by "NAME_INCOME_TYPE"

3. **Loading and analyzing** the "credit_record" dataset

4. **Transforming the "credit_record" data into a "Delinquent Status" dataset**: Any user ID has to be classified as delinquent if that user has EVER delayed their payment by more than 60 days. From the Credit Application Data Dictionary, this condition can be mapped as the "STATUS" feature EVER having a value within the set(['2','3','4','5']). We will create a user-defined-function to check if the set of all "STATUS" values for each user ID has any overlap or intersection with the delinquent statuses and accordingly mark the final delinquent status for each user ID.

5. **Generating the "credit_issued" merged dataset**: The "credit_issued" combined dataset will be generated with an inner join of the "Delinquent Status" and "application_record" datasets, since this is a Supervised Learning problem and we are only interested in user IDs that have the Delinquent label value already determined.

6. **Duplicate data cleanup**: We can see that the merged dataset has many duplicate rows if we ignore the user ID column. Accordingly we will drop this ID column and then drop duplicate rows from the merged dataset.

7. We can see that the merged and de-duplicated dataset comes out to 9997 rows. The **distribution of the Delinquent label** within this dataset comes out to be 9555 "good" users and 442 "bad" users at a ratio of **95.58% good vs. 4.42% bad users**.

8. **Variable exploration and outlier treatment**

9. We can see that all the numeric features show a proper progression from min to max values except for the "DAYS_EMPLOYED" feature includes significant outliers going by the difference between the 75th percentile and max values. Any such invalid (positive values will be invalid as per the data dictionary) outliers will be converted to null and then imputed with the median value for this feature

10. **Variable transformation and correlation check**

11. We will now convert the continuous numeric value features into ordinal categorical features for subsequent "Weight of Evidence" transformations. We will use pandas quantile binning (10% x 10 chunks) which will automatically coerce any outliers into the first or last bin respective groups

12. **Correlation check**: We can see that the feature "CNT_CHILDREN" is highly correlated with "CNT_FAM_MEMBERS". Accordingly we will go ahead and drop feature "CNT_CHILDREN" from the dataset

13. **Feature Selection and Model Building**

14. Applying the concepts of **Weight of Evidence and Information Value** to perform variable exploration and variable transformations

15. **Building the LogisticRegression Model and Evaluation**

16. Using a **Train-Test split of 70:30** and a Seed value of 2018 (as specified in the Evaluation Rubrics)

17. We can see that the **Recall for positive class (1 == DELINQUENT) is 0.0** which indicates that the trained model is unable to predict any True Positives. We already know that this is a highly unbalanced dataset with a Negative:Positive class ratio of roughly 96%:4%. Accordingly, **we will attempt the concept of "Weight of Class" to re-train the model** by specifying weights in the inverse ratio for the Negative and Positive classes

18. We can see that the **Recall for positive class (1 == DELINQUENT) has significantly improved to 0.617** along with Precision at 0.067 after re-training the model with the **"Weight of Class" hyperparameter tuning !!!**

19. Visualize the Precision-Recall graph for the predicted results

20. KS statistic analysis

21. **Final evaluation metrics for model trained with "Weight of Class" tuning :**
    - **Precision = 0.06789667896678966**
    - **Recall = 0.6174496644295302**
    - **F1_score = 0.12234042553191489**
    - **AreaUnderROC = 0.6064332162603887**
    - **KS_statistic = 0.06610564384323817**