Credit Scoring Analysis

## **Algorithmic Digital Marketing Assignment 2**

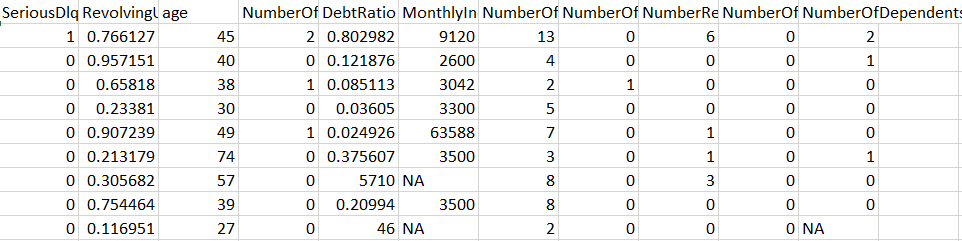
|  |  |
| --- | --- |
| **Summary** | In this codelab, we have analyzed the Credit Card dataset to predict the probability that someone would face financial distress in the next two years. |
| **URL** | <https://www.kaggle.com/c/GiveMeSomeCredit> |
| **Category** | Data analysis and Visualization |
| **Author** | Prathamesh Verlekar and Yash Pandya |

## **Overview**

Banks play a crucial role in market economies. They decide who can get finance and on what terms and can make or break investment decisions. For markets and society to function, individuals and companies need access to credit.

Credit scoring algorithms, which make a guess at the probability of default, are the method banks use to determine whether or not a loan should be granted. This dataset requires us to improve on the state of the art in credit scoring, by predicting the probability that somebody will experience financial distress in the next two years.

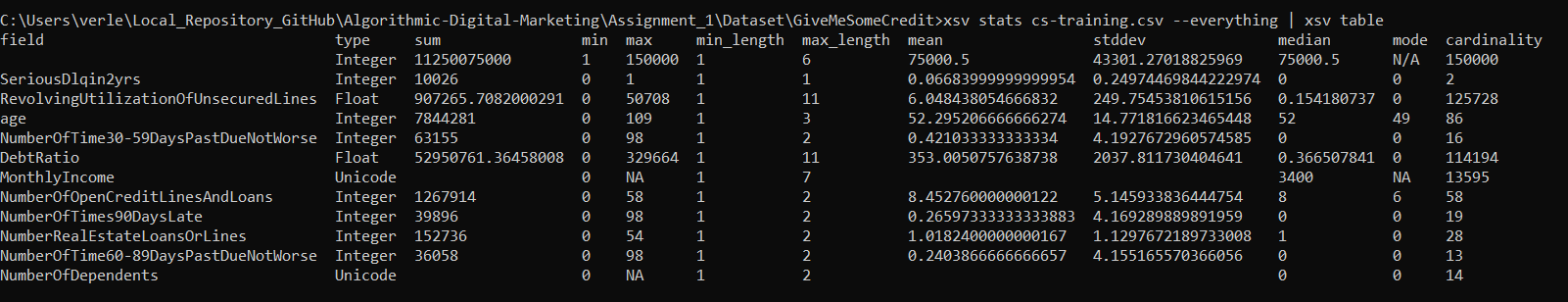
## **Data Sample**



## **About Features**

1. There are a total of 11 features in the dataset (SeriousDlqin2yrs, RevolvingUtilizationOfUnsecuredLines, age, NumberOfTime30-59DaysPastDueNotWorse, DebtRatio, MonthlyIncome, NumberOfOpenCreditLinesAndLoans, NumberOfTimes90DaysLate, NumberRealEstateLoansOrLines, NumberOfTime60-89DaysPastDueNotWorse, NumberOfDependents).
2. The dataset has 1.5 lakh observations.
3. The main feature is SeriousDlqin2yrs, which tells us whether the person experienced 90 days past due to delinquency or worse.
4. All the details of the borrowers are of the past two years only.
5. Two features(RevolvingUtilizationOfUnsecuredLines, DebtRatio) are of the data type percentage and monthly income is of data type real and the rest all are integers.

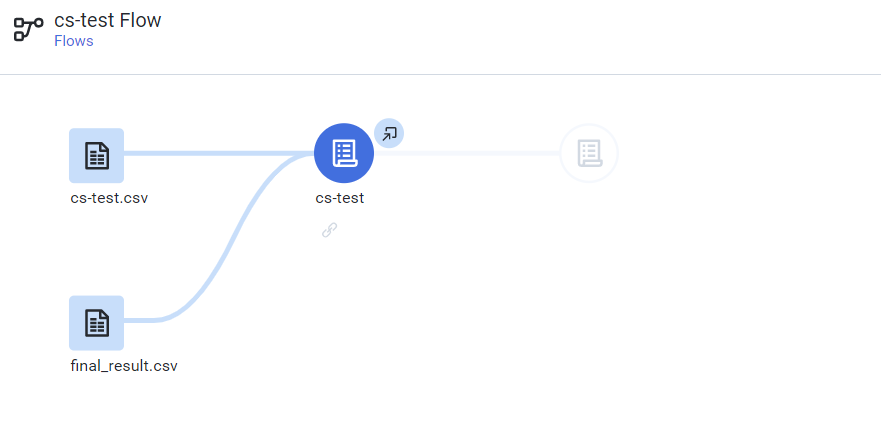
## **Using XSV for Statistical Information**



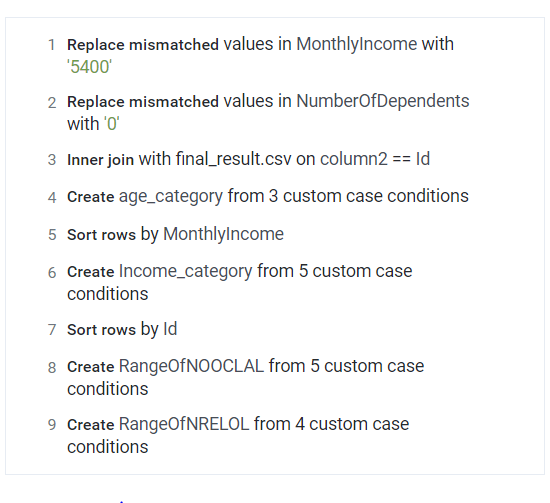
## 

## **Data Wrangling using Trifacta**

**Data Flow**

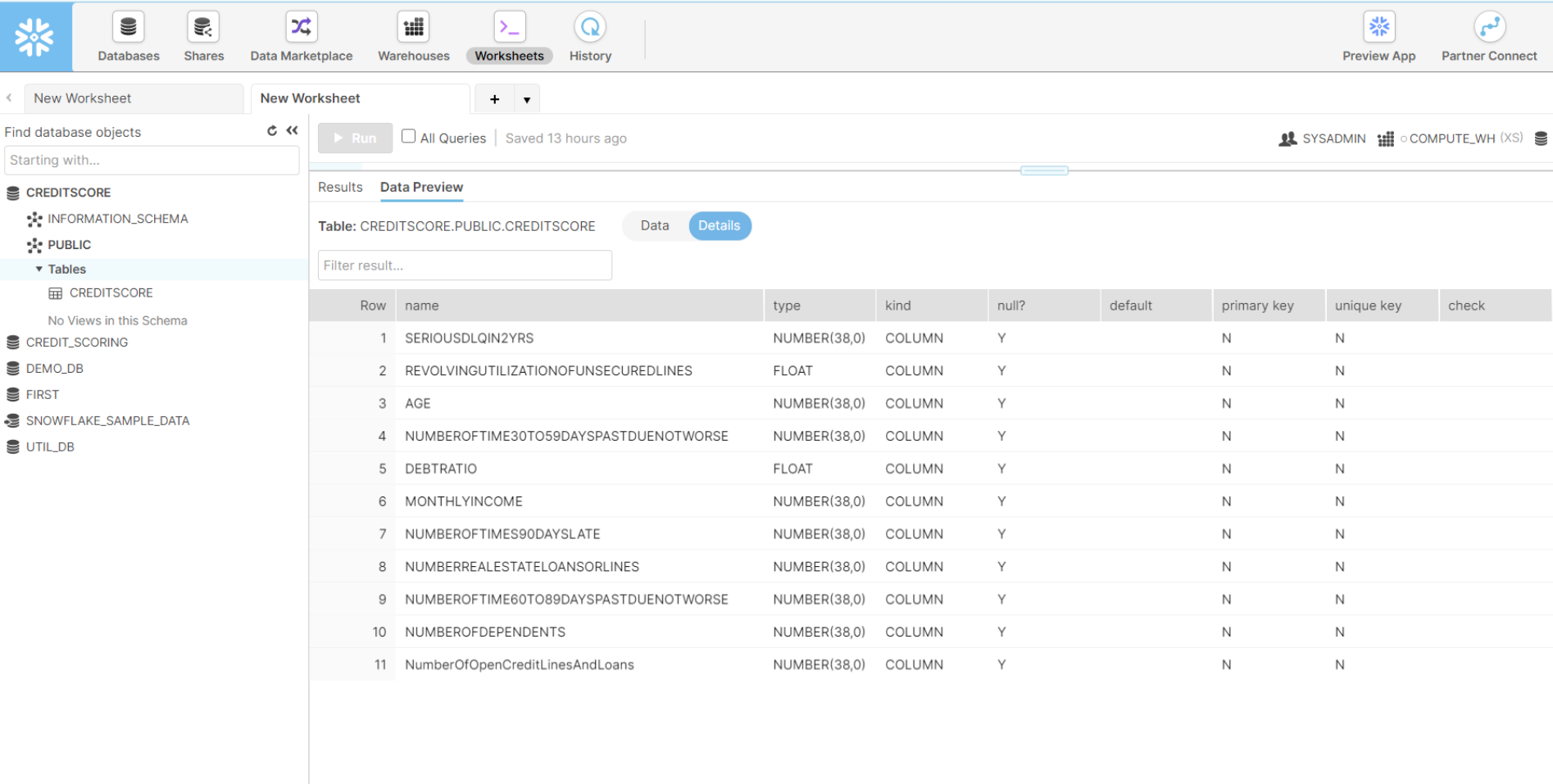
****

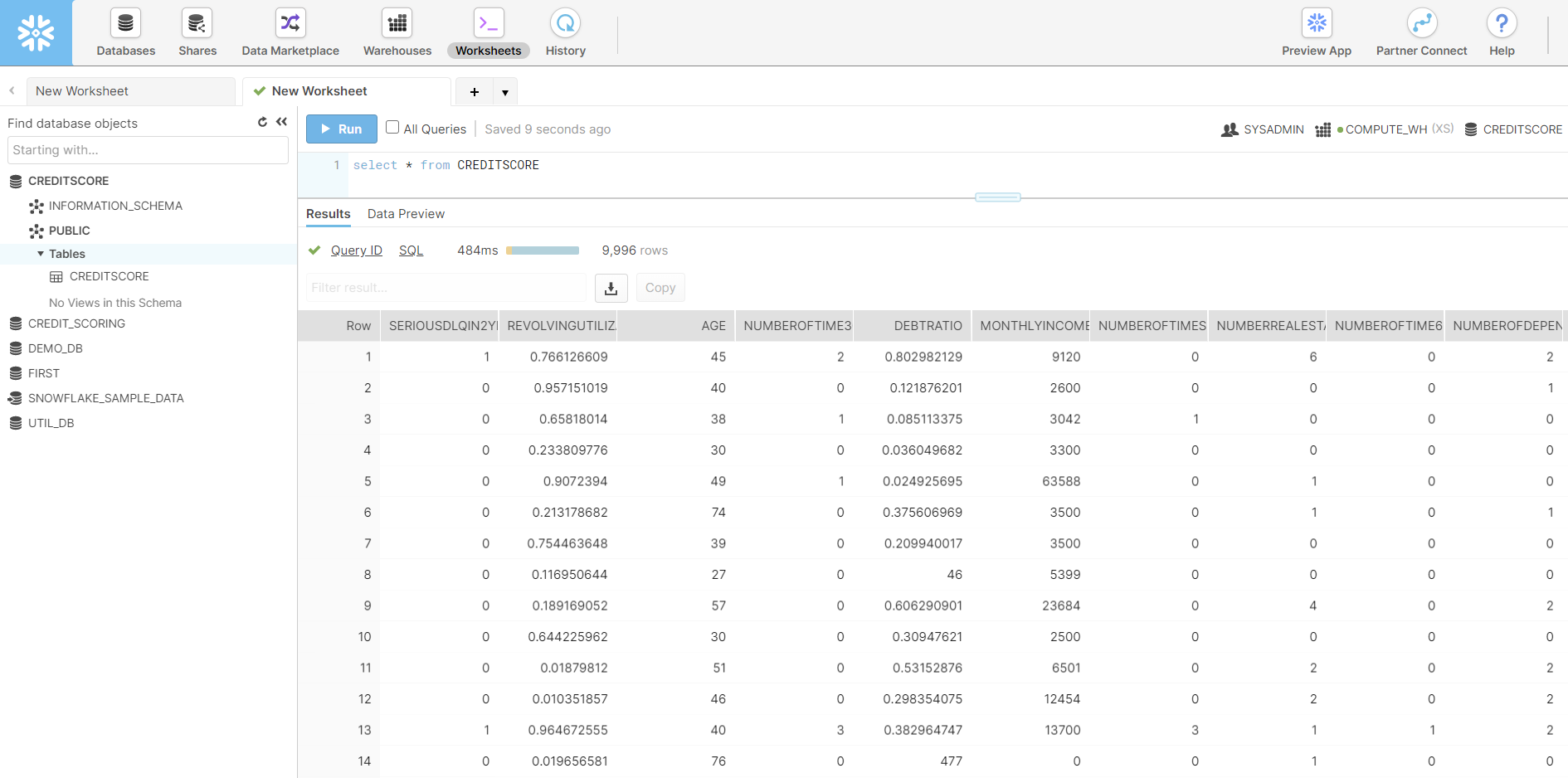
**Recipe**

****

## **Database in Snowflake**

We created a database in Snowflake and loaded the data which we cleaned and preprocessed using Trifacta.





## **Formulations we have used**

1. Replaced mismatched Values in MonthlyIncome with them the median value of the Monthly Income Column.
2. Replaced mismatched Values in the Number Of Dependent Column with the mode.
3. Inner Joined two CSV files together with the common columns.
4. Created Custom category(Age Category) with three conditions where data is distributed age-wise.
5. Created Custom category(Income Category) with five conditions where data is distributed income-wise.
6. The debt ratio is calculated by the division of the Liability and monthly income of the person given in the table.

## 

## **Insights we gained**

1. Around 6.68% of the data had records of defaulters.
2. There were 29731 missing values in Monthly Income and 3924 missing values in NumberOfDependents column.
3. The minimum age of the credit card holder in the dataset is 21 and there are more younger people who are defaulting.
4. For missing values in NumberOfDependents we have inputted mode of the column which is equal to zero.
5. For missing values in MonthlyIncome, since the distribution is skewed we will import the median(5399) of the column.
6. When the value for NumberOfTimes90DaysLate has values above 17 then NumberOfTime30-59DaysPastDueNotWorse, NumberOfTime60-89DaysPastDueNotWorse and

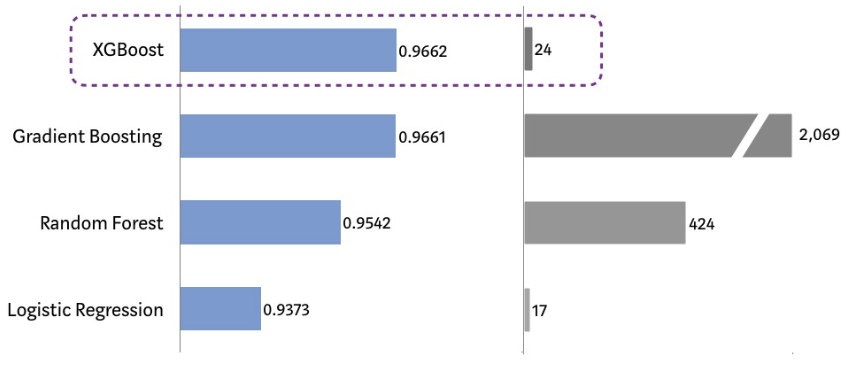
NumberOfTimes90DaysLate has the same values.

1. Since sharing the same values is not possible because 30 days past due 96 times for a single person in a timespan of 2 years is not possible so we removed values above 17 in the NumberOfTimes90DaysLate column.
2. There are around 2.5% of clients that owe more than 3490 or more times than what they own, but it is observed that for those clients mostly we don’t have monthly income and if we have it is either 0 or 1, so we removed values of DebtRatio above its 97.5 percentile.

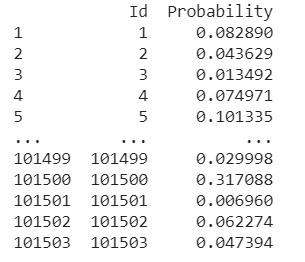
## **Prediction using XGBoost**

**XGBoost -** [XGBoost](https://xgboost.ai/) is a decision-tree-based ensemble Machine Learning algorithm that uses a [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting) framework. In prediction problems involving unstructured data. XGBoost performs well because it ensemble tree methods that apply the principle of boosting weak learners using gradient descent architecture.

**Performance comparison**

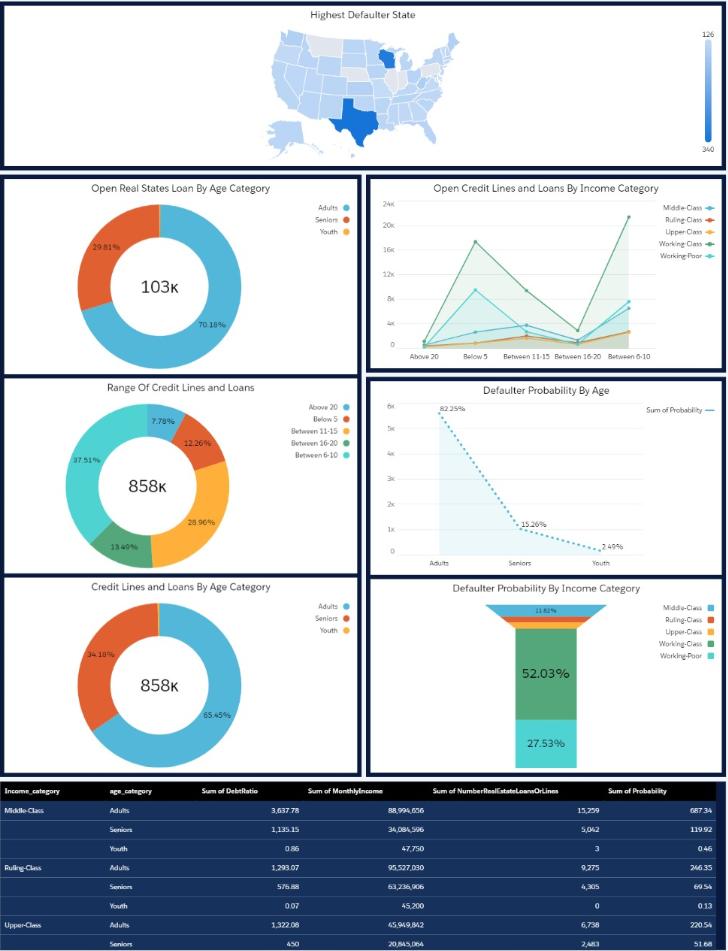
****

**Predicted Probability Sample**

****

## **Insights using Salesforce**

**Main Dashboard**

****

**Detailed Dashboards**

