Final Report

Credit Scoring Model for First-Time Customers

MSBA 5303: Programming for Analytics

University of Central Oklahoma

Completed by Group 8:

James Livingstone
Vladislav Volkov

All group members have contributed equally to this project.

Table of Contents

Executive Summary

Purpose

Flex Data is a data-driven consulting firm that specializes in transitioning our customers through the data literacy journey to enable smart, focused utilization of big data and data-driven solutions for empowered business users (Analytics 2022). Flex Data gathers and stores data for various companies. We create and design workflows that give our clients the ability to look at everything that can provide valuable insights. Our clients have a large volume of data. This data can be beneficial in many ways. It can also pose a challenge to implement those benefits. Flex Data ensures that our clients have high data quality. We also ensure that our clients comply with government regulations.

Every company has both internal and external stakeholders. Striving to deliver the best value to the stakeholders is mandatory for survival in today's marketplace. This is true for every company. Our clients want to improve customer satisfaction and increase their loyal customers base, however, they struggle with inefficiency in their organizational business processes. Our clients frequently hear so much about terminologies like "business process re-engineering", "business process improvement" (BPI), "business process management", but they find it difficult to apply any of them to redesigning and improving their existing business process to innovative new processes that are supported by the organizational environment, culture, and human and technological resources (Samsul, I. & Daud, A. 2012).

As a continuous improvement to our organization. We strive to leverage data and data technology to support and promote digital transformation for our clients. 65% of our clientele are banks that offer credit cards and retailers that provide store cards. Flex Data is searching for ways to improve our services.

Problem Summary

Recent studies have shown that Oklahoma is the 4th state in the country with the lowest average number of credit cards, making it an evolving market for attracting first-time customers who do not have an established line of credit (Tathah, M. 2018). Oklahoma National Credit Union provided our company with a project initiation request for a more objective and calculable approach to be used in the calculation of creditworthiness of their customers who are seeking to establish a line of credit. It has become difficult for them to use solely historical data in predicting credit scores due to fluctuations in the economy. Following modern trends in data analysis, the credit union has initially evaluated the possibility of hiring another consulting company to develop a machine learning predictive model to improve their credit scoring processes. However, after evaluating the cost of deploying and maintaining such a model against the benefits, this option has been ruled out. For this reason, Flex Data has offered an alternative, more cost-effective solution of employing a logistic regression model as an effective way to calculate and understand credit scores. The goal of Flex Data is to implement this model and improve the profitability of Oklahoma National Credit Union.

Solution Summary

Flex Data has gathered two data sets (Seanny 2020), one with the credit card applicant's personal information, and one with credit payment history. It is not only important to know whether a client is financially capable of making regular payments towards their credit, but if the client has a history of accruing overdue expenses. What Flex Data is trying to decipher is what credit score each client deserves based off of these two data sets and advising on increasing, lowering or unaltering their line of credit. The data has been collected through thorough and regulated background checks, and a six-month period of payment status monitoring. Each of the clients, regardless of personal history, had all been given entry-level credit cards and evaluated over a six-month period in order to assess their payment punctuality, and we have used Excel and JupyterLab to assess the data. A system was set-up which assigned a scoring system dependent on payment status. At the end of the six-month period, and after the points system was calculated, clients who were rewarded or remained unaltered were now subject to evaluation through their personal history to determine individual changes to lines of credit, as opposed to this information being omitted during the issuance of the entry-level credit cards. In particular, variances and changes in personal income were used alongside payment history as a means to determine future creditworthiness. Our solution is different from many other practices because we start each one of these new clients off at the exact same entry point regardless of their biographical information, in order to analyze their performance over an initial 6-month period. This allows the clientele to build a credit history as well as the financial institutions to decide if their money is in good hands.

Introduction

Client and Company Description

Oklahoma National Credit Union provided our company with a project initiation request for a more objective and calculable approach to be used in the calculation of creditworthiness of their customers who are seeking to establish a line of credit. This particular credit union was established in 1998 and is registered through NMLS. It has become difficult for them to use solely historical data in predicting credit scores due to fluctuations in the economy. Following modern trends in data analysis, the credit union has initially evaluated the possibility of hiring another consulting company to develop a machine learning predictive model to improve their credit scoring processes. However, after evaluating the cost of deploying and maintaining such a model against the benefits, this option has been ruled out.

Flex Data is a data-driven consulting firm that specializes in transitioning our customers through the data literacy journey to enable smart, focused utilization of big data and data-driven solutions for empowered business users (Analytics 2022). Flex Data gathers and stores data for various companies. We create and design workflows that give our clients the ability to look at everything that can provide valuable insights. Our clients have a large volume of data. This data can be beneficial in many ways. It can also pose a challenge to implement those benefits. Flex Data ensures that our clients have high data quality. We also ensure that our clients comply with government regulations.

As a continuous improvement to our organization. We strive to leverage data and data technology to support and promote digital transformation for our clients. 65% of our clientele are banks that offer credit cards and retailers that provide store cards. Flex Data is searching for ways to improve our services. We determined that we currently have two data sets that could assist us in leveraging data to our clientele and improving their services to their customers. Flex data can create an algorithm that can determine the customer’s creditworthiness and optimize the credit limit amount that will suit their customer’s needs and also their financial capability of making their payments.

Business Problem Details

Problem and Solution Details

The Oklahoma National Credit Union has made its mission to expand further across Oklahoma and become one of the state's largest financial institutions. One of the most prominent set-backs to this expansion has been figuring out how to expand their business model to new clients who have yet to have any credit history. The Oklahoma National Credit Union is wary of opening a line of trust to a population of individuals who have not been able to demonstrate financial aptitude. In doing so without a well-executed model in place risks the financial security, integrity, and future of the Oklahoma National Credit Union.

We at Flex Data feel that our mission is to provide ease of access for financial organizations looking to introduce new clientele into the financial trust and power that is credit. The biggest problem that these organizations have to face is how to build trust with these individuals who have no history of financial accountability (Consumer Financial Protection Bureau 2020).

The data has been collected through thorough and regulated background checks, and a six-month period of payment status monitoring. This data is now present in our two data sets (Seanny 2020), and we have used Excel and JupyterLab to assess the data. Each of the clients, regardless of personal history, had all been given entry-level credit cards and evaluated over a six-month period in order to assess their payment punctuality. A system was set-up which assigned a scoring system dependent on payment status. 0: 1-29 days past due=60 credit points, 1: 30-59 days past due=50 credit points, 2: 60-89 days overdue=40 credit points, 3: 90-119 days overdue=30 credit points, 4: 120-149 days overdue=20 credit points, 5: Overdue or bad debts, write-offs for more than 150 days=0 credit points, C: paid off that month=60 credit points, X: No loan for the month=50 credit points. The monitoring of credit is continuous, so if a client continues to be late on payments and falls into a higher score, their credit will be affected accordingly. At the end of the six-month period each month's credit score is averaged collectively for a final credit score.

At the end of the six-month period, and after the points system was calculated, clients were now subject to evaluation through their personal history to determine individual changes to lines of credit, as opposed to this information being omitted during the issuance of the entry-level credit cards. In particular, variances and changes in personal income were used alongside payment history as a means to determine future creditworthiness.

Our solution is different from many other practices because we start each one of these new clients off at the exact same entry point regardless of their biographical information, in order to analyze their performance over an initial 6-month period. This allows the clientele to build a credit history as well as the financial institutions to decide if their money is in good hands. It is after this trial period that previously discussed scores are calculated and credit is now additionally subjected to each individual's biographical information. Previous research models demonstrate that "Credit scores are a fundamental ingredient of a borrower's access to credit. In the United States, credit bureaus and credit rating agencies serve this function for individual and business credit by creating and maintaining credit scores for individual borrowers" (Chatterjee, S., Corbae, D., Dempsey, K., Ríos-Rull, J.-V. 2020, August). It is the hope of Flex Data, that with the success of our innovative start-up, those without a history of credit might enter the system easily and be able to build their credit faster, for the good of themselves as well as financial institutions

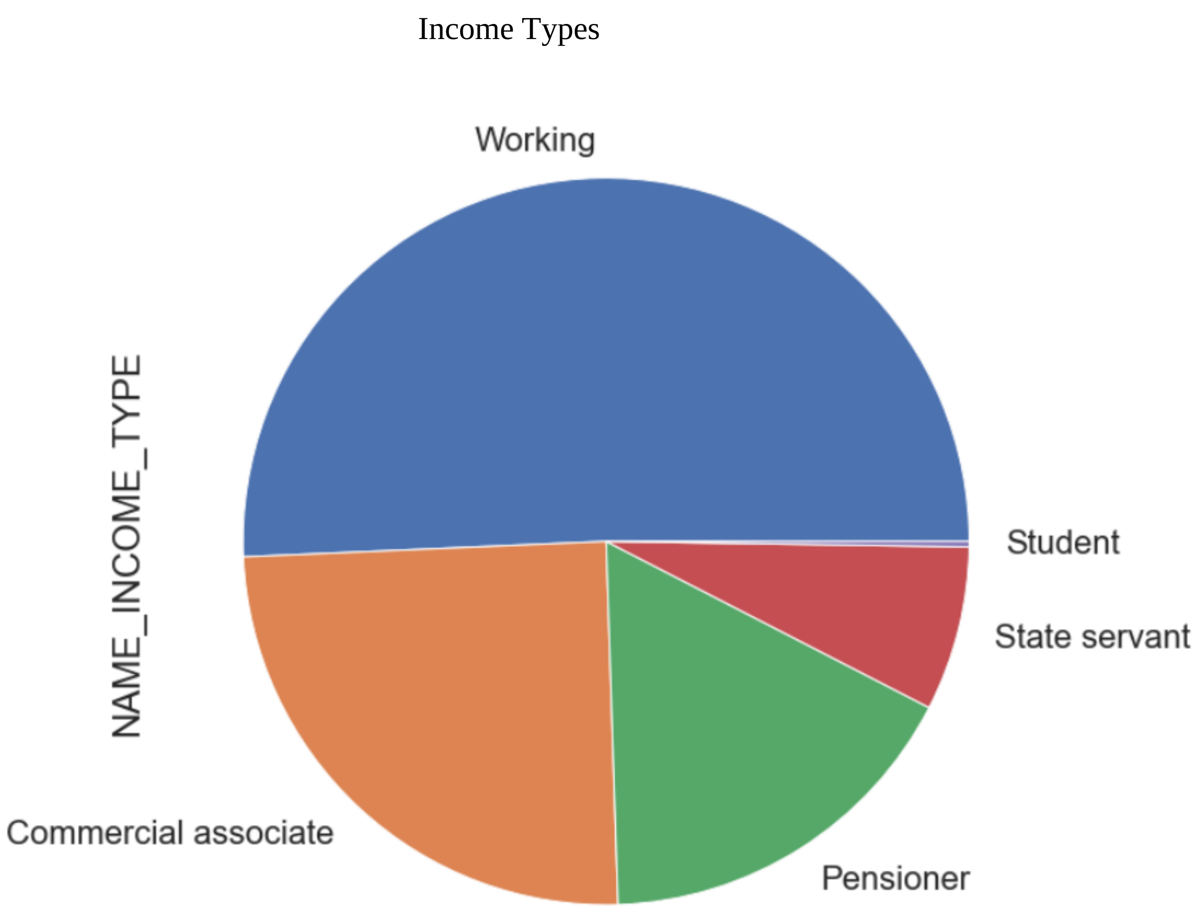


Figure 1: Job Types Ratio among Applicants

Our data clearly displays that most of the applicants fall under the “Working” income type, with 1267 applicants, while the least are in the “Student” income type, with only 6 applicants. As for the other categories, there are 621 “Commercial Associate”, 423 “Pensioner”, and 183 “State servant” applicants.

Income Total vs. Income Type

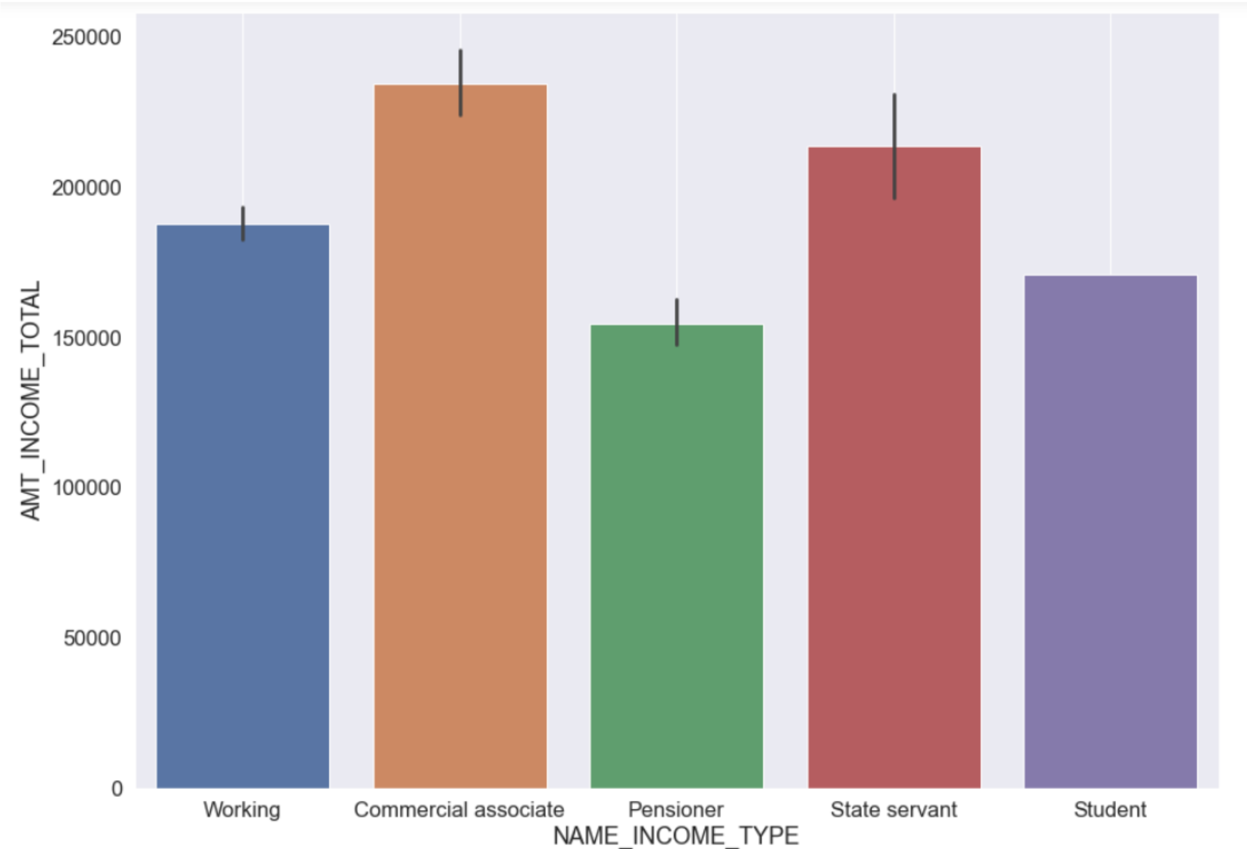


Figure 2: Income Type compared to Income Total

Our data supports the fact that "Commercial associate" applicants make the most amount of income compared to the other categories of professions.

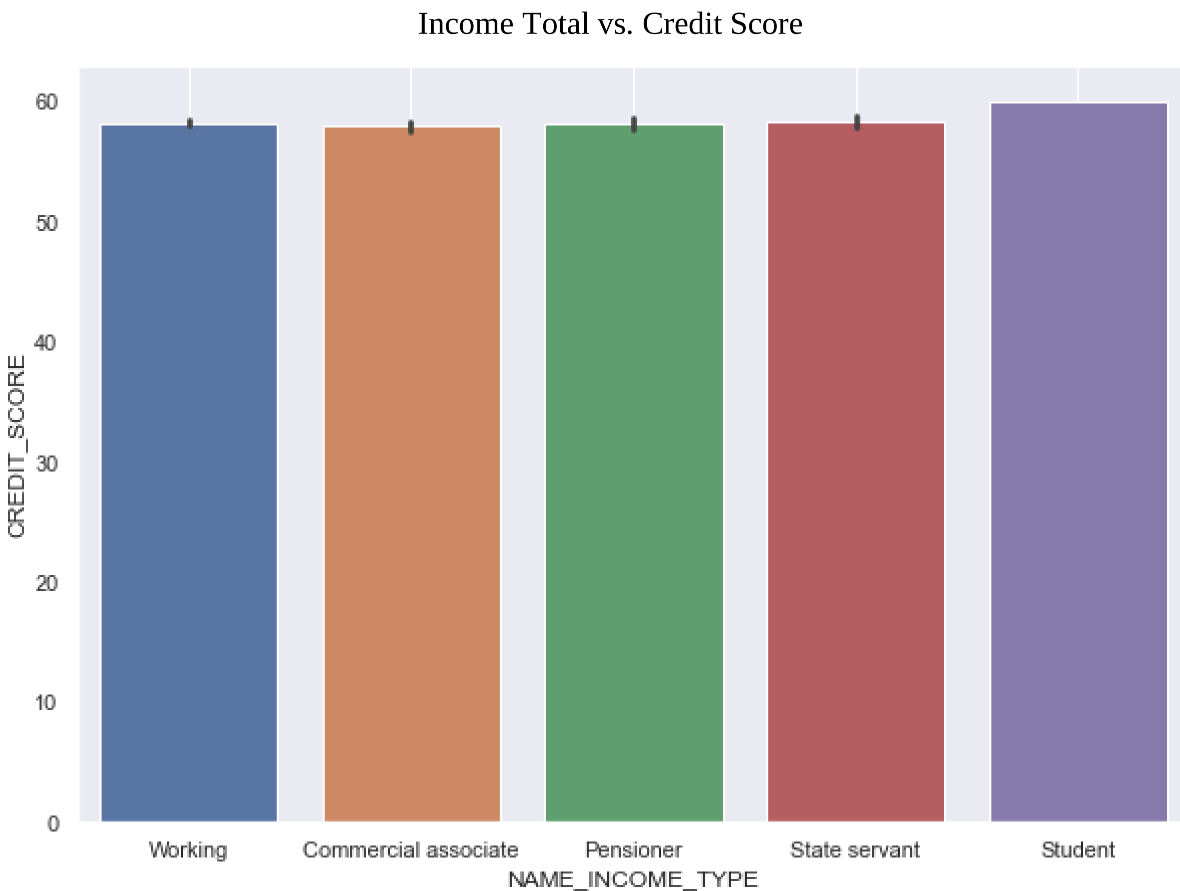


Figure 3: Income Type compared to Credit Score

The disproportion of “Student” applicants contributes to sample size error demonstrated when calculating a correlation between income type and credit score, this makes it seem as though “Student” income type correlates with the highest credit score of 60.00, however the difference in credit score between other income type categories is negligible with “State servant’ income type being ahead of the next highest score by a small margin of 0.09, with a credit score of 58.31.

Income Total vs. Credit Score

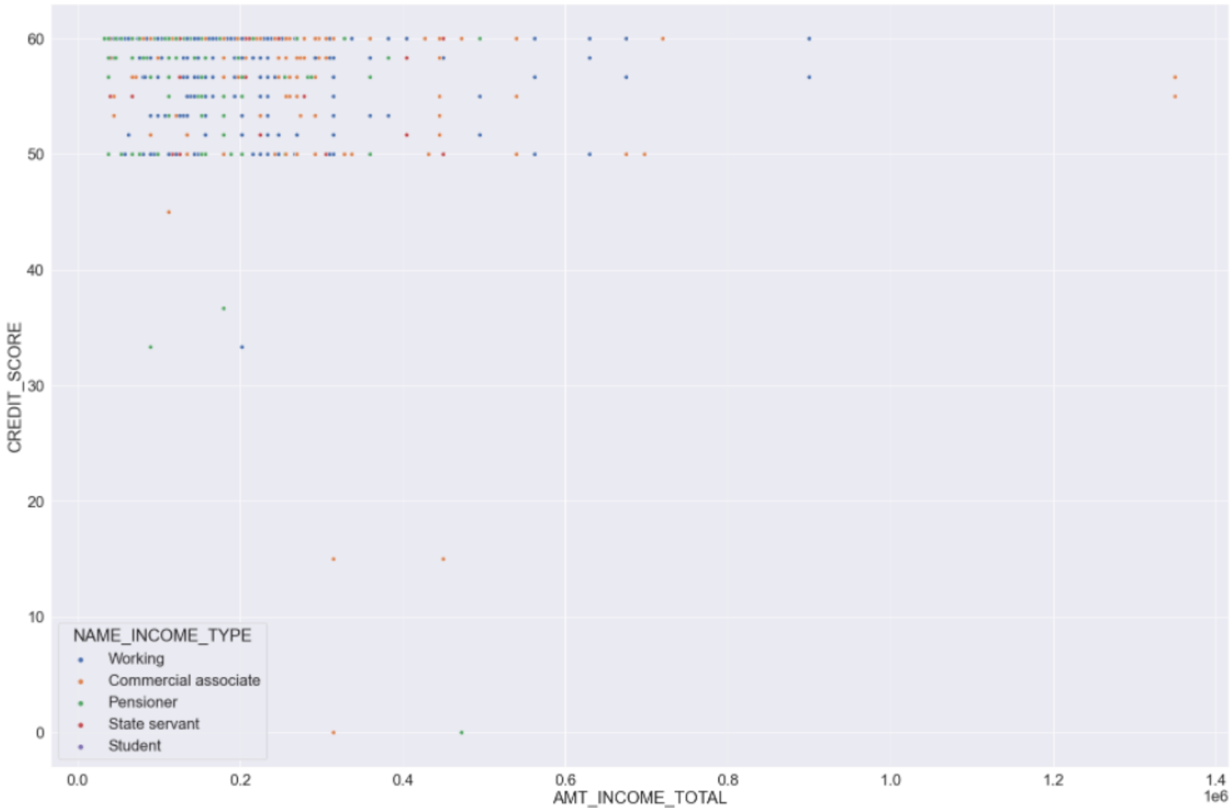


Figure 4: Income Total compared to Credit Score

As demonstrated by the above figure, there is a nearly perfectly uniform correlation between the total income (AMT_INCOME_TOTAL) and the partial credit score, which we calculated based on the 6 months of payment history (CREDIT_SCORE), which suggests that variation earnings amount has a little significant effect on the applicant's payment behavior.

Research Questions

The initial question we wanted to answer was, “Do certain biographical factors hold more weight towards affecting the credit score than others?” Our first impression of our data suggested there were some categories that held more suitable information that could change a credit score. In order to establish which categories to choose from we ran .value_counts() on each category of data to observe levels of variation within each category. The categories that seemed to have the most variation as well as a more obvious predetermination of significance were chosen. When looking through our dataset we singled out income type, family status, housing type, car ownership, realty ownership, and income total for our categories.

The second question we wanted to answer was, “What is the extent of the correlation between certain categories of biographical information and credit score?” The first impression we got from our data suggested that income type, family status, housing type, car ownership, realty ownership, and income total all had a significant effect on overall credit score. Through our analysis we discovered that the biographical information categories do not affect credit score to as significant of a degree than we had previously thought. Income total and Income type for example produced nearly uniform correlations when compared to credit score. Each of the other aforementioned categories produced similar results, showing little significant effect on credit score.

The third question is “Should we consider biographical data given its little significance?” Through our calculations in the previous questions and through our data analysis we discovered that with our particular demographic, biographical information held no significant effect on credit score. Due to these reasons biographical data should not be considered for our particular demographic.

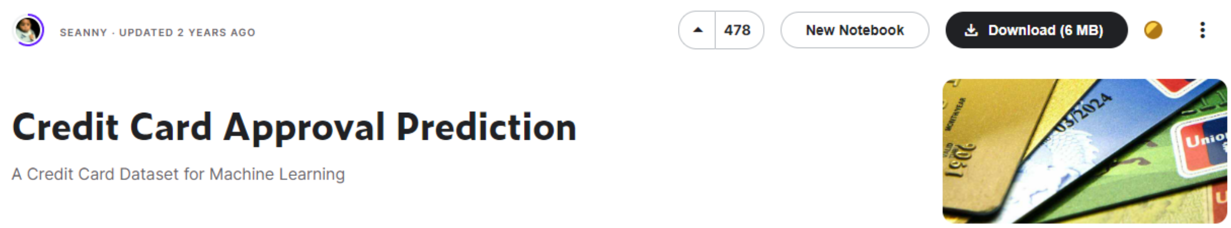
As may be observed from the questions' summaries above, our research goal has evolved throughout the duration of the study. We started out with a strong belief that personal records play a vital role in the applicant's spending habits and abilities to pay the debts back based on the previous studies by other data scientists. As the study progressed, we started observing hints that this may not be the case for the specific audience we are working with, which prompted further questions to be investigated.

Data Understanding

Obtaining Data

When approached by the Oklahoma National Credit Union, firstly, we set forth to establish clear objectives of the project and deliverables. The main deliverable that was to be produced was a credit score evaluation algorithm that could be applied to any future first-time applicants. The challenge in creating such an algorithm is having a dataset with a diverse enough audience that would represent people from different backgrounds, professions and generations. With biographic and credit payments data being private personally identifiable information, finding such a dataset can be challenging. Thankfully, we found a large credit card dataset from different statistic research published on Kaggle.com:

https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction



Initial Data Collection Report

Description of the Data

We have started out with 2 initial datasets: credit_record.csv (1,048,576 rows; 3 columns) and application_record.csv (438,558 rows; 18 rows). The first file contains the credit history by month with payment status info for each month and the second file contains the applicants' biographical information. Both datasets have the "ID" column as the common identifier.

Exploration of the Data

Before diving deeper into data analysis, we have to understand how our datasets are structured, what type of information is contained in each column and what specific values represent. We quickly learned that the value representation in our data is not intuitively understood and had to refer to the project documentation from our data source. After studying the source info and exploring the values in Excel were able to form and document a clear understanding of the datasets. This enabled us to construct the data dictionaries illustrated below:

application-record-dictionary.csv

Feature name	Explanation	Remarks
ID	Client number	-
CODE_GENDER	Gender	M or F
FLAG_OWN_CAR	Is there a car	Y or N
FLAG_OWN_REALTY	Is there a property	Y or N

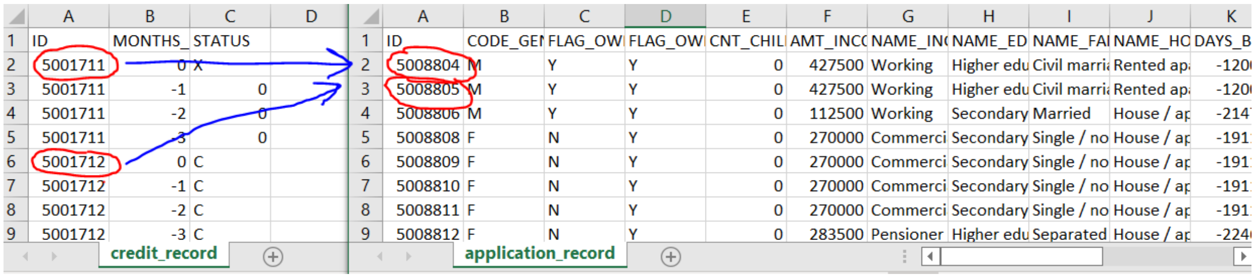
CNT_CHILDREN	Number of children	-
AMT_INCOME_TOTAL	Annual income	-
NAME_INCOME_TYPE	Income category	-
NAME_EDUCATION_TYPE	Education level	-
NAME_FAMILY_STATUS	Marital status	-
NAME_HOUSING_TYPE	Way of living	-
DAYS_BIRTH	Birthday	Count backwards from current day (0), -1 means yesterday
DAYS_EMPLOYED	Start date of employment	Count backwards from current day (0). If positive, it means the person currently unemployed.
FLAG_MOBIL	Is there a mobile phone	If (1) means has mobile phone | If (0) there is no mobile phone
FLAG_WORK_PHONE	Is there a work phone	If (1) means has work phone | If (0) there is no work phone
FLAG_PHONE	Is there a phone	If (1) means has phone | If (0) there is no phone
FLAG_EMAIL	Is there an email	If (1) means has email | If (0) there is no email
OCCUPATION_TYPE	Occupation	-
CNT_FAM_MEMBERS	Family size	-

application-record.dictionary.csv

Feature name	Explanation	Remarks
ID	Client number	-
MONTHS_BALANCE	Record month	The month of the extracted data is the starting point, backwards, 0 is the current month, -1 is the previous month, and so on
STATUS	Status	0: 1-29 days past due 1: 30-59 days past due 2: 60-89 days overdue 3: 90-119 days overdue 4: 120-149 days overdue 5: Overdue or bad debts, write-offs for more than 150 days C: paid off that month X: No loan for the month

Data Cleaning: Rationale for Inclusion and Exclusion of Data

Completing an initial visual assessment of our data in Excel, we have used the sorting tool to organize both of the files by ID column from smallest to largest. Immediately we observed a data discrepancy, because our smallest and largest IDs did not match. From here, we must make an assumption that not all customers' IDs represented in credit-record would be present in application-record and vice versa. Such data discrepancy would make it impossible to calculate the cumulative credit score for all customers, therefore we must limit the datasets to only include applicants with matching IDs between the 2 tables. To do the matching, we have utilized several *VLOOKUP* functions in conjunction with the innate Excel features for sorting and filtering tables.



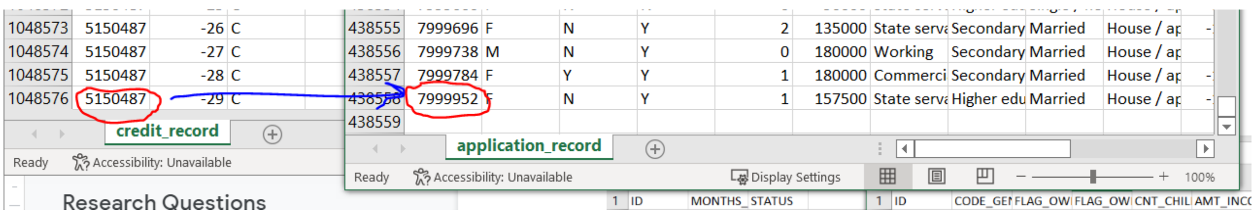
	A	B	C	D
1	ID	MONTHS_	STATUS	
2	5001711	0	X	
3	5001711	-1	0	
4	5001711	-2	0	
5	5001711	-3	0	
6	5001712	0	C	
7	5001712	-1	C	
8	5001712	-2	C	
9	5001712	-3	C	

credit_record

	A	B	C	D	E	F	G	H	I	J	K
1	ID	CODE_GEI	FLAG_OW	FLAG_OW	CNT_CHIL	AMT_INC	NAME_IN	NAME_ED	NAME_FA	NAME_HO	DAYS_B
2	5008804	M	Y	Y	0	427500	Working	Higher edu	Civil marri	Rented ap	-120
3	5008805	M	Y	Y	0	427500	Working	Higher edu	Civil marri	Rented ap	-120
4	5008806	M	Y	Y	0	112500	Working	Secondary	Married	House / ap	-214
5	5008808	F	N	Y	0	270000	Commerci	Secondary	Single / no	House / ap	-191
6	5008809	F	N	Y	0	270000	Commerci	Secondary	Single / no	House / ap	-191
7	5008810	F	N	Y	0	270000	Commerci	Secondary	Single / no	House / ap	-191
8	5008811	F	N	Y	0	270000	Commerci	Secondary	Single / no	House / ap	-191
9	5008812	F	N	Y	0	283500	Pensioner	Higher edu	Separated	House / ap	-224

application_record

<u>Lowest ID Numbers</u>



1048573	5150487	-26	C
1048574	5150487	-27	C
1048575	5150487	-28	C
1048576	5150487	-29	C

credit_record

Ready Accessibility: Unavailable

Research Questions

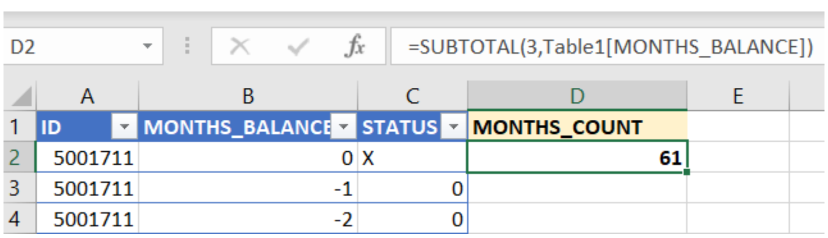
438555	7999696	F	N	Y	2	135000	State serv	Secondary	Married	House / ap
438556	7999738	M	N	Y	0	180000	Working	Secondary	Married	House / ap
438557	7999784	F	Y	Y	1	180000	Commerci	Secondary	Married	House / ap
438558	7999952	F	N	Y	1	157500	State serv	Higher edu	Married	House / ap
438559										

application_record

Ready Accessibility: Unavailable Display Settings 100%

<u>Highest ID Numbers</u>

Furthermore, after removing duplicates based on the MONTHS_BALANCE column in the credit-record table and finding the count of months, we learned that there were customers with up to 61 months of payment history.



D2 =SUBTOTAL(3,Table1[MONTHS_BALANCE])

	A	B	C	D	E
1	ID	MONTHS_BALANCE	STATUS	MONTHS_COUNT	
2	5001711	0	X	61	
3	5001711	-1	0		
4	5001711	-2	0		

This is a significantly longer history than our client bank could ever collect for the first-time credit card applicants, so we must limit the data to no more than 6 months for each unique ID. We have used the *COUNTIF* function to calculate the number of times every ID appears. After studying the results, we found that we also had applicants who have less than 6 months of payment history. Using filtering on the months number and count of months' payments, we limited the datasets to only include applicants with matching IDs in both tables as well as no more and no less than 6 months history. The final version of the datasets was limited to 2,500 rows in the application-record table and 15,000 rows in the credit-record table.

```
In [138]: application_raw.shape
Out[138]: (438557, 18)

In [139]: credit_raw.shape

Out[139]: (955637, 4)

In [140]: application.shape
Out[140]: (2500, 25)

In [141]: credit.shape
Out[141]: (15000, 4)
```

Understanding the Data and Dataset Description

For further understanding of the data we utilized Python via the Jupyter Notebook application. Firstly, we verified our table size:

```
In [6]: application.shape
Out[6]: (2500, 18)

In [113]: credit.shape
Out[113]: (15000, 4)
```

We ran .value_counts() on each of our application data categories in order to see how our output variables compared to each other within each category. This allowed us to validate the diversity of our sample demographic. Although in general the data does have significant variation, we did note that it is skewed in some of the categories. Examples of the skewness include disparities by gender, family status and number of children. The final dataset predominantly consists of women, who represent 66.6% of the population. Addionationally, the majority of applicants are not married (68.8%) and do not have any kids (70.4%). Because of our initial assumption that demographic information does have an effect on credit card use and on time payments, we must take these differences into consideration when analyzing the results. Null values and data types also affect how we can utilize the columns, so we ran .info() to gain more insight into the application_record dataframe:

```
application.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2500 entries, 0 to 2499
Data columns (total 25 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ID                    2500 non-null   int64
 1   CODE_GENDER           2500 non-null   object
 2   FLAG_OWN_CAR          2500 non-null   object
 3   FLAG_OWN_REALTY       2500 non-null   object
 4   CNT_CHILDREN          2500 non-null   int64
 5   AMT_INCOME_TOTAL      2500 non-null   int64
 6   NAME_INCOME_TYPE      2500 non-null   object
 7   NAME_EDUCATION_TYPE   2500 non-null   object
 8   NAME_FAMILY_STATUS    2500 non-null   object
 9   NAME_HOUSING_TYPE     2500 non-null   object
 10  DAYS_BIRTH            2500 non-null   int64
 11  DAYS_EMPLOYED         2500 non-null   int64
 12  FLAG_MOBIL            2500 non-null   int64
 13  FLAG_WORK_PHONE       2500 non-null   int64
 14  FLAG_PHONE            2500 non-null   int64
 15  FLAG_EMAIL            2500 non-null   int64
 16  OCCUPATION_TYPE       1752 non-null   object
 17  CNT_FAM_MEMBERS       2500 non-null   int64
 18  CAR_SCORE             2500 non-null   int32
 19  REALTY_SCORE          2500 non-null   int32
 20  INCOME_TYPE_SCORE     2500 non-null   int32
 21  FAMILY_STATUS_SCORE   2500 non-null   int32
 22  HOUSING_SCORE         2500 non-null   int32
 23  INCOME_SCORE          2500 non-null   int32
 24  APPLICATION_SCORE     2500 non-null   int64
dtypes: int32(6), int64(11), object(8)
memory usage: 429.8+ KB
```

Through the figure above we observe that OCCUPATION_TYPE contains a significant amount of NaN values. There are 748 NaNs in occupation type, meaning that for 748 of our applicants the occupation type is not documented. This is unfortunate, because that means that we cannot use this column in our credit score calculation. Fortunately, the NAME_INCOME_TYPE column which represents the income category contains 2500 non-null values and can be used instead.

However, the income type alone would not tell us the full picture of the current financial status of the applicants without knowing what income brackets they actually fall under. By applying the .describe() method to application['AMT_INCOME_TOTAL'] we can further group applicants by their level of income.

```
In [118]: #obtaining key statistical information
          application['AMT_INCOME_TOTAL'].describe()

Out[118]: count    2.500000e+03
          mean     1.955443e+05
          std      1.142462e+05
          min      3.330000e+04
          25%      1.260000e+05
          50%      1.800000e+05
          75%      2.250000e+05
          max      1.350000e+06
          Name: AMT_INCOME_TOTAL, dtype: float64
```

The minimum total income in our sample size is 33,300. We can see that the 25th percentile is immensely higher at 126,000, also, the 50th percentile and mean are close to each other at 180,000 and 195,544.3 respectively. This allows us to understand that the majority of applicants in our dataset are quite wealthy, which could also have an effect on their spending behaviors and the total scores.

Additional checks were performed to confirm the integrity of our data. We verified that all of the IDs are unique and that in the credit-record record table we indeed have 6 months' history per ID.

```
In [9]: #make sure we don't any outlying data contradicting to the data dictionary. For example,
        #we confirm that we have 2500 unique IDs and no more than 2 genders.
        application.nunique()

Out[9]: ID                       2500
        CODE_GENDER                 2
        FLAG_OWN_CAR                2
        FLAG_OWN_REALTY             2
        CNT_CHILDREN                4
        AMT_INCOME_TOTAL           96
        NAME_INCOME_TYPE            5
        NAME_EDUCATION_TYPE         5
        NAME_FAMILY_STATUS          5
        NAME_HOUSING_TYPE           6
        DAYS_BIRTH                849
        DAYS_EMPLOYED             503
        FLAG_MOBIL                  1
        FLAG_WORK_PHONE             2
        FLAG_PHONE                  2
        FLAG_EMAIL                  2
        OCCUPATION_TYPE            17
        CNT_FAM_MEMBERS             5
        dtype: int64
```

```
In [115]: #verifying that all months have an equal count corresponding with the number of unique IDs
          credit['MONTHS_BALANCE'].value_counts()

Out[115]:  0     2500
          -1     2500
          -2     2500
          -3     2500
          -4     2500
          -5     2500
          Name: MONTHS_BALANCE, dtype: int64
```

Methodology

Type of Analysis and Rationale

When we began with the two datasets, our plan included the need to analyze both sets of data, eventually merging properties of each dataset we deemed significant.

When we looked into past studies, we found that many made use of Logistic Regression algorithm in order to “Evaluate profiles and scores of bad customers to propose features that need improvement to give them the opportunity to approach loans with low interest…Analyze and add features related to different types of income, especially in the form of cash transfers to the dataset” (Cao, N. T., Tran, L. H., & Ton-That, A. H. 2021, December).

In [1]:
```
import pandas as pd
import numpy as np
#import the Applicant's biographical information
application = pd.read_csv('./data/application-record-final.csv')
application
```

Out[1]:

	ID	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INC
0	5008804	M	Y	Y	0	
1	5008805	M	Y	Y	0	
2	5008806	M	Y	Y	0	
3	5008810	F	N	Y	0	
4	5008811	F	N	Y	0	
...	...	...	...	...	...	
2495	5024320	F	N	Y	0	
2496	5024321	F	N	Y	0	
2497	5024324	F	N	Y	0	
2498	5024326	F	N	Y	0	
2499	5024328	F	N	Y	0	

2500 rows × 18 columns

We used the pandas library on Jupyter Notebook by running import pandas as pd. Next, we were able to import our Application csv as seen above. We also used the numpy library by running import numpy as np in order to complete machine learning mathematical operations.

In order to better understand our Application Record data we created an application_record_dictionary and imported it to Jupyter Notebook. (as seen in the application-record-dictionary.csv)

We used .isnull().sum() in order to identity missing values within our data

```
application.isnull().sum().sort_values(ascending=False)
```

```
OCCUPATION_TYPE        748
ID                       0
CODE_GENDER              0
FLAG_EMAIL               0
FLAG_PHONE               0
FLAG_WORK_PHONE          0
FLAG_MOBIL               0
DAYS_EMPLOYED            0
DAYS_BIRTH               0
NAME_HOUSING_TYPE        0
NAME_FAMILY_STATUS       0
NAME_EDUCATION_TYPE      0
NAME_INCOME_TYPE         0
AMT_INCOME_TOTAL         0
CNT_CHILDREN             0
FLAG_OWN_REALTY          0
FLAG_OWN_CAR             0
CNT_FAM_MEMBERS          0
dtype: int64
```

In order to calculate a score for each set of data based on pertinent information, we have implemented a new approach which we named “Algorithmic Data Computation”. In the Application dataset, we assigned a scoring method as seen in our evaluation_criteria_dictionary to calculate an application score. (as seen in the evaluation-criteria-dictionary.csv)

In this method we singled out income type, family status, housing type, car ownership, realty ownership, and income total for our categories. We decided to select particular categories based on past research which indicated that the "Extraction and identification of the right variables for determining the customer value and his credit score from the operation system database" (Ghassempouri, & Hoseini (2017) was essential.

We began to calculate the application record score based on the biographic data, first assigning 6 extra points to every applicant who owns a car.

```
conditions2=[
    (application['FLAG_OWN_CAR']=='Y'),
    (application['FLAG_OWN_CAR']=='N')

]
```

```
values2=['6','0']
```

```
application['CAR_SCORE']=np.select(conditions2,values2)
```

```
application[['ID','FLAG_OWN_CAR','CAR_SCORE',]]
```

	ID	FLAG_OWN_CAR	CAR_SCORE
0	5008804	Y	6
1	5008805	Y	6
2	5008806	Y	6
3	5008810	N	0
4	5008811	N	0
...	...	...	...
2495	5024320	N	0
2496	5024321	N	0
2497	5024324	N	0
2498	5024326	N	0
2499	5024328	N	0

2500 rows × 3 columns

Next, we follow the same approach to assign 6 points towards realty ownership.

Next, we assigned partial points based on Income Type

```
conditions4=[
    (application['NAME_INCOME_TYPE']=='Pensioner'),
    (application['NAME_INCOME_TYPE']=='Commercial associate'),
    (application['NAME_INCOME_TYPE']=='Working'),
    (application['NAME_INCOME_TYPE']=='State servant'),
    (application['NAME_INCOME_TYPE']=='Student'),

]
```

```
values4=['8','6','5','7','3']
```

```
application['INCOME_TYPE_SCORE']=np.select(conditions4,values4)
```

```
application[['ID','NAME_INCOME_TYPE','INCOME_TYPE_SCORE',]]
```

	ID	NAME_INCOME_TYPE	INCOME_TYPE_SCORE
0	5008804	Working	5
1	5008805	Working	5
2	5008806	Working	5
3	5008810	Commercial associate	6
4	5008811	Commercial associate	6
...	...	...	...
2495	5024320	Pensioner	8
2496	5024321	Pensioner	8
2497	5024324	Pensioner	8
2498	5024326	Pensioner	8
2499	5024328	Commercial associate	6

2500 rows × 3 columns

Next, we assigned partial points based on Family Status.

```
conditions5=[
    (application['NAME_FAMILY_STATUS']=='Married'),
    (application['NAME_FAMILY_STATUS']=='Single / not married'),
    (application['NAME_FAMILY_STATUS']=='Civil marriage'),
    (application['NAME_FAMILY_STATUS']=='Separated'),
    (application['NAME_FAMILY_STATUS']=='Widow'),

]
```

```
values5=['6','2','4','3','4']
```

```
application['FAMILY_STATUS_SCORE']=np.select(conditions5,values5)
```

```
application[['ID','NAME_FAMILY_STATUS','FAMILY_STATUS_SCORE',]]
```

	ID	NAME_FAMILY_STATUS	FAMILY_STATUS_SCORE
0	5008804	Civil marriage	4
1	5008805	Civil marriage	4
2	5008806	Married	6
3	5008810	Single / not married	2
4	5008811	Single / not married	2
...	...	...	...
2495	5024320	Married	6
2496	5024321	Married	6
2497	5024324	Married	6
2498	5024326	Married	6
2499	5024328	Married	6

2500 rows × 3 columns

Next, we assigned partial points based on Housing Type

```
conditions6=[
    (application['NAME_HOUSING_TYPE']=='House / apartment'),
    (application['NAME_HOUSING_TYPE']=='With parents'),
    (application['NAME_HOUSING_TYPE']=='Municipal apartment'),
    (application['NAME_HOUSING_TYPE']=='Rented apartment'),
    (application['NAME_HOUSING_TYPE']=='Co-op apartment'),
    (application['NAME_HOUSING_TYPE']=='Office apartment')
]
```

```
values6=['4','3','1','2','3','2']
```

```
application['HOUSING_SCORE']=np.select(conditions6,values6)
```

```
application[['ID','NAME_HOUSING_TYPE','HOUSING_SCORE',]]
```

	ID	NAME_HOUSING_TYPE	HOUSING_SCORE
0	5008804	Rented apartment	2
1	5008805	Rented apartment	2
2	5008806	House / apartment	4
3	5008810	House / apartment	4
4	5008811	House / apartment	4
...	...	...	...
2495	5024320	House / apartment	4
2496	5024321	House / apartment	4
2497	5024324	House / apartment	4
2498	5024326	House / apartment	4
2499	5024328	House / apartment	4

2500 rows × 3 columns

Next, we assigned partial points based on the amount of total income.

```
conditions7=[
    (application['AMT_INCOME_TOTAL']<126000),
    (application['AMT_INCOME_TOTAL']>=126000) & (application['AMT_INCOME_TOTAL
    (application['AMT_INCOME_TOTAL']>=180000) & (application['AMT_INCOME_TOTAL
    (application['AMT_INCOME_TOTAL']>=225000)
]
```

```
values7=['4','6','8','10']
```

```
application['INCOME_SCORE']=np.select(conditions7,values7)
```

```
application[['ID','AMT_INCOME_TOTAL','INCOME_SCORE',]]
```

	ID	AMT_INCOME_TOTAL	INCOME_SCORE
0	5008804	427500	10
1	5008805	427500	10
2	5008806	112500	4
3	5008810	270000	10
4	5008811	270000	10
...	...	...	...
2495	5024320	121500	4
2496	5024321	121500	4
2497	5024324	121500	4
2498	5024326	121500	4
2499	5024328	225000	10

2500 rows × 3 columns

Next, we converted all of our data-types for our chosen categories to integer values in order to be able to calculate .sum() within python for the total score of their combined points.

```
application['CAR_SCORE']=application['CAR_SCORE'].astype(int)
application['REALTY_SCORE']=application['REALTY_SCORE'].astype(int)
application['INCOME_TYPE_SCORE']=application['INCOME_TYPE_SCORE'].astype(int)
application['FAMILY_STATUS_SCORE']=application['FAMILY_STATUS_SCORE'].astype(int
application['HOUSING_SCORE']=application['HOUSING_SCORE'].astype(int)
application['INCOME_SCORE']=application['INCOME_SCORE'].astype(int)
```

```
#Begin totaling up the partial scores
```

```
scores_list=['CAR_SCORE',
                   'REALTY_SCORE',
                   'INCOME_TYPE_SCORE',
                    'FAMILY_STATUS_SCORE',
                    'HOUSING_SCORE',
                    'INCOME_SCORE']
```

```
application['APPLICATION_SCORE']=application[scores_list].sum(axis=1)
```

```
application[['ID','CAR_SCORE',
                   'REALTY_SCORE',
                   'INCOME_TYPE_SCORE',
                    'FAMILY_STATUS_SCORE',
                    'HOUSING_SCORE',
                    'INCOME_SCORE', 'APPLICATION_SCORE']]
```

The application score as well as individual category scores are now visible in the application data-frame.

	ID	CAR_SCORE	REALTY_SCORE	INCOME_TYPE_SCORE	FAMILY_STATUS_SCORE	HC
0	5008804	6	6	5	4	
1	5008805	6	6	5	4	
2	5008806	6	6	5	6	
3	5008810	0	6	6	2	
4	5008811	0	6	6	2	
...	...	...	...	...	...	
2495	5024320	0	6	8	6	
2496	5024321	0	6	8	6	
2497	5024324	0	6	8	6	
2498	5024326	0	6	8	6	
2499	5024328	0	6	6	6	

2500 rows × 8 columns

In the Credit dataset, we assigned a scoring method as seen in our credit_record_dictionary to calculate a credit score. (as seen in credit-record-dictionary.csv)

We began by importing our Credit data csv.

```
#Importing the 6 month history of credit payments by each customer
import pandas as pd
import numpy as np
credit = pd.read_csv('./data/credit-record-final.csv')
credit
```

	ID	MONTHS_BALANCE	STATUS
0	5008804	0	C
1	5008804	-1	C
2	5008804	-2	C
3	5008804	-3	C
4	5008804	-4	C
...	...	...	...
14995	5024328	-1	C
14996	5024328	-2	C
14997	5024328	-3	C
14998	5024328	-4	0
14999	5024328	-5	0

15000 rows × 3 columns

First, we assigned the credit score points based on the status of payment for each month.

```
conditions=[
    (credit['STATUS']=='C'),
    (credit['STATUS']=='0'),
    (credit['STATUS']=='X'),
    (credit['STATUS']=='1'),
    (credit['STATUS']=='2'),
    (credit['STATUS']=='3'),
    (credit['STATUS']=='4'),
    (credit['STATUS']=='5')
]
```

```
values=['60','60', '50','50','40','30','20','0']
```

```
credit['CREDIT_SCORE'] = np.select(conditions,values)
```

```
credit
```

	ID	MONTHS_BALANCE	STATUS	CREDIT_SCORE
0	5008804	0	C	60
1	5008804	-1	C	60
2	5008804	-2	C	60
3	5008804	-3	C	60
4	5008804	-4	C	60
...	...	...	...	...
14995	5024328	-1	C	60
14996	5024328	-2	C	60
14997	5024328	-3	C	60
14998	5024328	-4	0	60
14999	5024328	-5	0	60

15000 rows × 4 columns

Next, we verified the data-type for our new Credit Score Value, and changed it to Integer in order to successfully compute our calculations.

```
credit['CREDIT_SCORE'].dtype
```

```
dtype('O')
```

```
credit['CREDIT_SCORE']=credit['CREDIT_SCORE'].astype(int)
credit['CREDIT_SCORE'].dtype
```

```
dtype('int32')
```

```
credit['ID']=credit['ID'].astype(int)
credit['ID'].dtype
```

```
dtype('int32')
```

```
credit2=credit[['ID', 'CREDIT_SCORE']].copy()
credit2
```

	ID	CREDIT_SCORE
0	5008804	60
1	5008804	60
2	5008804	60
3	5008804	60
4	5008804	60
...	...	...
14995	5024328	60
14996	5024328	60
14997	5024328	60
14998	5024328	60
14999	5024328	60

15000 rows × 2 columns

Next, we calculated the average credit score based on the points assigned to each of the 6 months.

```
credit2['ID']=credit2['ID'].astype(int)
credit2['CREDIT_SCORE']=credit2['CREDIT_SCORE'].astype(int)
```

```
#Calculate the average credit score based on the points assigned to each of the 
credit3=credit2.groupby('ID')['CREDIT_SCORE'].mean().round(2)
credit3
```

```
ID
5008804    60.00
5008805    60.00
5008806    60.00
5008810    60.00
5008811    60.00
           ...
5024320    60.00
5024321    60.00
5024324    58.33
5024326    60.00
5024328    60.00
Name: CREDIT_SCORE, Length: 2500, dtype: float64
```

```
credit4=credit3.to_frame()
credit4
```

	CREDIT_SCORE
ID	
5008804	60.00
5008805	60.00
5008806	60.00
5008810	60.00
5008811	60.00
...	...
5024320	60.00
5024321	60.00
5024324	58.33
5024326	60.00
5024328	60.00

2500 rows × 1 columns

```
#make sure we converted the series to the dataframe
credit4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2500 entries, 5008804 to 5024328
Data columns (total 1 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   CREDIT_SCORE  2500 non-null   float64
dtypes: float64(1)
memory usage: 39.1 KB
```

Once both of these scores were calculated the data was merged into a dataset for credit score final, with a final score which was calculated by summing the scores found in application score and credit score.

```
#Now that we have calculated the credit points based on the 6 months credit payment history
#AND the credit points for the biographic data, we can merge the 2 dataframes together and calculate the overall score.
```

```
credit_score_final=pd.merge(application,credit4,on='ID')
credit_score_final
```

	ID	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	NAME_INCOME_TYPE	NAME_EDUCATION_TYPE
0	5008804	M	Y	Y	0	427500	Working	Higher education
1	5008805	M	Y	Y	0	427500	Working	Higher education
2	5008806	M	Y	Y	0	112500	Working	Secondary / secondary special
3	5008810	F	N	Y	0	270000	Commercial associate	Secondary / secondary special
4	5008811	F	N	Y	0	270000	Commercial associate	Secondary / secondary special
...	...	...	...	...	...	...	...	...
2495	5024320	F	N	Y	0	121500	Pensioner	Secondary / secondary special
2496	5024321	F	N	Y	0	121500	Pensioner	Secondary / secondary special
2497	5024324	F	N	Y	0	121500	Pensioner	Secondary / secondary special
2498	5024326	F	N	Y	0	121500	Pensioner	Secondary / secondary special
2499	5024328	F	N	Y	0	225000	Commercial associate	Secondary / secondary special

2500 rows × 26 columns

```
sum_column = credit_score_final["APPLICATION_SCORE"] + credit_score_final["CREDI
credit_score_final['FINAL_SCORE'] = sum_column
credit_score_final
```

	ID	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INC
0	5008804	M	Y	Y	0	
1	5008805	M	Y	Y	0	
2	5008806	M	Y	Y	0	
3	5008810	F	N	Y	0	
4	5008811	F	N	Y	0	
...	...	...	...	...	...	
2495	5024320	F	N	Y	0	
2496	5024321	F	N	Y	0	
2497	5024324	F	N	Y	0	
2498	5024326	F	N	Y	0	
2499	5024328	F	N	Y	0	

2500 rows × 27 columns

```
#Understanding our finals scores
credit_score_final['FINAL_SCORE'].describe()
```

```
count    2500.000000
mean       86.881756
std         6.773853
min        27.000000
25%        83.000000
50%        87.000000
75%        91.000000
max       100.000000
Name: FINAL_SCORE, dtype: float64
```

evaluation-criteria-dictionary.csv

Column Title	Points
-	**-**
NAME_INCOME_TYPE	**Points Assigned**
Pensioner	**8**
Commercial associate	**6**
Working	**5**
State servant	**7**
Student	**3**
-	**-**
NAME_FAMILY_STATUS	**Points Assigned**
Married	**6**
Single / not married	**2**
Civil marriage	**4**
Separated	**3**
Widow	**4**
-	**-**
NAME_HOUSING_	**Points**

TYPE	Assigned
House / apartment	4
With parents	3
Municipal apartment	1
Rented apartment	2
Co-op apartment	3
Office apartment	2
-	-
FLAG_OWN_CAR	Points Assigned
N	0
Y	6
-	-
FLAG_OWN_REALTY	Points Assigned
N	0
Y	6
-	-
AMT_INCOME_TOTAL	Points Assigned

0-24 %	**4**
25-49 %	**6**
50-74 %	**8**
75-100 %	**10**
-	**-**
STATUS	**Points Assigned**
0 or C	**60**
1 or X	**50**
2	**40**
3	**30**
4	**20**
5	**0**

ANALYSIS AND RESULTS

Conclusions, Recommendations, and Limitations

Conclusions

Through our analysis of the two datasets, we found there to be little significant influence of biographical information found in the Application data, on our credit score. Our initial analysis of the raw data led us to pursue a link between our carefully selected categories of income type, family status, housing type, car ownership, realty ownership, and income total. Previous research supported our rationale and methodology for implementing a point system, one study stated “This research investigates both financial and non-financial factors that influence credit risk evaluation. To classify different importance of all factors, we assign a weight to each factor” (Li, L., Jin, M., & Liang, X. 2011) as the structure for their work. However, when we computed our data, we found uniform correlations to credit score as well as negligible variation between different variables held within each category. It has become clear that within the scope of our particular population for our data that the credit payment status is superiorly significant in calculating credit score and had the greatest impact when it came to correlation and considerable variation.

This can be best observed visually by comparing the figures 5 and 6. Figure 5 shows the relationship between the combined final score (this includes the biographic score within it) and the occupation. Figure 6 shows the same relationship but uses only the credit score calculated based on the monthly payment statuses. We can see the bar plot in figure 6 is a lot more uniform with minor fluctuations.

Recommendations

When calculating credit scores of individuals who are new to the system and have no credit history, biographical information is not the best metric to use in credit score calculation. For that reason, the best metric is to analyze their behavior towards making payments over a suggested time-frame of at least six months in order to determine what value of score the applicant should be granted.

Limitations

The primary limitation we have encountered throughout the duration of our research were imperfections of the datasets we have started with. Due to the amount of mismatched IDs it was initially not plausible to estimate the final amount of entries we would end up with after cleaning the data. Additionally, the more we studied our final datasets through the means of Python, we learnt that some of the column's values were not specific enough to be used as we initially intended. For example, we realized that in the NAME_HOUSING_TYPE column the overwhelming majority of rows contain the "House / apartment" value (2255/2500). This was counterintuitive to us that those two values are concatenated like this and not split up into separate values, especially because there are other apartment types listed in other rows.

```
In [19]: application['NAME_HOUSING_TYPE'].value_counts()

Out[19]: House / apartment      2255
         With parents             99
         Municipal apartment      72
         Rented apartment         55
         Co-op apartment          10
         Office apartment          9
         Name: NAME_HOUSING_TYPE, dtype: int64
```

Another limitation with the dataset was the lack of background information of when, where and under which circumstances the data was collected. The small size of the final dataset and not enough diversity could also have had an effect on the final results.

Future Research

Now that Flex Data has established the robust Algorithmic Data Computation method for credit card scoring based on the credit payment history and biographical information, we intend to conduct a future study with a larger dataset which would preferably be collected in the state of Oklahoma with a wider range of respondents. We would also want the value counts within each category to be more proportionate to each other in order to eliminate sample bias and outliers.

In [122]:
```
plt.figure(figsize=(30,23))
sns.set(font_scale=1)
sns.barplot(x='OCCUPATION_TYPE', y='FINAL_SCORE', data=credit_score_final)
plt.grid()
```

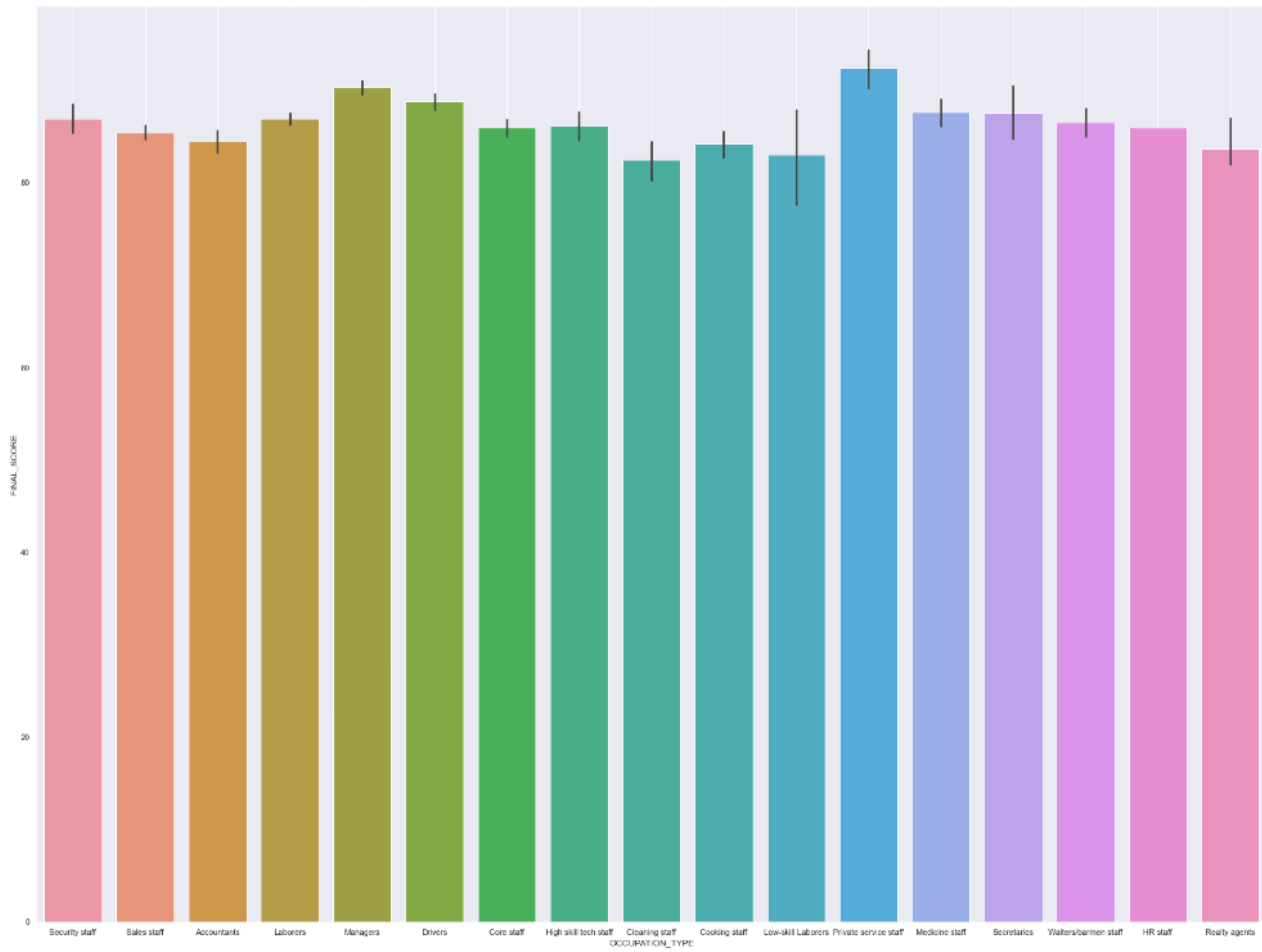


Figure 5: FINAL_SCORE (CREDIT_SCORE + APPLICANT_TYPE) ratio among different occupation types

```
In [123]: plt.figure(figsize=(30,23))
          sns.set(font_scale=1)
          sns.barplot(x='OCCUPATION_TYPE', y='CREDIT_SCORE', data=credit_score_final)
          plt.grid()
```

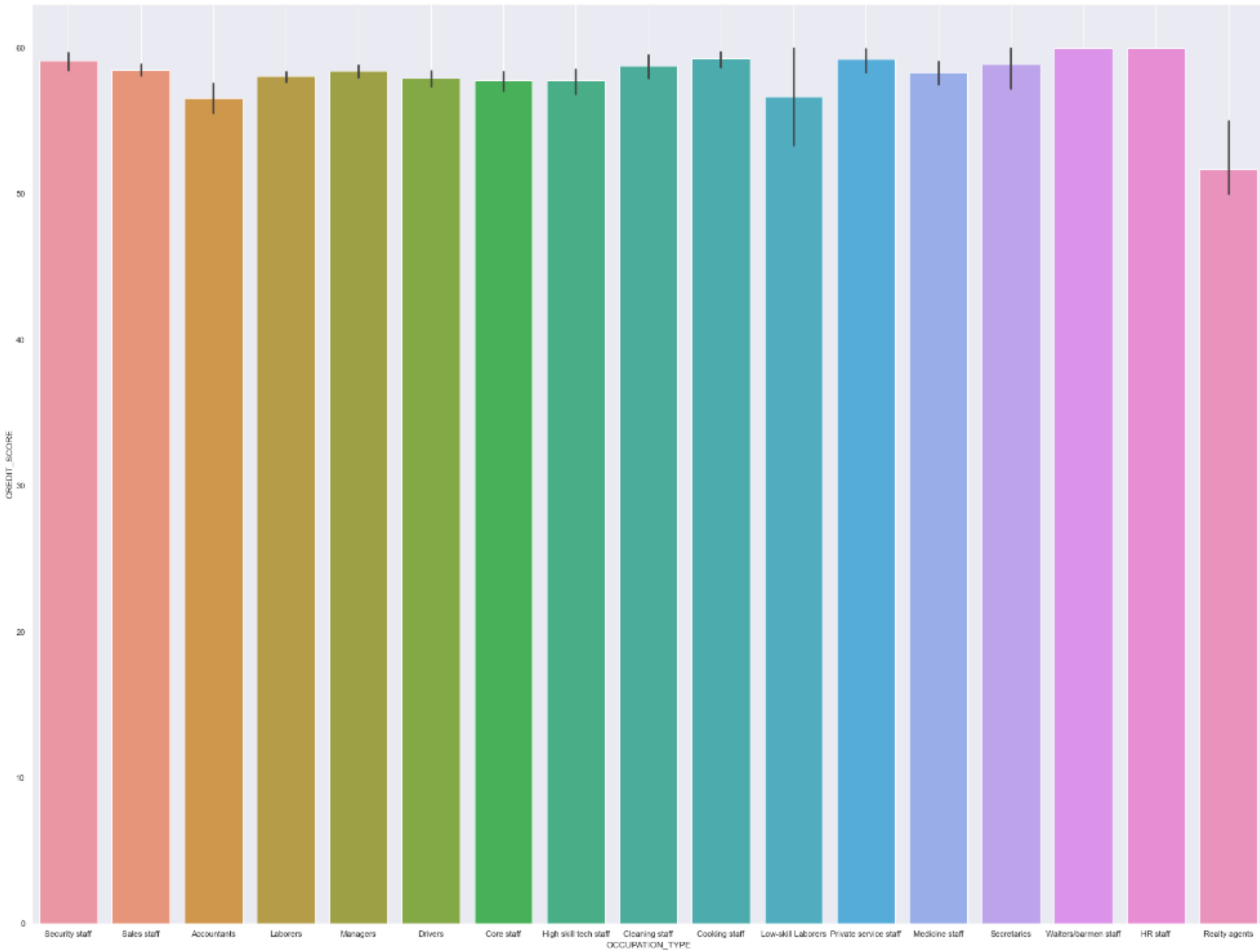


Figure 6: CREDIT_SCORE ratio among different occupation types

Appendix

Python Codes

```
import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

%matplotlib inline

df.describe()

df.sort_values

df.sort_index()

df.isnull().sum()

df.nunique()

df[].value_counts()

df.astype()

df.groupby()

df.to_frame()

np.select()

pd.merge()

plt.figure(figsize=(10,7))

sns.set(font_scale=1)

sns.barplot(x='NAME_HOUSING_TYPE', y='HOUSING_SCORE', data=credit_score_final)

plt.grid()
```

References

Analytics, J. (n.d.). *Just analytics: Engage.execute.exceed*. Just Analytics | Engage.Execute.Exceed. Retrieved April 19, 2022, from https://www.justanalytics.com/

Cao, N. T., Tran, L. H., & Ton-That, A. H. (2021, December). *Using machine learning to create a credit scoring model in banking and Finance*. IEEE Xplore. Retrieved April 5, 2022, from https://ieeexplore.ieee.org/abstract/document/9718414

Consumer Financial Protection Bureau. (2020, September 1). *What is a credit score?* ConsumerFinance.gov. Retrieved April 19, 2022, from https://www.consumerfinance.gov/ask-cfpb/what-is-a-credit-score-en-315/

Chatterjee, S., Corbae, D., Dempsey, K., & Ríos-Rull, J.-V. (2020, August). *A QUANTITATIVE THEORY OF THE CREDIT SCORE.* nber.org. Retrieved April 22, 2022, from https://www.nber.org/system/files/working_papers/w27671/w27671.pdf

Ghassempouri, & Hoseini. (2017). *A Methodology for Calculating Customer Credit Score Based on Customer Lifetime Value Model.* researchgate. Retrieved April 22, 2022, from https://www.researchgate.net/profile/Seyyed-Hoseini-4/publication/324551173_A_Methodology_for_Calculating_Customer_Credit_Score_Based_on_Customer_Lifetime_Value_Model/links/606f2183299bf1c911b9f925/A-Methodology-for-Calculating-Customer-Credit-Score-Based-on-Customer-Lifetime-Value-Model.pdf

Knutson,Melissa L. (2020). *Credit Scoring Approaches Guidelines*. The World Bank Group, from https://thedocs.worldbank.org/en/doc/935891585869698451-0130022020/CREDIT-SCORING-APPROACHES-GUIDELINES-FINAL-WEB

Li, L., Jin, M., & Liang, X. (2011). *Evaluating credit risk based on financial and non-financial factors. IIE Annual Conference.Proceedings*, , 1-8. Retrieved from https://libproxy.uco.edu/login?url=https://www.proquest.com/scholarly-journals/evaluating-credit-risk-based-on-financial-non/docview/1190345126/se-2?accountid=14516

Samsul, I., & Daud, A. (2012, April 13). *Business Process Improvement of Credit Card Department: Case study ... - researchgate*. ResearchGate. Retrieved April 20, 2022, from https://www.researchgate.net/publication/258076096_Business_process_improvement_of_credit_card_department_Case_study_of_a_multinational_bank

Seanny. (2020). Credit Card Approval Prediction. University of Maryland, from https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction

Tathah, M. (2018, October 25). *Which states have the most credit cards?* Experian. Retrieved April 19, 2022, from https://www.experian.com/blogs/ask-experian/which-states-have-the-most-credit-cards/