# ML Capstone Project-Social Development Bank Loans 2019

Social Development Bank Loans

## Content

## 1.Introduction:


Social Development Bank.

## About Social Development Bank:

“The Bank is considered to be one of the main government pillars for economic and social development funding to the citizens in Saudi Arabia. SDB focus’s in providing social financing products and business solutions to the low-income citizens, and creating awareness in financial planning & saving, as well as funding freelancers, micro, startup, and small businesses in way to enable them to contribute effectively to the economic growth of the country. The objectives have been verified as the following:

To provide free of interest loans for freelancers, micro, startup and small business to encourage them to run their own businesses independently.

To provide free of interest Social Loans for low-income citizens, in order to help them overcome their financial difficulties.

To encourage savings for individuals and institutions in the Kingdom, and to find the appropriate tools to achieve this goal.

The Bank has 24 branches in different regions of the Kingdom of Saudi Arabia to deliver services efficiently to its citizens."

Vision 2030.

**Vision 2030 and Problem Statement:**

In order to achieve the Bank's goals in achieving the goals and programs of Vision 2030 by enabling social development tools and enhancing the financial independence of individuals and families towards a vibrant and productive society, We found it necessary to anticipate the future value of financing loans granted to the individual, because through this it is possible to predict how the individual will become financially independent, enhance financial sufficiency and raise economic productivity.

### The goal:

What is the funding value for a particular customer? In our model we are predicting the funding value for a particular customer, based on the data provided in the Social Development Bank dataset.
After considering and exploring the customer information that has been provided to the Bank, we made a Random Forests model that predicts the appropriate funding value that the customer is willing to get.

## 2. Data Review:

Social Development Bank dataset is an open-source data provided by the Open Data portal of Saudi Arabia initiative. The data was obtained in the period of 2019 as described in the official website but we took our dataset from Kaggle as it was translated into English.

It contains 15 columns and 11,176 rows.

Saudi Arabia Social Development Bank Loans 2019
Saudi Arabia Social Development Bank Loans 2019 English version
www.kaggle.com

| | ID | bank branch | funding type | funding classification | customer sector | financing value | installment value | cashing date | sex | age | social status | special needs | number of family members | saving loan | income |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.0 | Tabūk | social | family | government employee | 60000.0 | >= 1000 | 2019/02 | MALE | >= 30 | married | No | >= 05 | No | < 5000 |
| 1 | 2.0 | Hail | project | solution | NaN | 160000.0 | >= 1000 | 2019/01 | MALE | < 30 | single | No | < 02 | No | < 5000 |
| 2 | 3.0 | Tabūk | social | marriage | government employee | 60000.0 | >= 1000 | 2019/02 | MALE | < 30 | married | No | >= 02 | No | >= 7500 |
| 3 | 4.0 | Medina | social | marriage | employee of a government company | 60000.0 | < 1000 | 2019/03 | MALE | < 30 | married | No | >= 10 | No | >= 5000 |
| 4 | 5.0 | Medina | social | family | private sector employee | 60000.0 | >= 1000 | 2019/02 | FEMALE | >= 30 | divorced | No | >= 02 | No | >= 10000 |

A view on the database.

### 3. Data Preprocessing:

**In this chapter**, we explored the database and found that it contains missing values. we have 4 columns with missing values.

```
ID                              0
bank branch                     0
funding type                    0
funding classification          0
customer sector              3950
financing value                 0
installment value               0
cashing date                    0
sex                             0
age                             6
social status                   0
special needs                   0
number of family members       43
saving loan                     0
income                        114
dtype: int64
```

Missing values.

### 1. Handle missing values.

To handle missing values each column's missing values were replaced with its most frequent value.

We filled in the missing values using the most frequent strategy only because the data type of these columns is Categorical.

```
# See what is the frequent value in each column (for missing values columns).
# customer sector coulmn
sector_freq = data['customer sector'].mode()[0]
print(sector_freq)

# income column
income_freq = data['income'].mode()[0]
print(income_freq)

# number of family members column
family_num_freq = data['number of family members'].mode()[0]
print(family_num_freq)

# age column
age_freq = data['age'].mode()[0]
print(age_freq)
# Fill in the Missing Values using the Simple Imputer with the Most Frequent strategy
imputer = SimpleImputer(strategy='most_frequent', missing_values=np.nan)
```

```
imputer = imputer.fit(data[['customer sector', 'income', 'number of family members', 'age']])
data[['customer sector', 'income','number of family members', 'age']] = imputer.transform(
    data[['customer sector', 'income', 'number of family members', 'age']])
```

## 2. Handling outliners.

The financing value column has notable outliers that needs to be handled.


Handling outliners.

**Feature Engineering:**

1. Delete unneeded columns ID, cashing date and social status.

2. Replace the cities in the bank branch columns with their main region. Instead of 27 different values, we ended up with 5 regions.

```
# Replace the cities in the bank branch columns with their main region.
# Instead of 27 different values, we ended up with 5 regions.
Central_Region = ['Riyadh', "Kharj", "Al Majma'ah", "Wadi ad-Dawasir", "Duwadimi"]
Eastern_Region = ["Dammam", "Hafar Al Batin"]
Southern_Region = ["Abha", "Khamis Mushait", "Al Bahah", "Jazan", "Najrān", "Bisha"]
Western_Region = ["Jeddah", "Yanbu", "Mecca", "Medina", "Taif", "Al Qunfudhah"]
Northern_Region = ["Tabūk", "Buraydah", "Hail", "Arar", "Al Jowf", "Ar Rass", "Al Namas", "Al Qurayyat"]

data['bank branch'] = data['bank branch'].replace(Central_Region,'Central Region')
data['bank branch'] = data['bank branch'].replace(Eastern_Region,'Eastern Region')
data['bank branch'] = data['bank branch'].replace(Southern_Region,'Southern Region')
data['bank branch'] = data['bank branch'].replace(Western_Region,'Western Region')
data['bank branch'] = data['bank branch'].replace(Northern_Region,'Northern Region')
data.head()
```

### 3. Replace certain columns values with 0 and 1.

- Replace (Yes/No) columns with 0/1 (special needs and saving loan features).
- Replace sex column with 0 -> male , 1 -> female.

### 4. Apply Label Encoding on certain columns.

```
# Apply Label Encoding to convert certain columns from a categorical type into a numerical one.
# Create a list of the columns to be converted into numerical values.
cols = ['installment value', 'age', 'number of family members', 'income']

# Encode labels of multiple columns at once
data[cols] = data[cols].apply(LabelEncoder().fit_transform)

# Print head
data.head()
```

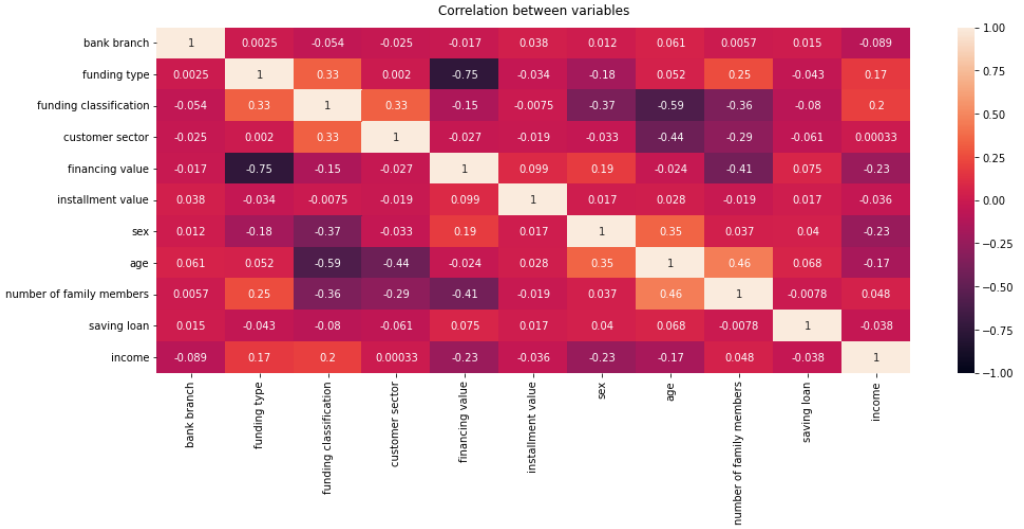
## The important columns' values before the encoding and after:

The unique values of the (bank branch) column are ['Northern Region', 'Western Region', 'Central Region', 'Southern Region', 'Eastern Region'], and after encoding become [2, 4, 0, 3, 1].

The unique values of the (Funding type) column are ['social', 'project', 'transfer'], and after encoding become [1, 0, 2].

The unique values of the (income) column are ['< 5000', '>= 7500', '>= 5000', '>= 10000'], and after encoding become [0, 3, 2, 1].

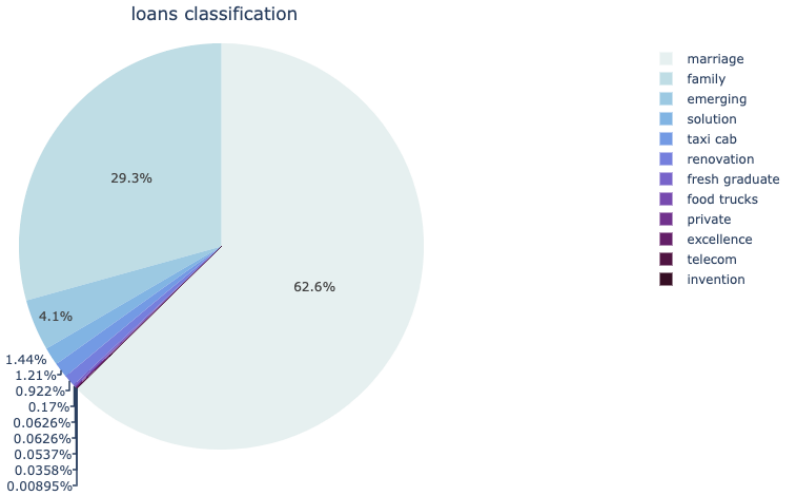
## A plot to shows the Correlation between features:




correlation between features.

# 4. Data Exploration:

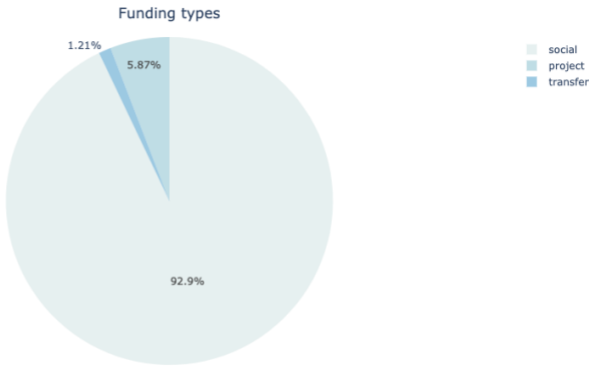
## 1. Funding classification.




Funding classification.

*The type of funding varies from funding a loan for marriage and family, or for small projects such as buying a taxicab, the highest percentage is the financing of a loan for marriage.*

## 2. Funding type.




Funding type.

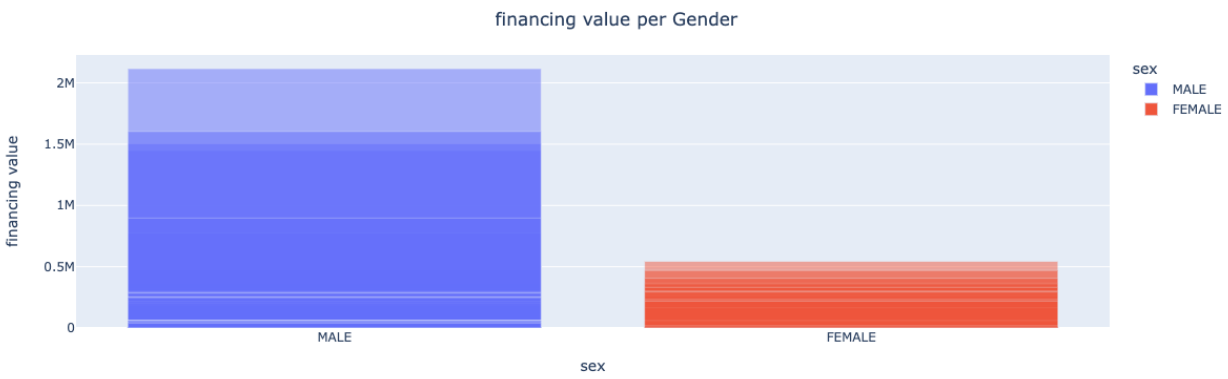
*Most clients take social loans.*

## 3.*Total of financing value for each financing purpose.*


Total of financing value for each financing purpose.

*Total of financing value for each financing purpose in three months, for example, in February, 20 million was given for marriage loans.*

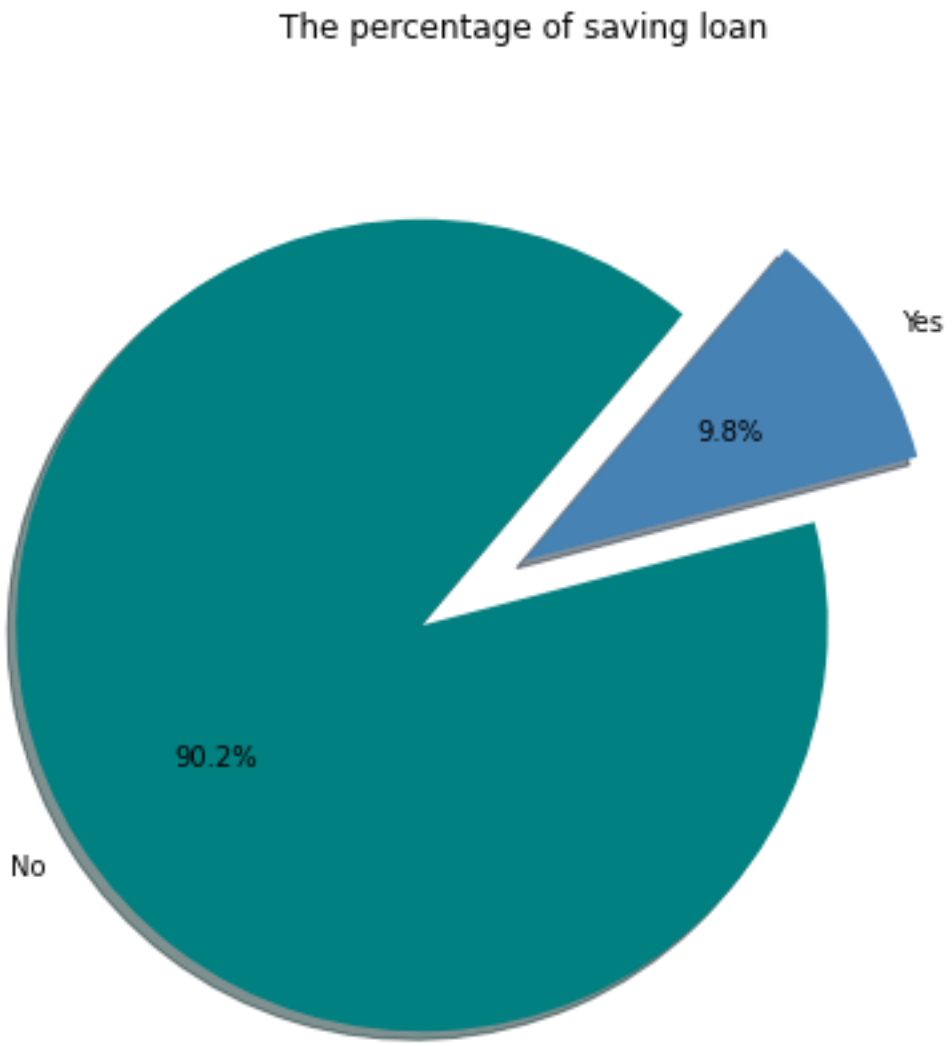
## 4. Financing value for each gender.




Financing value for each gender.

*This chart shows the total financing value of the loans per gender.*

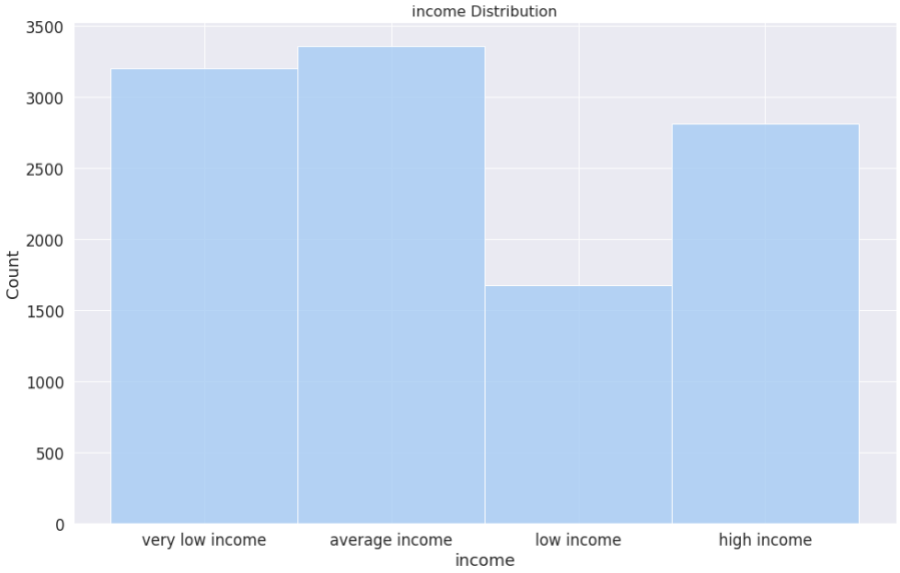
## 5. The percentage of saving loan.




The percentage of saving loan.

*The percentage of clients who have savings loans is 9.8%.*

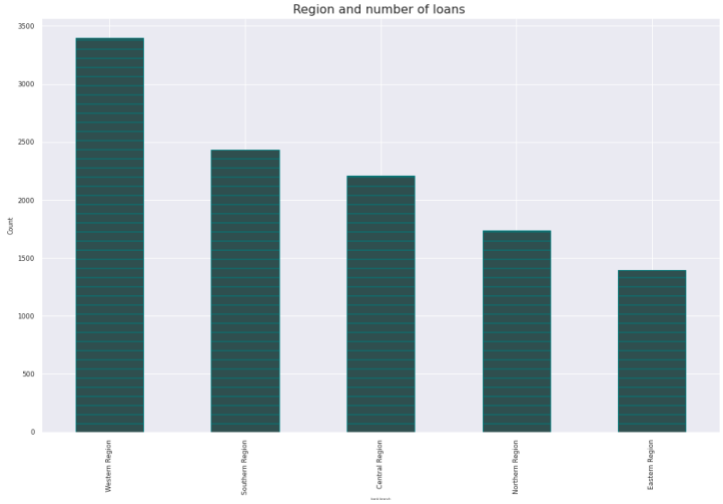
## 6. Income Distribution.




income Distribution.

*This graph shows that most clients who apply for a loan are average income and the least number of clients who apply for loans are low-income people.*

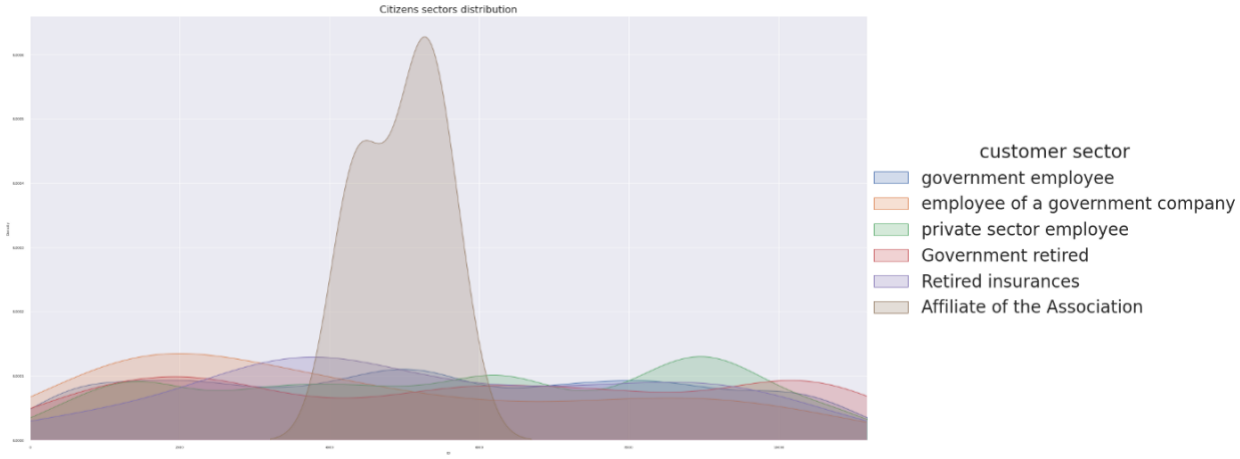
## 7. Bank branches.




Bank branches by region.

*Most of the bank's branches are located in the western region of the Kingdom of Saudi Arabia.*

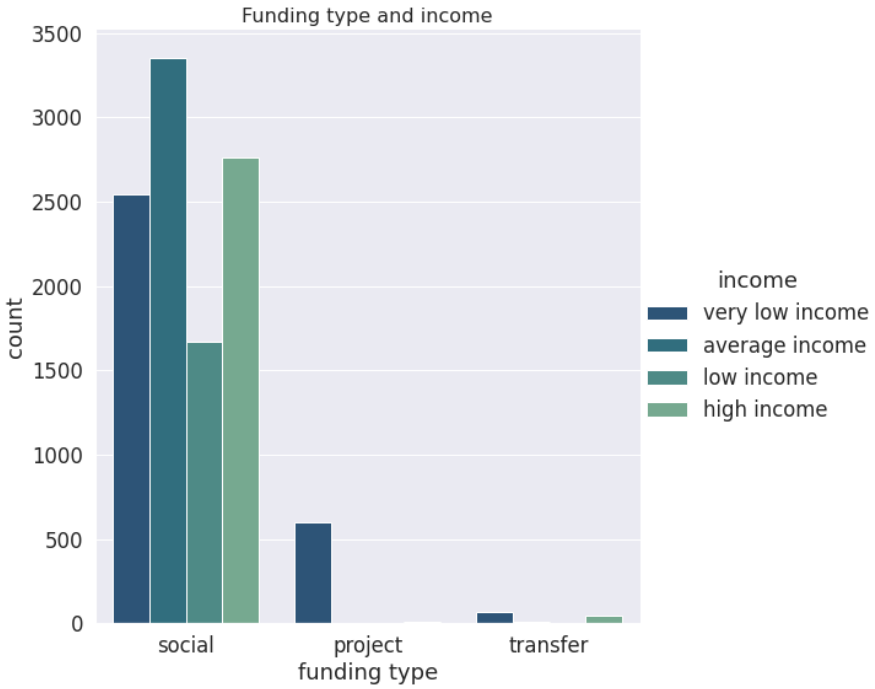
## 8. clients sector.




clients sector.

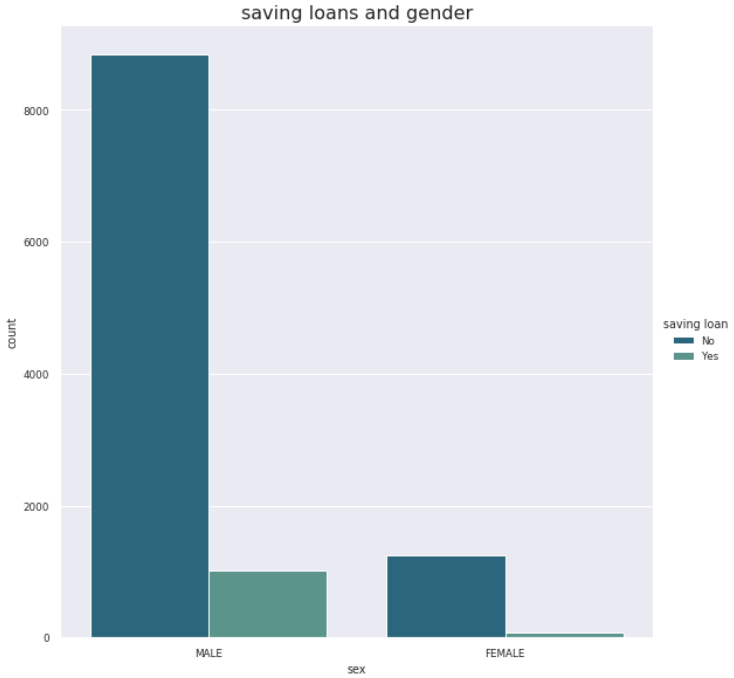
*clients sector varies from government work to private sector or affiliated of the association but the majority of the clients works in government companies.*

## 9. The number of clients with their income category and financing type.



*In this chart, we see that most of the loan requests are social loans, and they are mostly applied by average -income clients.*

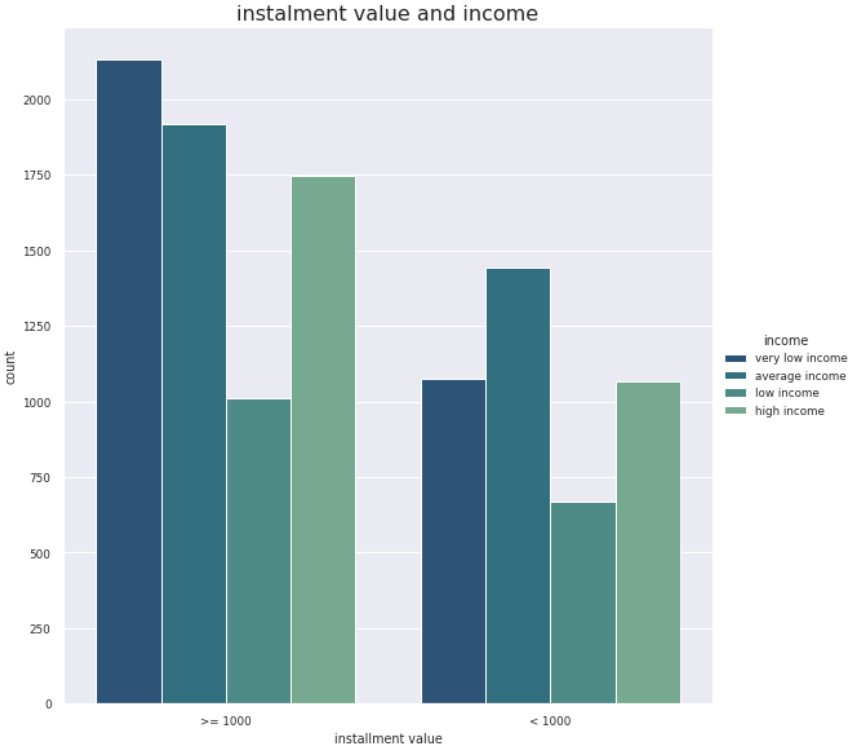
## 10. The number of clients who have savings loans and their gender.




The number of clients who have savings loans and their gender.

*We see that the number of males is much more than females, and most of them do not have saving loans.*

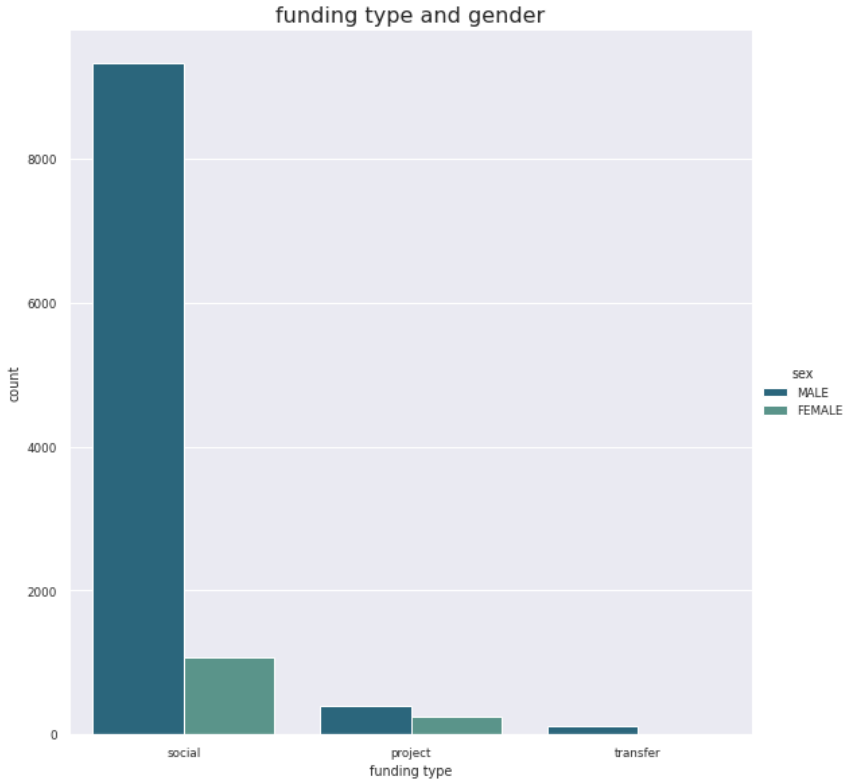
## 11. The value of the loan installment in relation to the income category of the clients.




The value of the loan installment in relation to the income category of the clients.

*The value of the loan installment can be higher than or equal to 1000 Riyals per month or less than 1000 riyals per month, and we see here that very low-income people often take a loan with a monthly installment value higher than 1000 Riyals.*

### 12. The number and gender of clients in relation to the type of funding.




The number and gender of clients in relation to the type of funding.

*We see here that most of the males get a loan for a social purpose.*

## *5. Building Regression Models:*

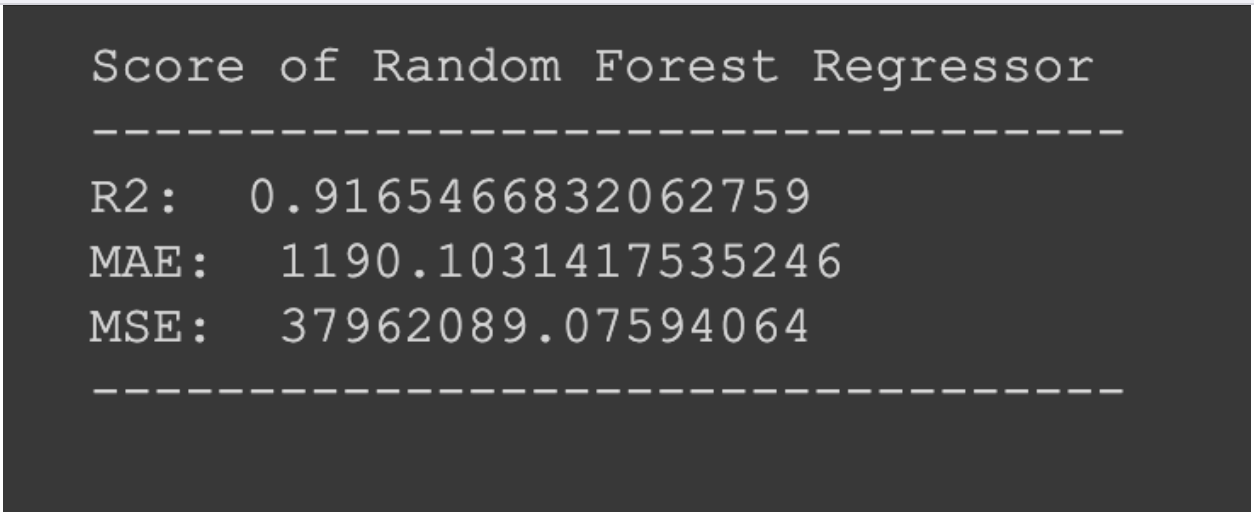
### Based on our target class selection, we build regression models.

We trained 4 models linear Regression — Random Forest Regression — Decision Tree Regression — Support Vector Regression, There was almost no difference between the random forest model and the decision tree model. Therefore, we applied the tuning method to them to conclude with only one best model.

Random Forest Regression

```
rf_reg = RandomForestRegressor(n_estimators=10, max_depth=6, random_state=42) # Initialize the model
rf_reg.fit(X_train,y_train) # Fit the model

preds_rfr = rf_reg.predict(X_test) # Predict X_test
```




## Grid Search for Random Forest Regression.

```
param_grid = {

    "n_estimators": [5,7,10, 15], # how many trees in our forest
    "max_depth": [2,4,6] # how deep each decision tree can be
}

grid = GridSearchCV(
```

```
    rf_reg,
    param_grid,
    cv = 5,
    n_jobs=-1,
    verbose=1,
    scoring="neg_mean_absolute_error"
)

grid.fit(X_train, y_train)
# Re-create the model using the best parameters
Rf = RandomForestRegressor(max_depth = 6, n_estimators = 5)
Rf.fit(X_train,y_train)
preds_rf = Rf.predict(X_test)

# Calculate the accuracy score for Decision Tree regression
r2_score(y_test,preds_rf)
0.9187159701066172
# Calculate the MSE for Random Forest Regression
mean_absolute_error(y_true=y_test, y_pred=preds_rf)
1186.6477430321252
```

As a result of tuning the models, we can conclude that the Random Forest Regression is our best model. A better MAE was achieved than with a Decision Tree model.

In the Random Forest, the MAE is 1186, while in the Decision Tree, it is 1197.

### ML Pipeline for Best Model.

```
pipe = make_pipeline(
    # Step-1 Scale parameters
    StandardScaler(),
    # Step-2 fit the principles to the ML model
    RandomForestRegressor(max_depth = 6, n_estimators = 5)
)

pipe.fit(X_train, y_train)
pipe.score(X_train, y_train)
0.9278863735651952
```

## *6. Results:*

In the first part of the project, we cleaned up the data by handling the missing values and outliers in it and preforming some Feature Engineering and encoding . We placed the target variable at the center and explored the data around it.

In the second part, we Explore the data further and made some initial conclusions based on Explore the features.

In the building models part, we build 4 algorithms linear Regression — Random Forest Regression — Decision Tree Regression — Support Vector Regression.
In this process, we used GridSearchCV to set parameters for two algorithms. the two models were then tested on a new train and test set and benchmarked to ensure they were not overfitted.

As for model accuracy and sensitivities, we'd say, they've worked very well for the most part. The numbers are given above.

## 7. Future Work:

For further improvements in the future, we aim to enhance our model by getting more data over the next years. Such improvements would help predict the future value of financing loans granted to the individual and predict how the individual will become financially independent, enhance financial sufficiency and raise economic productivity. Gathering and preparing more data will also be beneficial to improve the model's performance.

We can also extend the model's capabilities by deploying it to make analytical predictions or feeding it with new types of data. Moreover, creating a model that can classify requests as being approved or rejected by training the model on the complete set of data where some citizen requests were denied.