# Problem Statement:

A person's creditworthiness is often associated (conversely) with the likelihood they may default on loans. Data on about 1000 loan applications is provided, along with a certain set of attributes about the applicant itself, and whether they were considered high risk.

0 = Low credit risk i.e high chance of paying back the loan amount

1 = High credit risk i.e low chance of paying back the loan amount

# Data Description:

Data is provided using 2 files- 1) applicant.csv 2) loan.csv.

Applicant.csv has 15 features and loan.csv has 13 features having 1 common features as applicant_id.

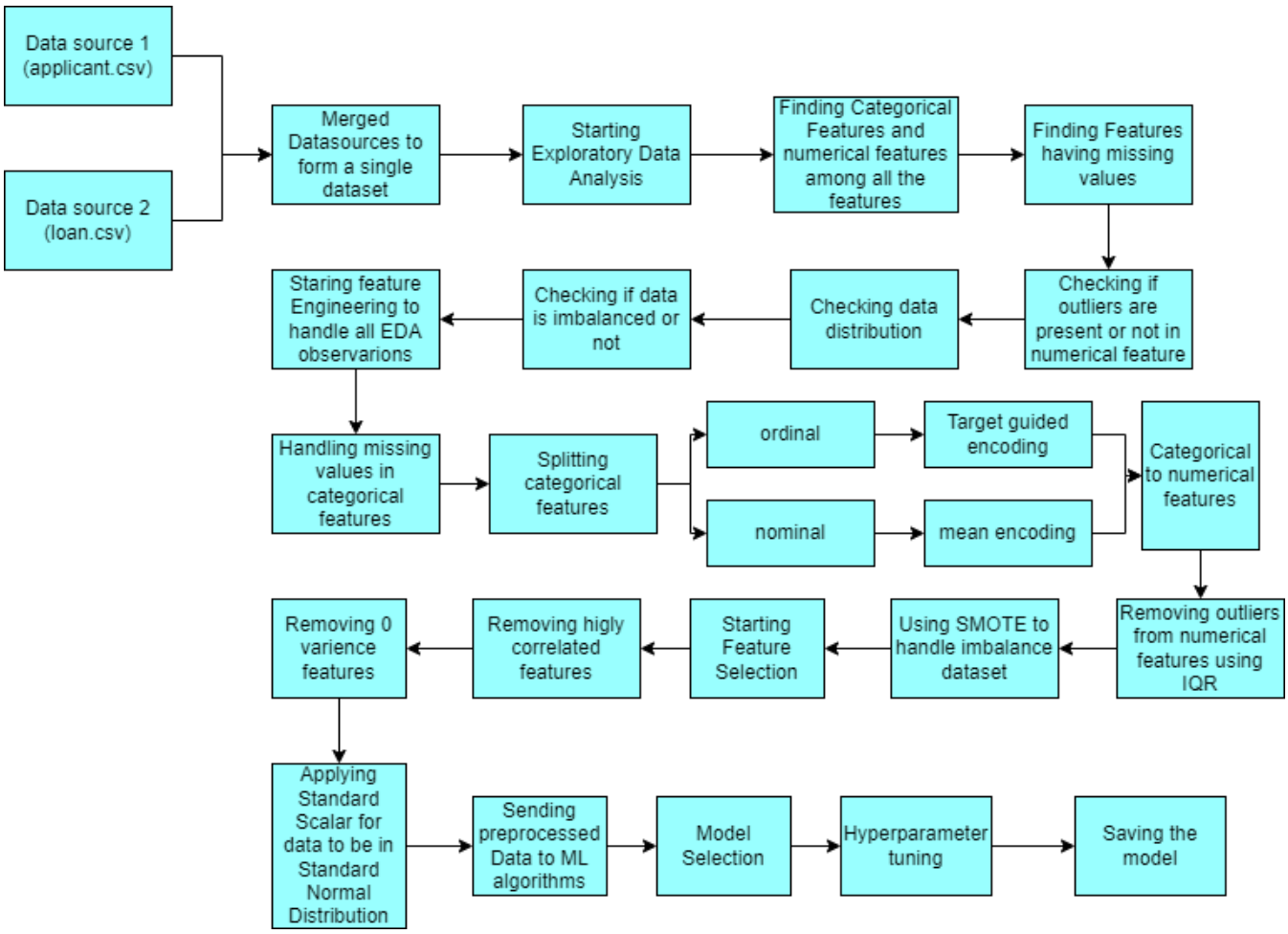
## Features in applicant.csv:

1. applicant_id
2. Primary_applicant_age_in_years
3. Gender
4. Marital_status
5. Number_of_dependents
6. Housing
7. Years_at_current_residence
8. Employment_status
9. Has_been_employed_for_at_least
10. Has_been_employed_for_at_most
11. Telephone
12. Foreign_worker
13. Savings_account_balance
14. Balance_in_existing_bank_account_
15. Balance_in_existing_bank_account_

## Features in Loan.csv:

1. loan_application_id
2. applicant_id
3. Months_loan_taken_for
4. Purpose
5. Principal_loan_amount
6. EMI_rate_in_percentage_of_disposable_income
7. Property
8. Has_coapplicant
9. Has_guarantor
10. Other_EMI_plans
11. Number_of_existing_loans_at_this_bank
12. Loan_history
13. high_risk_applicant

# Workflow for the ML Pipeline:



# Approach:

- As any Data Science project starts with data collection from various sources. Same was followed in this project also. 2 Data sources were provided – 1) applicant.csv and 2) loan.csv. In order to find some insights from this data, it was essential that all the data was stored into 1 common file. So based on the common feature (applicant_id) in both the dataset, data was merged into 1 common file.
- After the data was available in a single file. EDA process was initiated
- Performing EDA had several subtask within it, for e.g.-
    1. Finding total no of features in the data
    2. Dividing features in **categorical** and **numerical** features
    3. Dividing **numerical** features into **discrete** and **continuous** features
    4. Finding features having **missing values**
    5. Finding if **outliers** were present in **numerical** features
    6. Checking the **data distribution**
    7. Checking if data was **balanced** or **imbalanced**
    8. Performing various visualization to get more insights(bar plots, correlation matrix)
- Using the EDA analysis, feature Engineering was initiated to resolve all the scenarios before data modelling for e.g.-
    1. It was found that all the missing values belonged to the categorical features. Thus, they were handled by adding a new category **"Missing"** and replacing the missing values with this new category
    2. The next step was to convert categorical features into numerical features. But before performing that, categorical features where divided into ordinal features and nominal features.
    3. **Ordinal features** were converted into numerical features using **target guided encoding technique** and **nominal features** were converted into numerical features using **mean encoding technique**.
    4. Then, **outliers** from numerical features were handled using **IQR method**
    5. **SMOTE** technique was used to handle the **imbalanced dataset**

- After Feature Engineering, Feature Selection process was initiated to handle **curse of dimensionality** issue
    1. First Step in Feature Selection was to remove the **highly correlated features**. **Threshold of 80%** was set for the same. It means features which are correlated with each other for more than 80% will be dropped
    2. The next step was to drop features which show **0 variance** in all its records. The reason for dropping them were that if fed to the ML algorithm then no new pattern will be learned and it will be a waste of computation resources.
    3. After these steps, Data was sent through Standard Scalar to bring all features data into Standard Normal Distribution.
- After these data pre-processing steps, Data was ready to be fed into various ML algorithms. Various ML algorithms were selected like
    1. Logistic Regression
    2. Naive Bayes
    3. Decision Tree
    4. Random Forest
    5. Support vector classifier
    6. XGBoost
- Depending upon the best model, its hyper parameter was tuned and performance metrics were calculated and the model was saved.

# Question and Answers:

## Task 1:

1. **Do the Exploratory Data Analysis & share the insights.**

**Solution**: Data Analysis is performed in the Jupyter notebook as well as the html file is provided. Please follow the same to understand the various steps.

2. **How would you segment customers based on their risk (of default).**

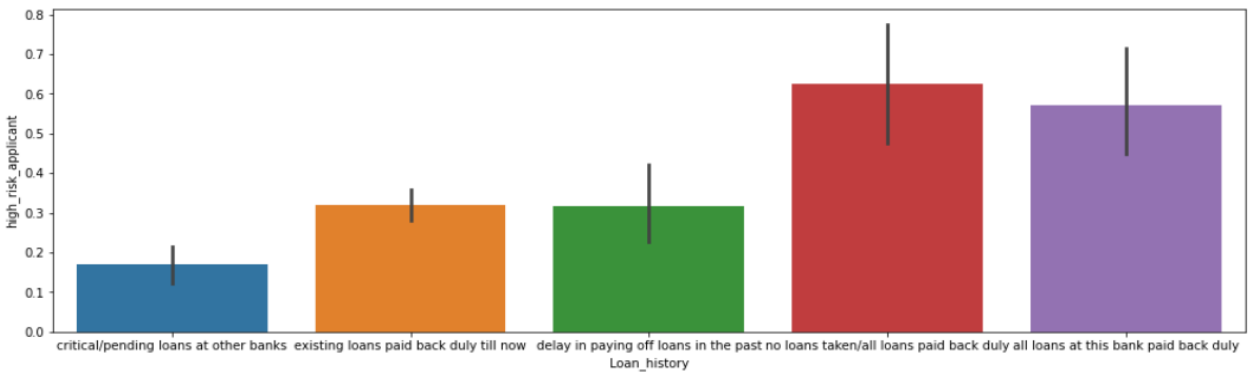
**Solution**: This is binary classification problem as there are 2 types of output possible here: Low risk and high risk. So we can use various Classification ML algorithms like:

- Logistic Regression
- Naive Bayes
- Descision Tree
- Random Forest
- Support vector classifier
- XGBoost

and depending upon the performance metrics select any one this model to solve the problem

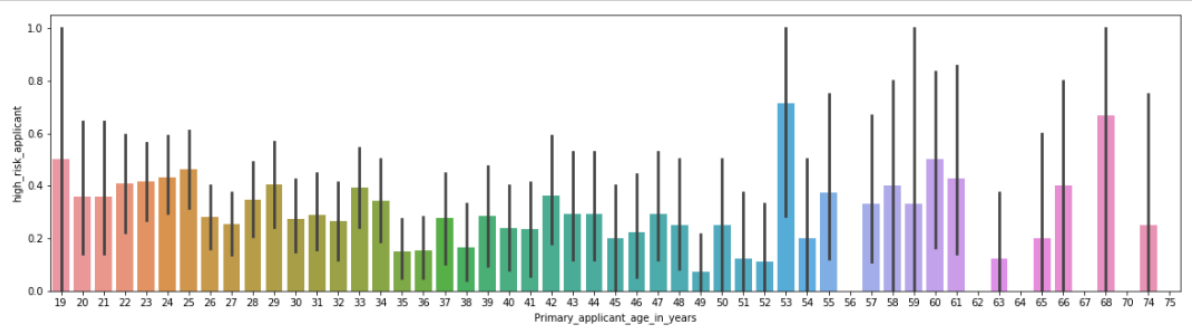
3. **Which of these segments / sub-segments would you propose be approved?**

   For e.g. Would a person with critical credit history be more creditworthy?



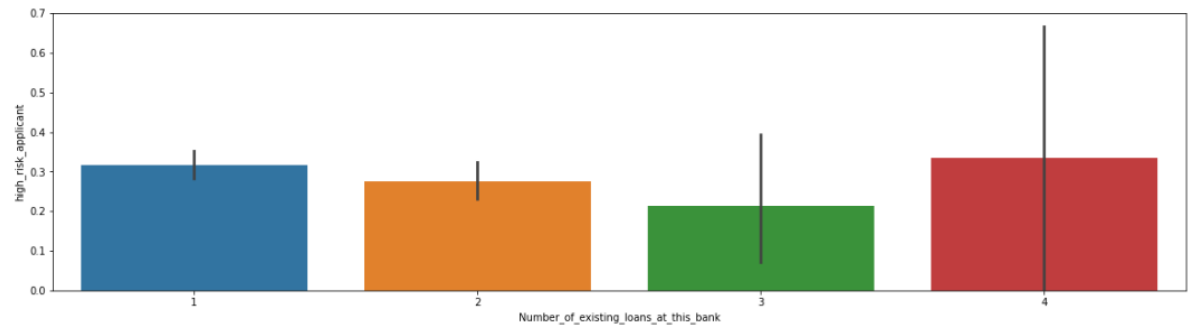
**Solution**: According to me, the person with a critical credit history is less creditworthy. But, from the visualization shown above it shows that the person with critical credit history is a low risk applicant and thus more creditworthy.So to summarize this, if we follow the data it seems person with critical credit history is more creditworthy

Are young people more creditworthy?




**Solution**: According to me, young people should be more creditWorthy. But if we follow the data, according to the visualizations we cannot conclude that as there is no pattern to suggest that

Would a person with more credit accounts be more creditworthy?




**Solution**: According to me a person with more no of existing loans in this same bank is less creditworthy as shown in the visualization that this person maybe a high risk applicant. So to summarize this, a person with more no of existing loans is less creditworthy

4. **Tell us what your observations were on the data itself (completeness, skews).**

**Solution:** The given data has around 27 features (1 Target column) and 1000 rows. There are 9 features having missing values. After looking at the data distribution it seems data is not normally distributed and some features are right skewed. So before applying ML algos to these data, data transformation is essential. Data is missing in some features

# Task 2:

1. **Explain your intuition behind the features used for modelling.**

   **Solution**: After performing feature Selection(Removing features having 0 varience and features which are correlated with each other more than 80%) we were left with 21 independent features

2. **Are you creating new derived features? If yes explain the intuition behind them.**

   **Solution**: No, I have not created derived features

3. **Are there missing values? If yes how you plan to handle it.**

   **Solution**: Yes,There were missing values in the given dataset. Upon some analysis it was found that the missing values were from categorical features.So for handling missing values in categorical features we just added a new Category with name "Missing"

4. **How categorical features are handled for modeling.**

   **Solution**: It is important that before modelling we need to convert categorical features into numerical features. So for that we first need to divide categorical features into ordinal and nominal features.In ordinal features rank matters(for eg. if

we are converting salary category column into numerical then we need to assign high values to the senior employees and low value to the junior employees).For nominal features order does'nt matter. For ordinal features technique call target guided encoding was used and for nominal features technique called mean encoding was used

5. **Describe the features correlation using correlation matrix. Tell us about few correlated feature & share your understanding on why they are correlated.**

   **Solution**: Correlation matrix is drawn in this notebook while performing feature Selection you can just refer that on the top.
   While keeping Threshold = 80% only 2 features were found correlating to each other.They are (has_been_employed_for_atleast) and (has_been_employed_for_atmost).
   They are correlated because they are related to experience

6. **Do you plan to drop the correlated feature? If yes then how.**

   **Solution**: Yes, I have dropped the correlated features by creating a function that takes input as a dataframe and a threshold value and then creating a correlation matrix and comparing each row with each column correlation value with the threshold value. If this values is greater than the threshold value then it will be inserted into the set variable and after, all rows and columns are traversed the set having correlated features will be returned

7. **Which ML algorithm you plan to use for modeling.**

   **Solution**: I tried to use all the classification algorithms known to me like Logistic regression,Descision Tree Classifier, Random forest Classifier, XGBoost Classifier,Support Vector Classifier,Naive Bayes.

8. **Train two (at least) ML models to predict the credit risk & provide the confusion matrix for each model.**

   **Solution**: For this solution please check the jupyter notebook on github or kaggle.

9. **How you will select the hyperparameters for models trained in above step.**

   **Solution**: In this solution I used optuna for hyperparameter tuning, so this libary provides the graphical visualization to show which hyperparameter contributes how much.The same is demonstrated above.Depending upon this contribution we can select the same

10. **Which metric(s) you will choose to select between the set of models.**

    **Solution**: As asked in the problem statement that it is ok to state an applicant to be a high credit risk when they aren't but not true vice versa.So I think we should focus more on reducing False negative and thus to do that we need to focus on recall to achieve it.

11. **Explain how you will export the trained models & deploy it for prediction in production.**

    **Solution**: We Can use joblib.dump() as well as pickle.dump() to save our model and then load it while doing prediction. We can deploy it into production after dockerizing it and then establishing a CI-CD pipeline to the production environment

# Results:

Among all the models, **Naïve Bayes** perfomed well considering the evalution metric to be **recall**. Because According to the business it was told that that it is worse to state an applicant as a low credit risk when they are actually a high risk, than it is to state an applicant to be a high credit risk when they aren't. Thus it means we should reduce the **False Negative** so thus **Recall**. The performance metrics are as follows:

In [86]:
```
y_pred=nb.predict(X_test)
print(confusion_matrix(y_test,y_pred))
print(accuracy_score(y_test,y_pred))
print(roc_auc_score(y_test, y_pred))
print(classification_report(y_test,y_pred))
```

```
[[229   0]
 [  0 229]]
1.0
1.0
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00       229
         1.0       1.00      1.00      1.00       229

    accuracy                           1.00       458
   macro avg       1.00      1.00      1.00       458
weighted avg       1.00      1.00      1.00       458
```

# Summary:

Thus to summarize the entire project here are some of the key points:

- Data was collected from 2 sources having 1000 records and then the data was merged into a single data source
- Exploratory data analysis was performed and some key insights were taken
- On the basis of EDA, Features Engineering was performed
- Feature Selection was done in order to reduce curse of dimensionality
- Pre-processed data was provided to various ML classification algorithms
- Among all algorithms, Naïve Bayes performed well with respect to recall metric as specified by the business constraint
- This model was saved as a file file and can be used for prediction when deployed to production using MLops pipeline.