# ML Model Development and Packaging:

We have crated a machine learning model which banks can leverage to approve or reject the credit card for customers. The bank has access to dataset which contains historical data about their old customers which is being used to build the model.

**Problem Type:** Supervised Binary Classification

**Data Availability:**

* dataset has following columns:
* user id
* gender
* date of birth
* work class
* education level
* education Num:
* education level as a continuous variable. E.g., Masters (14) > Bachelors (13)
* marital status
* occupation
* relationship: current relationship status
* capital gain: capital gain made in the last financial year in USD through investments
* capital loss: capital loss in the last financial year in USD through investments
* Hours per week: Number of hours worked by the person per week.
* approved: Whether customer was approved for the Credit Card.
* address
* email

**Model Development and Packaging:**

1. Data Analysis
   1. Target Distribution

**A graph of distribution of target variable

Description automatically generated**

* 1. Other Distribution and Relation with Target Checks

A screenshot of a chart

Description automatically generatedA graph of a person with a red bar

Description automatically generated with medium confidenceA graph of a bar graph

Description automatically generated with medium confidenceA graph showing a number of boxes

Description automatically generated with medium confidenceA graph of a number of applications

Description automatically generated

A graph showing the amount of time

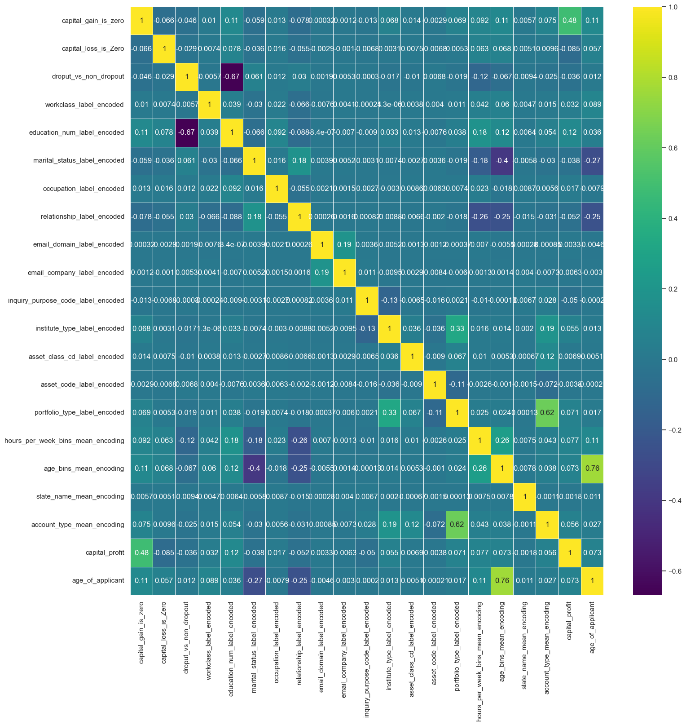
Description automatically generatedA graph of different occupations

Description automatically generated with medium confidence

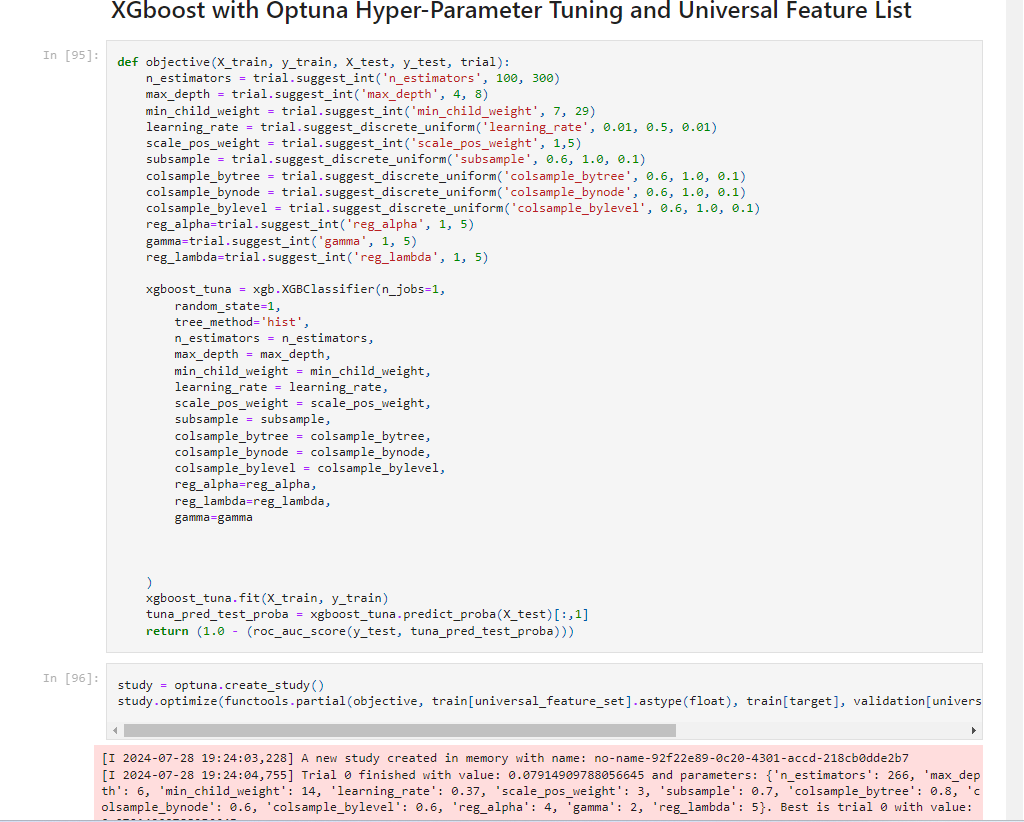
A graph of a graph of a number of individuals

Description automatically generated with medium confidence

1. Feature Engineering: Created sever new features using various ML techniques like:
   1. Extracting age from the date of birth and dividing it into different bins
   2. Calculating net profit from capital gain and capital loss
   3. Finding if capital gain and capital loss is 0 or any other positive number  
      4) Extracting Microfinance or Priority Sector information
   4. Defining dropout vs non-dropout
   5. Interaction of Domain with Occupation
   6. Extracting domain and company from email address
   7. Extracting name of the state from Address
   8. Label Encoding
   9. Target Encoding
   10. Direct Features
2. Feature Selection: Many features can affect the ML model and get the performance down rather than improving it. So, we ran multiple feature selection methods to select the best features.
   1. Feature Pearson Correlation to remove correlated features



* 1. Backward Feature Selection: Which removes one feature at a time to measure the drop in performance and thus remove the feature If the drop is not significant.
  2. Manul Selection: Manually selected features by estimating the business values and so that we can avoid using lot of features and increase the complexity.
     1. Final 10 features that were used : ['capital\_gain\_is\_zero', 'capital\_loss\_is\_Zero', 'workclass\_label\_encoded', 'education\_num\_label\_encoded', 'marital\_status\_label\_encoded', 'occupation\_label\_encoded', 'relationship\_label\_encoded', 'asset\_code\_label\_encoded', 'capital\_profit', 'age\_of\_applicant']

1. Model Building and Tuning: Created and validated multiple models on the dataset such as Xgboost and RandomForest.
   1. The hyper parameter tuning to get the best parameters for a given model was done using a Bayesian Optimization library called Optuna.
   2. Both the model were working good with No overfitting, though Xgboost was slightly better.

A screen shot of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Feature Importance: Measure the feature importance using the Random Forest feature importance techniques to find out which feature was more predictive than other.

A graph with a bar graph

Description automatically generated with medium confidence

1. Saving Model Artifacts: All the model artifacts that can be required for inferencing the model afterwards were stored using Pickle and Joblib libraries.

A screenshot of a computer

Description automatically generated

1. Model Inferencing: We have written a python file to perform inferencing using the stored artifacts. This logic can be used in the flask application to expose the API on a UI or though postman.

A screenshot of a computer

Description automatically generated



1. A screenshot of a computer code

   Description automatically generatedModel Packaging: We have packaged the model using the Docker, which is a nice containerization tool to abstract out all the libraries, code, python version etc. so that it can run any system by building and then running the docker image.

A screenshot of a computer

Description automatically generated

1. Docker Image Creation: Created the docker image by building the docker file and then also tested it using the flask API which get’s exposed by the docker image.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated