Department of Computer Science

Loan Amount Prediction By FCC

CS711 - Foundation of Data Science

Guided By:

Dr. Alireza Manashty

Prepared By:

Meet Patel - 200467513 – MDP965@uregina.ca

Vedant Patel - 200469201 – VPP336@uregina.ca

Role of Team Members

Meet Patel

Email: MDP965@uregina.ca

Expertise: Intermediate knowledge in python programming, Data Preprocessing

Role: Data Preprocessing, implementing models like Decision Tree, Random Forest and project report.

Vedant Patel

Email: VPP336@uregina.ca

Expertise: Basic knowledge of python, feature engineering, data cleaning and exploration.

Role: Feature scaling, Data Exploration, implementing models like Neural Network, XGBoost and Extra Tree Classifier, project report.

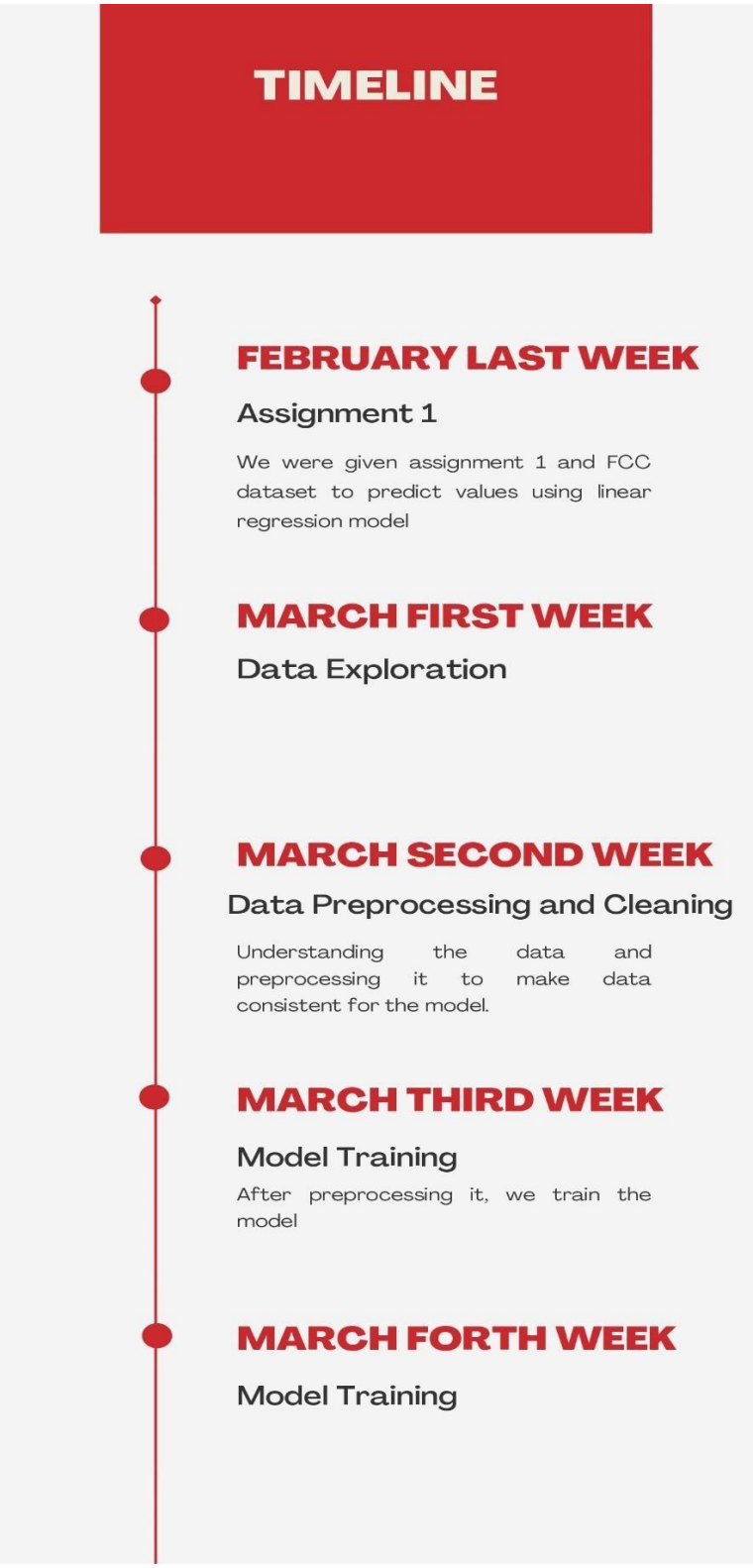
Description of Project

The major goal of this project is to use several machine learning models to determine an approximate capital amount to recover losses for the instrument in the event of any form of loss event. Calculate the capitalization rate, using formula which is by predicted value of capital amount divided by proportion of MTM exposure. This Project is given by FCC Canada.

Main Objective of the Project

- **Data exploration**: First we explored the dataset to see the various relationship between variables for training the model. To understand the data in better way we visualized the datausing various libraires like seaborn and matplotlib.
- **Data Pre-Processing**: With the help of data exploration, we performed data pre-processingon dataset like removing null values, dropping unnecessary columns.
- **Model Implementation**: In this project we performed various machine learning models likeDecision Tress, random forest, XgBoost, ExtraTreeClassifier and Neural Network.
- **Evaluating Result**: We observed that which model performed well in mean absolute errorand R2 score and after that we predicted the Capital -EL tail Risk Contribution on test dataand exported into csv file.

- PROJECT TIMELINE



APRIL FIRST WEEK

Model Evaluation

APRIL SECOND & THIRD WEEK

Exporting the Result and Checking Accuracy

Tried to increase the accuracy of the model using different techniques.

APRIL LAST WEEK

Project Report

PART 2

ABSTRACT

As per the note of risk assessment in any portfolio of clients, financial institutions are expected to hold capital reserves sufficient to withstand unlikely circumstances but potentially catastrophic loss events. Sometimes it is hard to forecast such events and on account of this, the reserve for risky loans is higher than the reserve for a safer loan. Likewise, this difference should be reflected in the interest rate offered by the lender. However, the loss calculations are too cumbersome to be scrutinized every time someone seeks a loan. The purpose of this project is to estimate those loss calculations, with a machine learning approach that could be fast enough to utilize during lending, but adequately accurate to avoid under- or over- pricing loans. As per given datasets, one is for development or training for the model and another to test or evaluate the target variables. The variables comprise both numerical and categorical data types. We dropped insignificant categorical variables. Firstly, we tried linear regression methods and observed that the ridge model yielded considerable accuracy with less mean absolute error (MAE). After performing exploratory data analysis and feature selection method, we obtained significant features and executed experiments with different platters of features. The approach of feature engineering results in an efficient way to train the input features with various models like decision tree, random forest, gradient boosting, xgboost, and extra trees. Likewise, we implemented a neural network architecture and observed that random forest leads to the overall performance. To sum up the assessment, we received the lowest MAE from the extra trees regressor, the highest accuracy from the decision tree, and overall best performance from the random forest model. Our thorough evaluation, proposed method, and outright insights can be exposed to financial sector risk assessment scrutiny to assist the loss calculations quickly with minimal effort.

Table of Content

Introduction

In this project, we strive to build an accurate and robust machine learning model to help the financial organization to predict the capital amount required to cover losses and using which they can also determine the capitalization rate which is a measure of the amount the firm stands to lose if an instrument fails (MTM Exposure) relative to the amount of capital required to cover losses.

To build machine learning model first we need to understand the data and need to preprocess it in such a way that it will give us a better result. So, first we have explored dataset using various libraries like pandas, seaborn and matplotlib. After exploration of we have found relationships of various features with our target variable and dropped unnecessary data. There were few features like Instrument and Counterparty Alias which doesn't need to be used as features for our machine learning model as Instrument contains all unique values and Counterparty Alias is also a unique identifier which won't help to predict our target variable. Also, after dropping few columns like Instrument and Counterparty we have found null values and as there were only few null values, we have dropped all of that as the dataset is large enough so it won't affect much if we drop null values instead of filling them with mean, median or mode. Once, we have removed unnecessary data in terms of columns and rows we have started implementing various machine learning algorithms to find most optimal model which gives us better MAE and R2 score.

Firstly, we have started our machine learning models PCA (Principal component Analysis) to reduce the dimensions or we can say number of columns from our dataset and trained it on various machine learning algorithms like Decision Tree, Random Forest, Gradient Boosting, XG Boost and Extra trees but we haven't found much promising results as in our given dataset we don't have much columns so there is no practical reason to reduce the number of columns. After, PCA we have tried with various other machine learning models without performing PCA and using selected features to train. At last, but not the least we have also tried to create Neural Network but there we haven't found much promising results despite of time consuming and resource intensive training.

Main Ideas

The major goal of this project is to use several machine learning models to determine an approximate capital amount to recover losses for the instrument in the event of any form of loss event. Calculate the capitalization rate, using formula which is proportion of MTM exposure divided by predicted value of capital amount. This Project is given by FCC Canada.

Main Steps

- **Data Exploration**
  - We have explored given dataset to identify relationships between feature and target variable which help us to select best features to train machine learning model.
  - To understand data in a better way we have visualized data with using various graphs and plot which gives us better understanding of data like how much features are correlated with each other, identify outliers and many more.
- **Data Pre-processing**
  - Based on our findings in data exploration we have performed various tasks in our dataset like removing null values, dropping unnecessary columns and removing outliers.
- **Model Implementation**
  - We have trained various machine learning models like Decision Tree, Random Forest, Gradient Boosting, XGBoost, ExtraTreesRegressor and Neural Network.
  - We have also performed PCA and tried to train all machine learning models mentioned above except Neural Network.
- **Evaluating Results**
  - We have observed which model outperformed well in giving both least mean absolute error and a good R2 Score.
  - After evaluating results, we have used our models to predict target variable i.e., Capital - EL Tail Risk Contribution on test dataset and based on that we have also find Capitalization Rate and exported it into a csv file.

Algorithms/Methods

- **PCA - Principal Component Analysis**
  - We have used principal component analysis (PCA) to reduce the dimensions of the given dataset and trained various machine learning models with PCA.
  - We have built a pipeline of standard scaler, pca, and regressor model which can be any machine learning model from Decision Tree, Random Forest, Gradient Boosting, XGBoost and ExtraTreesRegressor.
- **Decision Tree**
  - We have used DecisionTreeRegressor with 5 as maximum features.
  - When we have plotted a bar graph to see feature importance of all columns, we have found that mostly 5 features are contributing more than 80% to predict in decision tree.
  - We are able to achieve 5422.17 as MAE and 0.82 as R2 score.
  - This was the best model in terms of highest R2 score among all machine learning models.
- **Random Forest**
  - We have utilized RandomForestRegressor with 8 as maximum features and 100 as n_estimators which is a default.
  - We have selected 8 as maximum features based on feature importances.
  - We are able to achieve 4140 as MAE and 0.73 as R2 score.
  - This was the best model which have good R2 score as well as low MAE among all machine learning models.
- **Gradient Boosting**
  - We have used GradientBoostingRegressor to train our regression model with X_train and y_train which is in 5:1 ratio.
  - We are able to achieve 6948.01 as MAE and 0.66 as R2 score.
- **XGBoost**
  - We have used XGBRegressor with squarederror as objective which shows that we need to achieve least squarederror.
  - We are able to achieve 4428.25 as MAE and 0.64 as R2 score.
- **Extra Trees**
  - We have used ExtraTreeRegressor with X_train and y_train to train our regression model.
  - We are able to achieve 3904.1 as MAE and 0.76 as R2 score.
  - This was the best model in terms of least MAE among all machine learning model.
- **Neural Network**
  - We have built neural network with 2 dense layers and one input and one output layer.

- To build neural network we have used relu as activation function, mean_squared_error as loss function.
- We have utilized adam optimizer and trained our model with 50 epochs, 32 as batch size and 0.2 as validation split.
- After 50 epochs, we have achieved 13398.22 MAE and 0.47 as R2 score.

Dataset Description

Two .csv files are provided by FCC

1. Development dataset with all dependent variables
2. Test dataset with dependent variables

Feature Columns with Object Datatype

Column	Description	Count	Unique Values
Instrument Type	Financial type of Instrument	151532	2
Instrument Subtype	Financial subtype of the instrument	151500	2
Counterparty Alias	Unique user-defined identifier for counterparty	151532	56917
Industry Alias	Model defined industry groupings that instruments belong to	151532	40
Country of Incorporation	Country of incorporation of the counterparty	151532	1
Maturity Date	Maturity date of the instrument	151532	4562

Feature Columns with Numeric Datatype

Column	Description
Instrument	Unique user-defined identifier for the instrument
MTM Exposure	Mark to model exposure is the modeled estimate of the total possible losses the firm stands to lose in the event of instrument failure.
Book Exposure	The total possible losses the firm stands to

	lose in the event of instrument failure.
1-Year Likelihood	An estimate of the likelihood of an instrument default over the 1-year time horizon
Severity Used	Refers to eh severity of the recovery losses when an instrument defaults.
Commitment	The exposure's total commitment amount
Time to Maturity	The time before an instrument matures measured in years.
Asset Recovery correlation	Correlation between the asset and recovery processes for an instrument.
Recovery R-Squared for Simlation	It measures the proportion of the instrument's recovery risk that is systematic.
Asset R-Squared for Simulation	Asset R-squared is a measure that determines how much of the asset value is affected by the overall market.

Target Variable Description

Capital - EL Tail Risk Contribution (Capped)	The capital amount required to cover exposure losses for the instrument in the case of an extraordinary loss event in the portfolio.
Capitalization Rate - EL Tail Rist Contribution	Proportion of Capital and MTM Exposure.

Capitalization Rate (%) = Capital ($) / MTM Exposure ($)

IMPLEMENTATION DETAILS

All the required libraries are imported to explore, visualize, transform, train and predict. It always starts with exploring the datasets and observing the skewness and variance of data. We observed descriptive statistics of the dataset and figured out the relation of outliers with other variables. The datatype of the maturity date variable changed to datetime for efficient use.

The variables having categorical data type and insignificant to the model have been dropped, also capitalization rate is dependent variable on target variable so it would be used after predicting the target variable.

The different data visualization methods are implemented with target variables to analyze the relation. After visualizing, the maturity date variable is dropped as it is totally relatable with time to maturity variable. The heatmap of correlation portrays close correlation with MTM exposure and book exposure. It results in the highest correlation of target variables with MTM exposure, book exposure, and commitment variables.

The approach of feature scaling is used on instrument subtype and converted into binary by label encoding. We found the indexes of outliers from the time to maturity variable and dropped it for significant performance. In feature selection method, at first, we implemented pearson correlation using heatmap to perceive better insights, and mutual information method provides an ordinal approach to features. We carried out experiments with different threshold values but still all features result in remarkable performance. Therefore, all the features have been included while training and testing the dataset with considerable models.

Basically, we started with linear regression and achieved quite good accuracy from the ridge model. Then we fostered a pipeline to execute PCA with all concerned models and data standardized by standard scaler. A dictionary pool having all the models with considerable random state is used with a loop to perform and compare results of given models with PCA.

The decision tree model gives the highest accuracy above all and we visualized the feature importance through a horizontal bar graph which concludes commitment, severity used, 1-year likelihood, book exposure, and time to maturity are the topmost vital features among the variables.

We implemented a random forest model with the max 8 features and 44 random states, it seems to have overall great performance by this model. The gradient boosting hits high MAE among the models.

And XGBoost with regression squared error objective and default params makes it more appropriate for accuracy and mean absolute error. As per our insights, we tried the ExtraTrees regressor and it yields pretty good performance, the lowest MAE and considerable accuracy go with it.

Likewise, the neural network can be implemented on regression problems so we fostered a model comprising 4 dense layers with relu as activation function and input data has been scaled via standard scaler, the output layer goes with linear as activation function, and only one output. That results in quite considerably lower performance.

Lastly, all the trained models are utilized to predict the target variable and evaluate the capitalization rate. Later all the data has been appended in both given datasets and exported to CSV format.

Problems we faced & solved

The given dataset is raw and we need to pre-process it first to gather insights for model implementation. We observed that it contains both numerical and categorical data so we dropped the insignificant categorical data by visualizing the statistics of categorical data. The implementation of label encoding made it easy to add significant categorical variables as features. We exposed to outliers and found out the relation with another variable and considering the evaluation, we actually dropped the outliers. There was little confusion about resemblance between time and date variable but it was clear after data visualization, so the maturity date variable has been dropped. We are unable to achieve quite good results so the experiment with different random states and params led us to significant results. Additionally, we implemented one more model that gave us pretty good results.

Problems that we were not able to solve

We are unable to achieve good performance while using PCA technique with different models. Likewise, the neural network is implemented but after several experiments, we received 46.35% accuracy which is quite low to consider for further scope.

Material Learned in the Class and Used in the Project

Decision tree and Neural Network (brief): Learned in class, Random Forest, XGBoost, PCA, ExtraTree, Gradient Boosting algorithms: self-study.

Conclusion

Model Name	MAE	R2 Score
Decision Tree	5422.17	0.82
Random Forest	4140.78	0.73
Gradient Boosting	6948.01	0.66
XGBoost	4428.25	0.64
Extra Trees	3904.1	0.76
Neural Network	13398.22	0.46

The calculation of the loss function is more important when lending a loan to analyze the risk factor and handle the reserve funds for any circumstances. The purpose of this project is to estimate those loss calculations, with a machine learning approach that could be fast enough to utilize while lending the loans or bonds. The results of the regression showed that the proposed method has potential as far as different known quality metrics. In the dataset, there are 10 numeric variables and 6 categorical variables which excludes two target-dependent variables. In the wake of implementing and assessing, we observed that the best performing models are Random Forest, Decision Tree, and Extra Trees. Furthermore, we implemented feature scaling and PCA technique but the results were not as significant as individual performance. Later, models can be more robust and precise by applying feature engineering techniques. To sum up, test evaluation and outright insights can be implemented in financial institutions that will help to assist the smooth lending process quickly and precisely.

PART:3

Self-contained Setup Guide

To use our system user first needs to install Anaconda for Python which contains most of the libraries pre-installed which we have used.

Steps to run our system:

1. Install Anaconda for your operating system from this link
2. After installing it, open Jupyter Notebook
3. Navigate to the folder where our project is saved
4. Open Jupyter Notebook from that folder
5. Install required python packages as mentioned in requirements section from anaconda shell

Requirements

User needs to install certain packages before running our project as follows:

- Pandas
  - To install pandas run `pip install pandas` in anaconda shell
  - We have used pandas for exploring dataset in a form of dataframe
- Numpy
  - We have used numpy for data pre-processing
  - To install numpy run `pip install numpy` in anaconda shell
- Matplotlib
  - We have used pyplot from matplotlib to visualize data in form of graphs.
  - To install matplotlib run `pip install matplotlib` in anaconda shell
- Seaborn
  - We have utilized seaborn to visual data in form of graphs.
  - To install seaborn run `pip install seaborn` in anaconda shell
- Scikit-learn
  - We have extensively used scikit learn library for feature selection, feature scaling, splitting dataset, and performing PCA.
  - We have also used various machine learning models from sklearn.ensemble and sklearn.tree like Random Forest, GradientBoosting, Extra Trees and Decision Tree.

    - To install scikit-learn library user needs to run `pip install scikit-learn` in anaconda shell

- XGBoost
    - We have used XGBRegressor from XGBoost library to build XGBRegressor model.
    - To install XGBoost user needs to run `pip installs xgboost` in anaconda shell.
- Keras
    - We have used karas to build Artificial neural network.
    - To install karas run `pip install karas` in anaconda shell.
    - User also need to install TensorFlow using `pip install TensorFlow` in anaconda shell.

Code/Setup Files

- To run our project first user needs to assure that they have all required files in a same folder i.e. dataset and jupyter notebook.
- To run our project user can click on cell and Run all which will run all cells of Jupyter notebook.
- User can run individual cells from the notebook by click Run cells from cells or can use a shortcut key CTRL + Enter.

User-guide for Analysts

- First of all, you make sure that the project file and data you want perform this method remains in the same directory.
- After running all the required pre-processing and model training cells, you can proceed with the Final export of results section where you can analyze the dataset and calculate the loss function.
- Run all the cells and user will get the results of different models but the best performance considering MAE would be an Extra Tree model.
- At last, this method would return the results and calculate the capitalization rate which will be reflected directly in the output dataset.