Murali Krishna Chintha

Bankruptcy Prediction

Table of Contents

[1. Introduction 2](#_Toc42646077)

[1.1. Data 2](#_Toc42646078)

[1.2. Technologies Used: 4](#_Toc42646079)

[2. Process Flow 5](#_Toc42646080)

[2.1. Retraining the model 6](#_Toc42646081)

[2.1.1. Data Validation Check 6](#_Toc42646082)

[2.1.2. Data Splitting 6](#_Toc42646083)

[2.1.3. Data Pre-processing 7](#_Toc42646084)

[2.1.4. Training Different Models 7](#_Toc42646085)

[2.1.5. Choosing the best model based on the metric. 7](#_Toc42646086)

[2.1.6. Saving the model 8](#_Toc42646087)

[2.2. Bulk Prediction 8](#_Toc42646088)

[2.2.1. Data Validation Check 8](#_Toc42646089)

[2.2.2. Data Pre-processing 8](#_Toc42646090)

[2.2.3. Loading the model and predicting using the model 9](#_Toc42646091)

[2.3. Single Value Prediction 9](#_Toc42646092)

[2.3.1. Data Validation Check 9](#_Toc42646093)

[2.3.2. Loading the model and predicting the data. 10](#_Toc42646094)

[3. Contents 10](#_Toc42646095)

[3.1. Random Forest Classifier 10](#_Toc42646096)

[3.2. XGBoost Classifier 10](#_Toc42646097)

[3.3. GridSearchCV 10](#_Toc42646098)

1. Introduction

Bankruptcy prediction is the task of predicting bankruptcy and various measures of financial distress of firms. It is a vast area of finance and accounting research. The importance of the area is due in part to the relevance for creditors and investors in evaluating the likelihood that a firm may go bankrupt. The aim of predicting financial distress is to develop a predictive model that combines various econometric parameters which allow foreseeing the financial condition of a firm. The purpose of the bankruptcy prediction is to assess the financial condition of a company and its future perspectives within the context of long-term operation on the market . It is a vast area of finance and econometrics that combines expert knowledge about the phenomenon and historical data of prosperous and unsuccessful companies. Typically, enterprises are quantified by numerous indicators that describe their business condition that are further used to induce a mathematical model using past observation

## Data

The dataset is about bankruptcy prediction of Polish companies. The data was collected from Emerging Markets Information Service (EMIS]), which is a database containing information on emerging markets around the world. The bankrupt companies were analysed in the period 2000-2012, while the still operating companies were evaluated from 2007 to 2013.

Basing on the collected data five classification cases were distinguished, that depends on the forecasting period:

1. 1st year: The data contains financial rates from 1st year of the forecasting period and corresponding class label that indicates bankruptcy status after 5 years. The data contains 7027 instances (financial statements), 271 represents bankrupted companies, 6756 firms that did not bankrupt in the forecasting period.

2. 2nd year: The data contains financial rates from 2nd year of the forecasting period and corresponding class label that indicates bankruptcy status after 4 years. The data contains 10173 instances (financial statements), 400 represents bankrupted companies, 9773 firms that did not bankrupt in the forecasting period.

3. 3rd year: The data contains financial rates from 3rd year of the forecasting period and corresponding class label that indicates bankruptcy status after 3 years. The data contains 10503 instances (financial statements), 495 represents bankrupted companies, 10008 firms that did not bankrupt in the forecasting period.

4. 4th year: The data contains financial rates from 4th year of the forecasting period and corresponding class label that indicates bankruptcy status after 2 years. The data contains 9792 instances (financial statements), 515 represents bankrupted companies, 9277 firms that did not bankrupt in the forecasting period.

5. 5th year: The data contains financial rates from 5th year of the forecasting period and corresponding class label that indicates bankruptcy status after 1 years.

The data contains 5910 instances (financial statements), 410 represents bankrupted companies, 5500 firms that did not bankrupt in the forecasting period.

|  |  |
| --- | --- |
| **Features** | **Description** |
| **X1** | net profit / total assets |
| **X2** | total liabilities / total assets |
| **X3** | working capital / total assets |
| **X4** | current assets / short-term liabilities |
| **X5** | [(cash + short-term securities + receivables - short-term liabilities) / (operating expenses - depreciation)] \* 365 |
| **X6** | retained earnings / total assets |
| **X7** | EBIT / total assets |
| **X8** | book value of equity / total liabilities |
| **X9** | sales / total assets |
| **X10** | equity / total assets |
| **X11** | (gross profit + extraordinary items + financial expenses) / total assets |
| **X12** | gross profit / short-term liabilities |
| **X13** | (gross profit + depreciation) / sales |
| **X14** | (gross profit + interest) / total assets |
| **X15** | (total liabilities \* 365) / (gross profit + depreciation) |
| **X16** | (gross profit + depreciation) / total liabilities |
| **X17** | total assets / total liabilities |
| **X18** | gross profit / total assets |
| **X19** | gross profit / sales |
| **X20** | (inventory \* 365) / sales |
| **X21** | sales (n) / sales (n-1) |
| **X22** | profit on operating activities / total assets |
| **X23** | net profit / sales |
| **X24** | gross profit (in 3 years) / total assets |
| **X25** | (equity - share capital) / total assets |
| **X26** | (net profit + depreciation) / total liabilities |
| **X27** | profit on operating activities / financial expenses |
| **X28** | working capital / fixed assets |
| **X29** | logarithm of total assets |
| **X30** | (total liabilities - cash) / sales |
| **X31** | (gross profit + interest) / sales |
| **X32** | (current liabilities \* 365) / cost of products sold |
| **X33** | operating expenses / short-term liabilities |
| **X34** | operating expenses / total liabilities |
| **X35** | profit on sales / total assets |
| **X36** | total sales / total assets |
| **X37** | (current assets - inventories) / long-term liabilities |
| **X38** | constant capital / total assets |
| **X39** | profit on sales / sales |
| **X40** | (current assets - inventory - receivables) / short-term liabilities |
| **X41** | total liabilities / ((profit on operating activities + depreciation) \* (12/365)) |
| **X42** | profit on operating activities / sales |
| **X43** | rotation receivables + inventory turnover in days |
| **X44** | (receivables \* 365) / sales |
| **X45** | net profit / inventory |
| **X46** | (current assets - inventory) / short-term liabilities |
| **X47** | (inventory \* 365) / cost of products sold |
| **X48** | (profit on operating activities - depreciation) / total assets |
| **X49** | (profit on operating activities - depreciation) / sales |
| **X50** | current assets / total liabilities |
| **X51** | short-term liabilities / total assets |
| **X52** | (short-term liabilities \* 365) / cost of products sold) |
| **X53** | equity / fixed assets |
| **X54** | constant capital / fixed assets |
| **X55** | working capital |
| **X56** | (sales - cost of products sold) / sales |
| **X57** | (current assets - inventory - short-term liabilities) / (sales - gross profit - depreciation) |
| **X58** | total costs /total sales |
| **X59** | long-term liabilities / equity |
| **X60** | sales / inventory |
| **X61** | sales / receivables |
| **X62** | (short-term liabilities \*365) / sales |
| **X63** | sales / short-term liabilities |
| **X64** | sales / fixed assets |
| **Class** | 0 did not get bankrupt/ 1 - got bankrupt |

## Technologies Used:

|  |  |
| --- | --- |
| IDE | PyCharm |
| Database | MySQL |
| Frontend | HTML5, CSS3, Bootstrap |
| Integration | Flask |
| Deployment | Google Cloud Platform |

# Process Flow

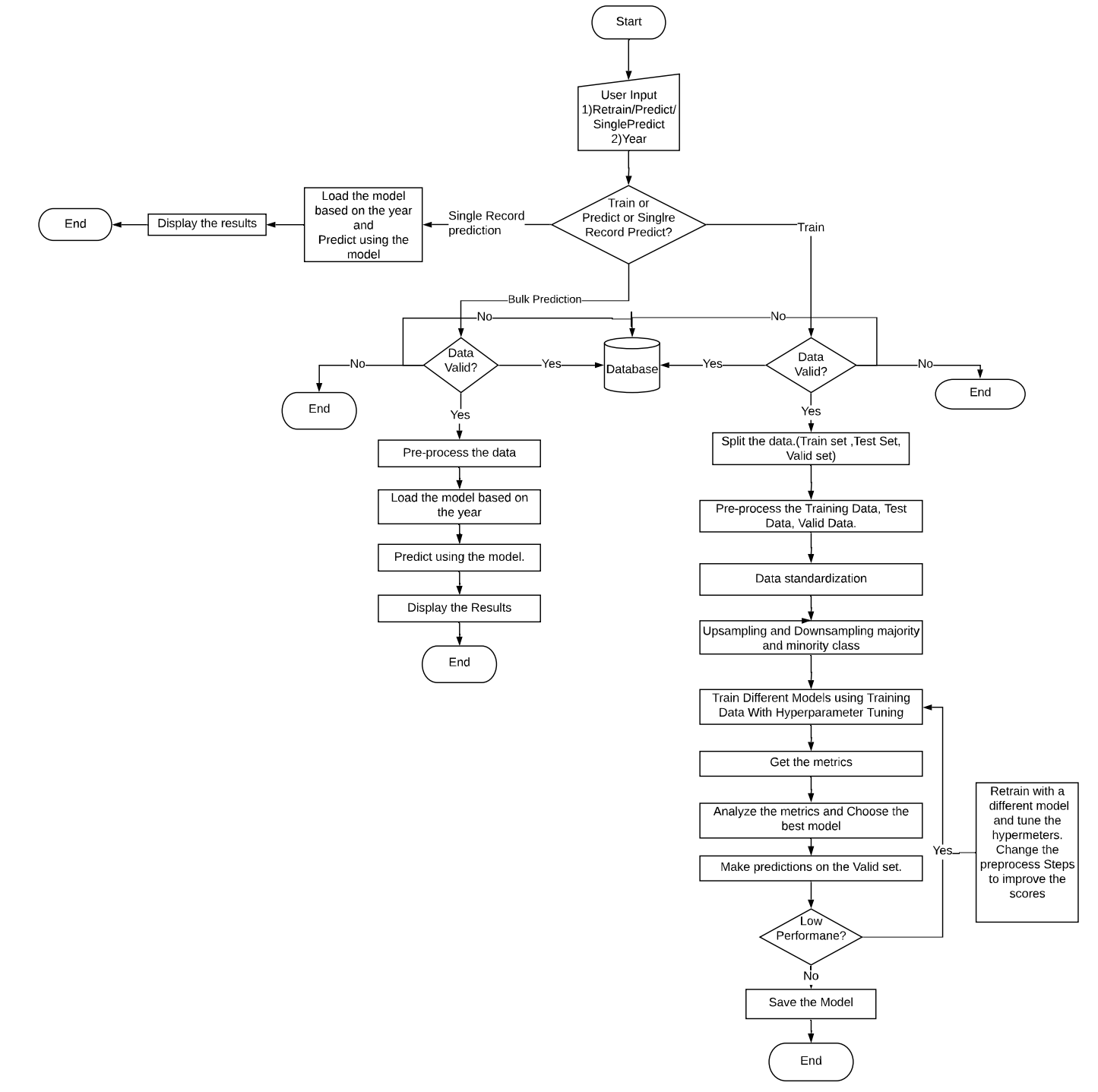
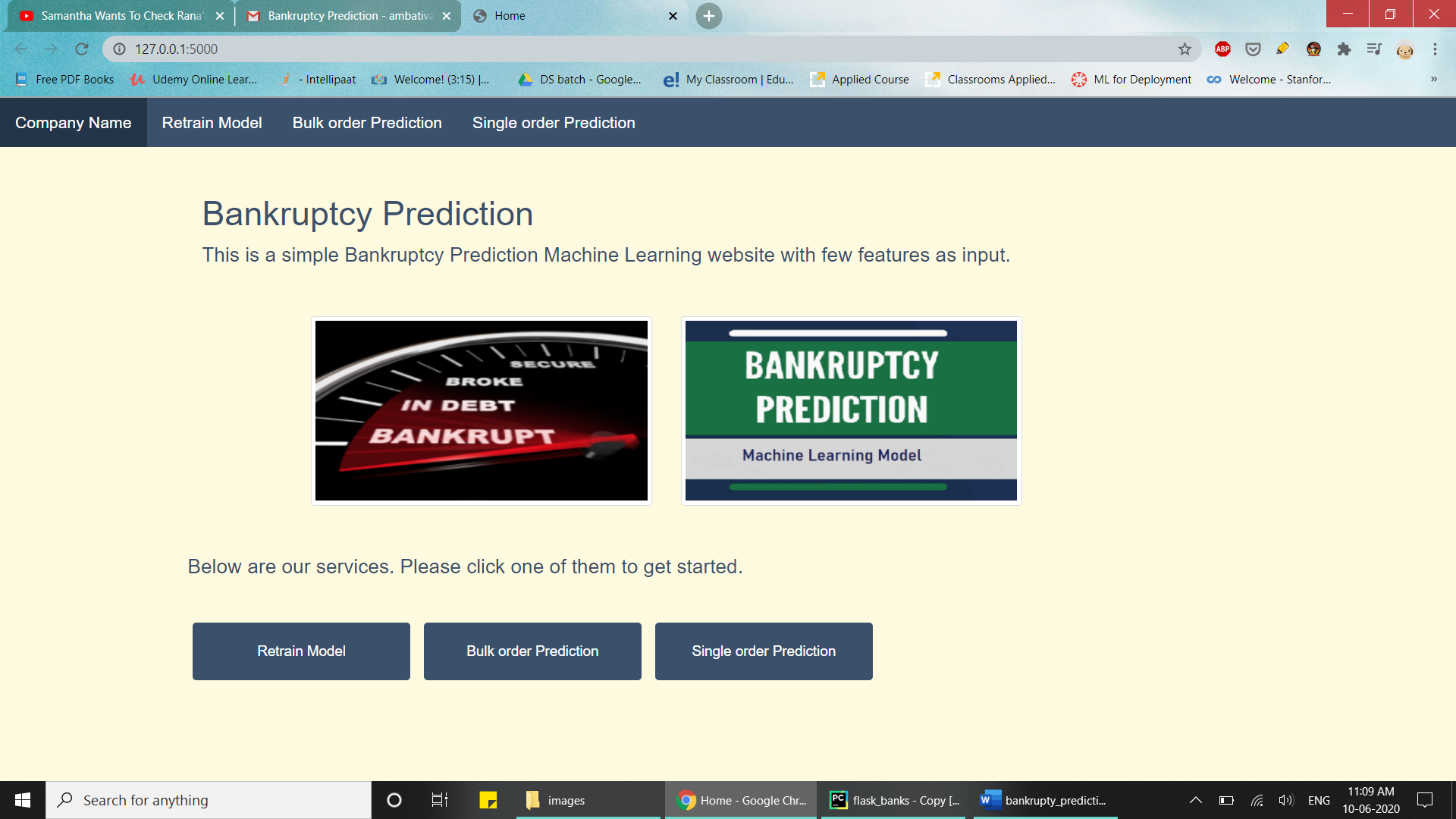


Fig. 1: Flowchart of the process.

The Flow chart above explains the process flow of the entire solution. There are three paths to the flow based on the user input –

1. Retraining the model
2. Bulk Prediction
3. Single Value Prediction.



## Retraining the model

The steps involved in retraining the model are

1. Data Validation Check
2. Data Splitting
3. Data Pre-processing
4. Training Different Models
5. Choosing the best model based on the metric.
6. Saving the model.

### Data Validation Check

Data Validation is performed to check if the data provided is valid or not. The main tasks in data validation check are –

1. Checking the number of columns agreed as per SLA
2. Checking the datatypes of each column agreed as per SLA.
3. Checking the column names agreed as per SLA.
4. Checking if any of the columns have more than 75% null values. In this case, data will be considered as invalid.

If the data meets all the four conditions then the data is considered to be Valid data. If it does not meet the conditions, data will be considered as invalid data.

Both Valid and Invalid data are pushed into the database.

### Data Splitting

The data for retraining the model will be split into three unequal sets in the ratio 70:15:15.

1. Training set.
2. Validation set.
3. Test set.

Training and Validation sets will be used for training different models and choosing the best one among them. Test set will be used to validate the chosen model.

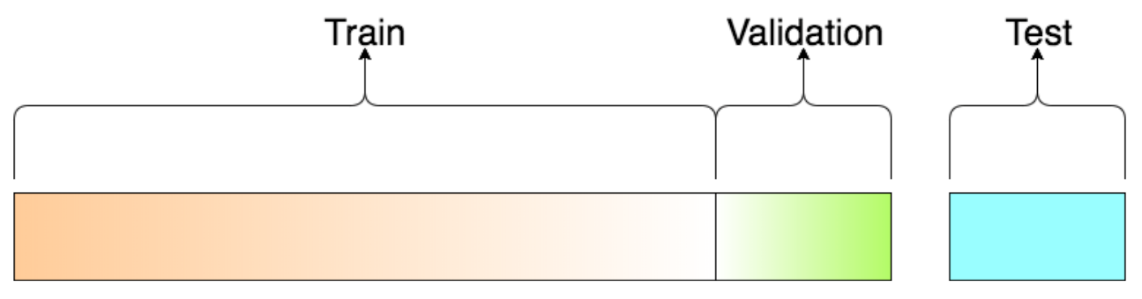


Fig 3: Diagram of Train Validation Test splitting of the data.

### Data Pre-processing

Data pre-processing is the most important step in training the model. In this step, we will prepare the data to be fed the model for training. Pre-processing includes –

1. Dealing with null values – Null values in the dataset are present as ‘?’. . Since the null value count in rest of the columns is very less. Dropping the null values does not affect the distribution of the data. So, the null values in other columns have been dropped.

Note: If the null values in any column are greater than 50% of the total data. The data will be considered as invalid.

1. Data Standardization – Since most of the features have different units and are measured in different scales, data standardization should be done to bring all the features to a same scale.

### Training Different Models

After pre-processing, the imbalance in the data should be handled by upsampling the minority class and downsampling the majority class in the data. After imbalance in the data is handled, data can be used to train different models. The different models used are

1. Random Forest Classifier
2. XGBoost Classifier

GridSearchCV has been used for hyperparameter optimization of the models. Each model will be trained on the pre-processed data using GridSearchCV.

### Choosing the best model based on the metric.

After training different and tuning the respective hyperparameters, best model will chose based highest F1-Score .. The metric used here is the F1-Score. The model with the highest F1-score will be chosen.

### Saving the model

After the model has been chosen. The existing model will be deleted and the new model will be dumped into a pickle file which can be used for loading the model any number of times.

## Bulk Prediction

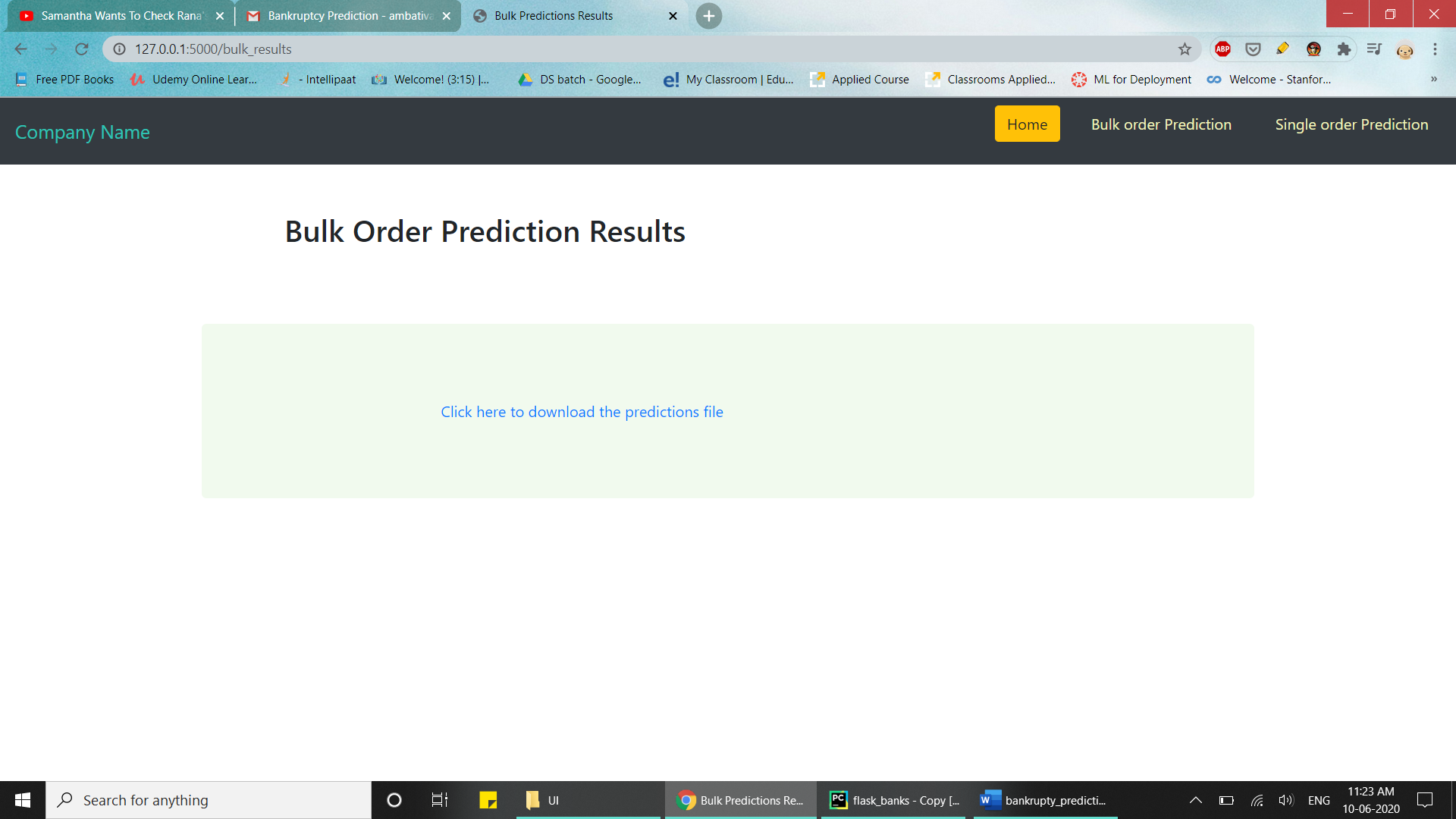


Fig 4: Page for Bulk predictions

The steps involved in predicting the model are

1. Data Validation Check
2. Data Pre-processing
3. Loading and Predicting using the model

### Data Validation Check

Data Validation is performed to check if the data provided is valid or not. The main tasks in data validation check are –

1. Checking the number of columns agreed as per SLA
2. Checking the datatypes of each column agreed as per SLA.
3. Checking the column names agreed as per SLA.
4. Checking if any of the columns have more than 75% null values. In this data will considered as invalid.

If the data meets all the three conditions then the data is considered to be Valid data. If it does not meet the conditions, data will be considered as invalid data.

### Data Pre-processing

Data pre-processing is the most important step in training the model. In this step, we will prepare the data to be fed the model for training. Pre-processing includes –

1. Dealing with null values – Null values in the dataset are present as ‘?’. . Since the null value count in rest of the columns is very less. Dropping the null values does not affect the distribution of the data. So, the null values in other columns have been dropped.

Note: If the null values in any column are greater than 50% of the total data. The data will be considered as invalid.

1. Data Standardization – Since most of the features have different units and are measured in different scales, data standardization should be done to bring all the features to a same scale.

### Loading the model and predicting using the model

After pre-processing the data for prediction, the saved model will be loaded from the pickle file and the prediction data will be given to the model as input. The results file can be downloaded.

## Single Value Prediction

The user can choose to enter the values manually or can upload the CSV file containing a single record.

The Steps for prediction if the user chose to enter the values manually are

1. Data Validation Check.
2. Loading the model and predicting the data.

### Data Validation Check

Data Validation is performed to check if the data provided is valid or not. The main tasks in data validation check are –

1. Checking the data types of each of the variables entered by the user.
2. Checking if any of the entered values are null values.

If the data meets all the conditions then the data is considered to be Valid data. If it does not meet the conditions, data will be considered as invalid data.

### Loading the model and predicting the data.

After validating the data for single value prediction, the saved model will be loaded from the pickle file and the prediction data will be given to the model as input. The output file can be downloaded.

# Contents

## Random Forest Classifier

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. In random forests, each tree in the ensemble is built from a sample drawn with replacement from the training set. Also, when splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features. Instead, the split that is picked is the best split among a random subset of the features. As a result of this randomness, the bias of the forest usually slightly increases but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model. In our model, the number of estimators used are 5 and we have considered ‘Entropy’ as a measure of the quality of a split.

## XGBoost Classifier

Extreme Gradient Boosting (XGBoost) is built on the principles of gradient boosting framework. Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function. XGBoost uses a more regularized model formalization to control over-fitting, which gives it better performance. In our model, the number of estimators used are 100. The model internally uses log-linear classifier for regularizing the model with λ = 1.

## GridSearchCV

GridSearchCV is a library function that is a member of sklearn’s model\_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters.

In addition to that, you can specify the number of times for the cross-validation for each set of hyperparameters.