Project Report

CS 5790 Data Mining – Spring 2023

Dr. Yijun Zhao

Manali Chordia, Arna Sadia

Table of Contents

1.1 Introduction

An individual's annual income results from various factors. Intuitively, it is influenced by the individual's education level, age, gender, occupation, etc. The dataset used in the project contains an individual's educational, demographic, and family information and was taken from the census bureau database found at http://www.census.gov/ftp/pub/DES/www/welcome.html. The dataset is represented in a standard format, consisting of 3 files: “census-income.names” which describes the categories and features of the dataset, census-income.data which is the training data, and census-income.test which is the test data. In this report, we will analyze the behavior of multiple classification algorithms using the ensemble technique to determine the best results. The classification algorithms used in this project are K nearest neighbor, Logistic Regression, Naïve Bayes, and Random Forest.

2.1 Problem Statement

Using census bureau data, the aim is to build a predictive model that determines income level for adults. Income levels are classified in two classes below $50K and above $50K annually (given in the dataset).

2.2 Dataset

Our census data contained 15 variables of three distinct types: continuous, categorical and ordinal.

2.2.1 Continuous Variables

- age
- wgt
- capital-gain
- capital-loss
- hours_per_week

2.2.2 Categorical Variables

- workclass
- education
- marital_status
- occupation
- relationship
- sex
- race
- native
- Country
- income

We converted the Categorical data (age , income) to Numeric Data.

2.2.3 Ordinal Variables

- education_id

3.1 Data Preprocessing

Before training the classification algorithms, we performed preprocessing steps on our dataset such as handling missing values, duplicate values, removed extra spaces from text, converting the categorical data to numeric data, encoding categorical variables using the One Hot Encoding, and scaling continuous variables using methods like Normalization.

3.1.1 Handling missing values

After a preliminary exploration of the census data, we found that both training and testing data sets contained missing values. For training and test data alike, all of the missing values were confined to three categorical values: native_country, workclass and occupation. We replaced the missing value with a new category 'Unknown'.

3.1.2 Handling Duplicate values

We found that both training and testing data contained duplicate records, there were 24 duplicate records in the training data and 5 duplicate records in the testing data.

3.1.3 Converting the categorical data to numeric data

We converted the categorical variables (age and income) to numeric data as follows:

Age {'Male':1,'Female':0}

Income {'>50K':1,'<=50K':0}

3.1.4 Z score Normalization

In normalization, giving scores a common standard of zero mean and unity standard deviation facilitates their interpretation. This is a common procedure in statistics because values that roughly follow a standard normal distribution are easily interpretable. We use Z-score normalization which replaces the measurement unit with "number of standard deviations" away from the mean. Hence, it's a convenient tool when someone wants to compare two variables that are measured in different units. We did z score normalization on Continuous Variables (age, wgt, capital_gain, capital_loss, hours_per_week).

3.1.5 Encoding

As we found during our initial data analysis, there were few categorical features in the given dataset. There are many machine learning algorithms which can support categorical features in computation without any manipulations but there are many more which do not support. Machine learning algorithm use for this project does not support the categorical feature directly and requires further manipulation in the data. Therefore, we had to figure out how to turn these categorical features into numerical features for algorithm processing. We used the one hot encoding method to convert the categorical features to numerical features.

4.1 Splitting Data

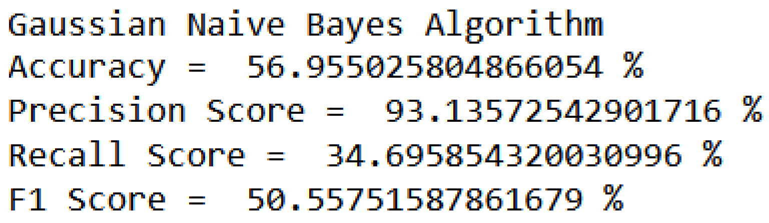
Then we split the dataset into X_train, y_train, X_test, y_test to split the features and the target values in order to train the dataset on the input values to predict the target variable.

5.1 Algorithms

We will first find the accuracy , precision, recall and F1 score of each model independently and then combine all the models using the ensemble technique to check if the accuracy increases or decreases.

5.1.1 Gaussian Naive Bayes Algorithm

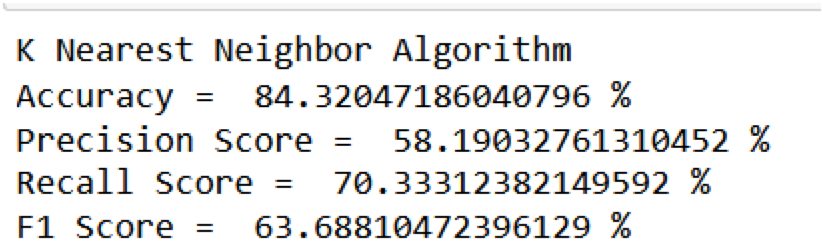
Gaussian Naive Bayes is a probabilistic algorithm that uses Bayes' theorem to predict the probability of a particular event based on prior knowledge. It assumes that all features are independent of each other and calculates the probability of each feature to determine the probability of a particular class. Naive Bayes is widely used in text classification and spam filtering. We got the below results when we ran the gaussian naive bayes algorithm independently.



```
Gaussian Naive Bayes Algorithm
Accuracy =  56.955025804866054 %
Precision Score =  93.13572542901716 %
Recall Score =  34.695854320030996 %
F1 Score =  50.55751587861679 %
```

5.1.2 K-Nearest Neighbors Algorithm

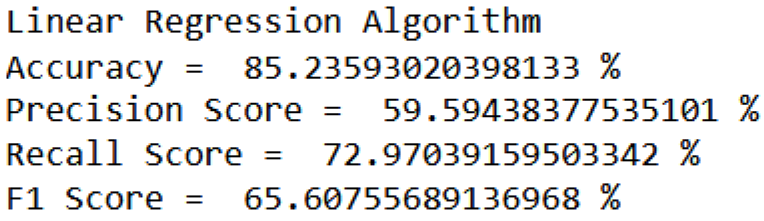
KNN is a non-parametric algorithm that uses a distance metric to find the K nearest neighbors of a given data point. It then predicts the class of the data point based on the majority class of its nearest neighbors. KNN is a simple but effective algorithm that can be used for both regression and classification. We used the k-neighbors = 80. We got the below results when we ran the K-Nearest Neighbors Algorithm independently.



```
K Nearest Neighbor Algorithm
Accuracy =  84.32047186040796 %
Precision Score =  58.19032761310452 %
Recall Score =  70.33312382149592 %
F1 Score =  63.68810472396129 %
```

5.1.3 Logistic Regression Algorithm

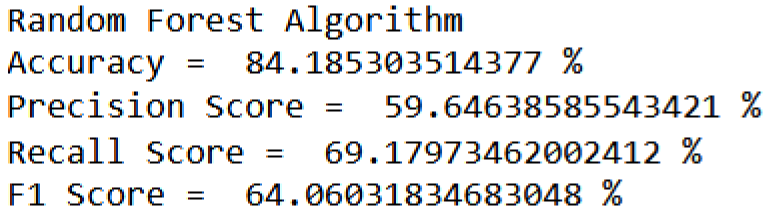
Logistic Regression is a statistical algorithm that is used for binary classification problems. It models the probability of a binary outcome based on one or more predictor variables. We got the below results when we ran the Logistic Regression Algorithm independently.



```
Linear Regression Algorithm
Accuracy =  85.23593020398133 %
Precision Score =  59.59438377535101 %
Recall Score =  72.97039159503342 %
F1 Score =  65.60755689136968 %
```

5.1.4 Random Forest Algorithm

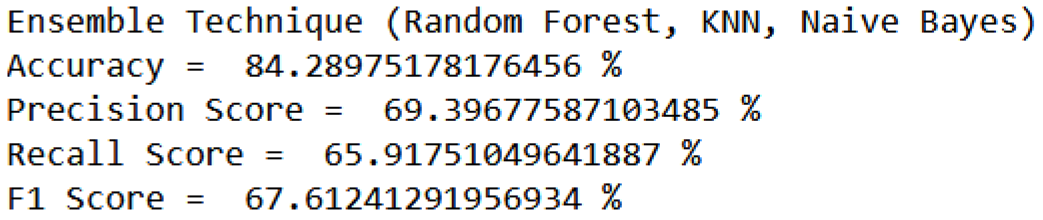
Random Forest is an ensemble algorithm that uses multiple decision trees to make predictions. It randomly selects a subset of features and samples from the dataset to create each decision tree, and then combines the predictions of all trees to make a final prediction. Random Forest is a powerful algorithm that is widely used in classification and regression tasks. We used n_estimators = 80. We got the below results when we ran the Random Forest Algorithm independently.



```
Random Forest Algorithm
Accuracy =  84.185303514377 %
Precision Score =  59.64638585543421 %
Recall Score =  69.17973462002412 %
F1 Score =  64.06031834683048 %
```

5.1.5 Ensemble Technique(Random Forest, KNN, Naive Bayes)

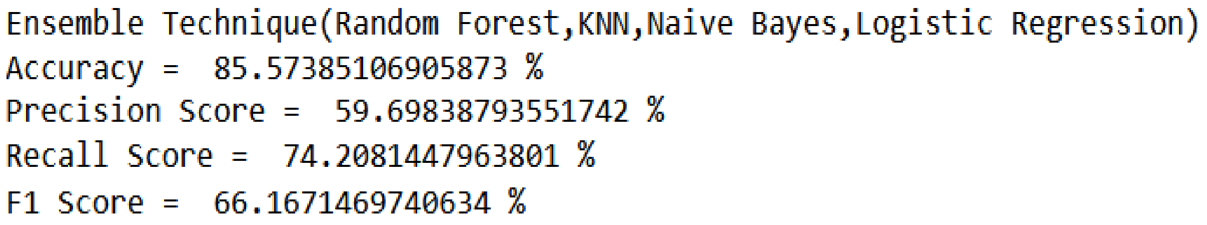
The ensemble technique combines the predictions of multiple models to improve accuracy and reduce variance. In this case, we will combine Naive Bayes, KNN, and Random Forest algorithms to create an ensemble model. The ensemble technique will make a final prediction based on the majority vote of the three algorithms.Below are the results of ensemble technique(Random Forest, KNN, Naive Bayes).



```
Ensemble Technique (Random Forest, KNN, Naive Bayes)
Accuracy =  84.28975178176456 %
Precision Score =  69.39677587103485 %
Recall Score =  65.91751049641887 %
F1 Score =  67.61241291956934 %
```

5.1.6 Ensemble Technique(Random Forest, KNN, Naive Bayes, Logistic Regression)

In this case, we will combine Naive Bayes, KNN, and Random Forest and Logistic Regression algorithms to create an ensemble model. The ensemble technique will make a final prediction based on the majority vote of the three algorithms.Below are the results of the ensemble model (Random Forest,KNN,Naive Bayes,Logistic Regression).



```
Ensemble Technique(Random Forest,KNN,Naive Bayes,Logistic Regression)
Accuracy =  85.57385106905873 %
Precision Score =  59.69838793551742 %
Recall Score =  74.2081447963801 %
F1 Score =  66.1671469740634 %
```

6.1 Comparison of Algorithms

	Random Forest	K Nearest Neighbor	Gaussian Naive Bayes	Logistic Regression	Ensemble Technique (Random Forest, KNN, Naive Bayes)	Ensemble Technique (Random Forest, KNN, Naive Bayes, Logistic Regression)
Accuracy	84.22 %	84.32 %	56.96 %	85.24 %	84.29 %	85.57 %
Precision	60.06 %	58.19 %	93.14 %	59.60 %	69.40 %	59.70 %
Recall	69.12 %	70.33 %	34.70 %	72.97 %	65.92 %	74.21 %
F1 score	64.27 %	63.69 %	50.56 %	65.61 %	67.61 %	66.17 %

7.1 Conclusion

In this report, we discussed the use of Naive Bayes, KNN, Random Forest, and Logistic Regression algorithms in an ensemble technique to improve classification accuracy using the US Census Bureau dataset. We applied the ensemble technique to the dataset and achieved an accuracy of 85.57%, which is higher than the individual accuracies of Naive Bayes, KNN, and Logistic Regression and Random Forest. The ensemble technique is a powerful approach to data mining that can be used to combine the strengths of multiple algorithms and improve overall performance.