Census Bureau: Income Prediction

CISC-6930 - Data Mining - Spring 2020 - Final Report

Arlind Stafaj, Dino Becaj, Nick Howard, Jonathan Uy

INTRODUCTION

In this project, we will be exploring different classification algorithms on a real world dataset and report our experimental results. We used Python to implement our project because it expedites the computing process.

The data examined for this project was extracted from the census bureau database. Each instance contains an individual's educational, demographic and family information. Income level is separated into two classes - less than or equal to $50,000 and greater than $50,000.

OBJECTIVES

Using the classification algorithms we covered during class we will implement a method that best predicts the two class labels (<=50K, >50K). We are given a test data set containing the same features as our training data set. To accomplish this we will preprocess and clean our data, create an algorithm that provides the best result, and generate the criteria we will use to measure our success. We will then use accuracy, false positive rate, false negative rate, and various other statistics to measure our program's success.

DATA

Variables:

- Our census data contained 16 variables of two distinct types: continuous and categorical.
    - Continuous Variables:
        - age
        - fnlwgt
        - education-num
        - capital-gain
        - capital-loss
        - hours-per-week
    - Categorical Variables:
        - workclass

- state-gov
- education
- assoc-voc
- marital-status
- occupation
- relationship
- race
- sex
- native-country

Missing Values:

- For training and test data alike, all of the missing values were confined to three categorical values: native_country, workclass and occupation.

Unbalanced Data:

- The training data was unbalanced with a negative skew:
    - Probability for the class label '>50K' : 23.93% and 24.78% without unknowns.
    - Probability for the class label '<=50K' : 76.07% and 75.22% without unknowns.

DATA PRE-PROCESSING

- After observing the census data, we discovered that both training and testing data sets contained unknown values. In order to clean the data, we first decided to fill the unknowns rather than removing them. In addition we decided to drop the education feature as it was repetitive (because we already have education_num). We chose to fill rather than remove data in order to not lose thousands of instances that could help be a better predictive measure.
- In order to analyze the data, we decided to split the continuous features from the categorical ones. We did so in the following ways:
    - For the continuous data we used the KNN imputation method with k = 7 and kept weights uniform throughout.
    - For the categorical data, since all of our data was nominal and not ordinal, we used a simple imputer method to fill data with those most frequent.

- The last column of the data described the individual's income and represented the classification label. We changed this label to be 0 when less than or equal to $50,000 and 1 when greater than $50,000.
- We then proceeded to normalize the continuous features by using z-score normalization.
- We decided to use ordinal encoding for our categorical data so we can compute the pearson correlation coefficient (PCC) of each feature accurately.
- We then proceeded to balance our data using up-sampling and down-sampling methods.
    - Random up-sampling suggests selecting the minority examples multiple times to create a balanced dataset.
    - Random down-sampling suggests selecting a subset of the majority examples to match the number of minority examples to create a balanced dataset.
- We were then able to compute the PCC which would measure the statistical correlation between the variables.
- Based on the PCC values, we decided to use the filter method for feature selection. We chose the filter method because the wrapper method would take too much time to process and our algorithms were all independently classifying the data.
    - The filter method ranks features independently of the algorithm and selects the top ranked features. Some advantages of using this method are that it is very fast and will not overfit the data.
    - The wrapper method uses a classifier to assess the features.
- During filter selection we converted our data from ordinal encoding to hot one encoding to provide more accurate distance measurements since our categorical data was nominal. We originally had ordinal encoding for discrete features so we can isolate each row when transforming them into one hot encoded features. This allowed for accurate filter selection because otherwise during feature selection each would consider each one hot encoded data column as an independent feature. We also dealt with any discrepancies between test and training data features. Sometimes within the data, one feature had more characteristics represented in the test set than the training set or vice versa. To solve this discrepancy, we added a row of zeros for the missing characteristics which complied with one hot encoder for the calculation of distance.

RESULTS

Ensemble Training Set Accuracies

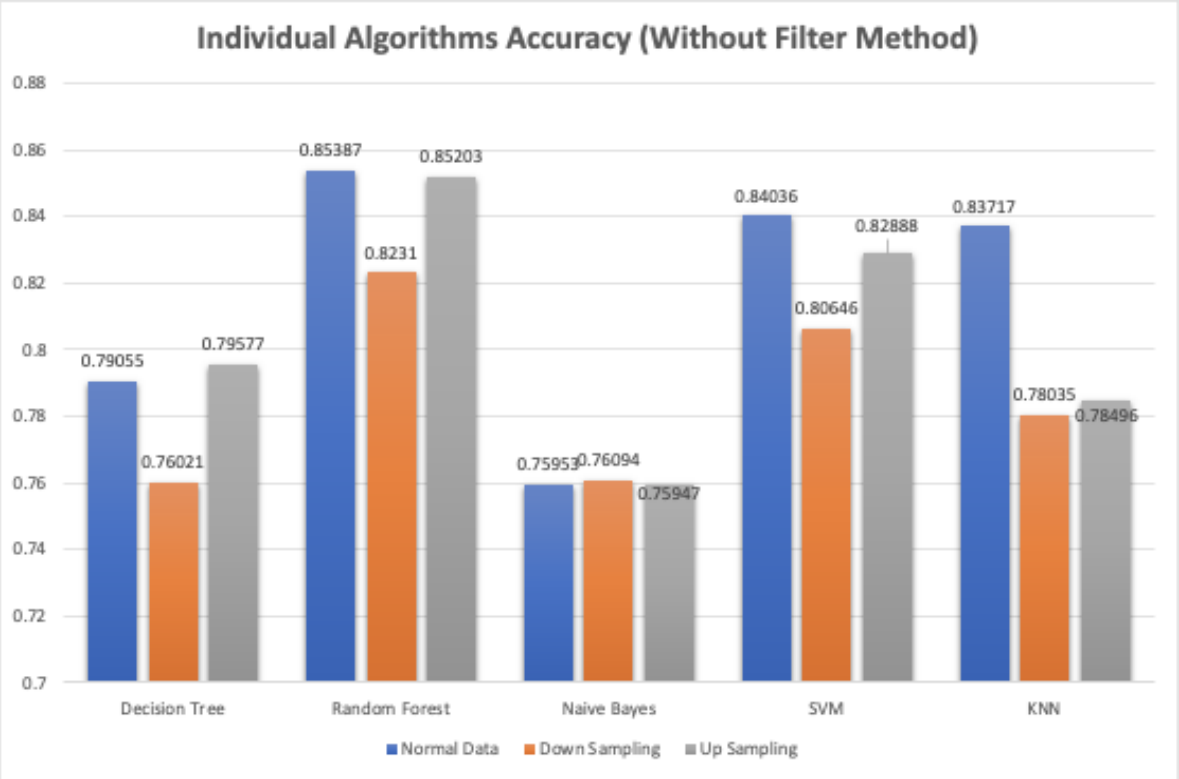
Normal Data ensemble Training set accuracy: 0.9542090230644023
Recall: 0.9674786379288356
Precision: 0.8598957152573112
F-measure: 0.9105203144691832

Down Sampling ensemble Training set accuracy: 0.8958678739956638
Recall: 0.9776814181864558
Precision: 0.840201665935993
F-measure: 0.9037430002947245

Up Sampling ensemble Training set Total Accuracy: 0.9093648867313916
Recall: 0.9837783171521035
Precision: 0.856332969470756
F-measure: 0.9156422372409119

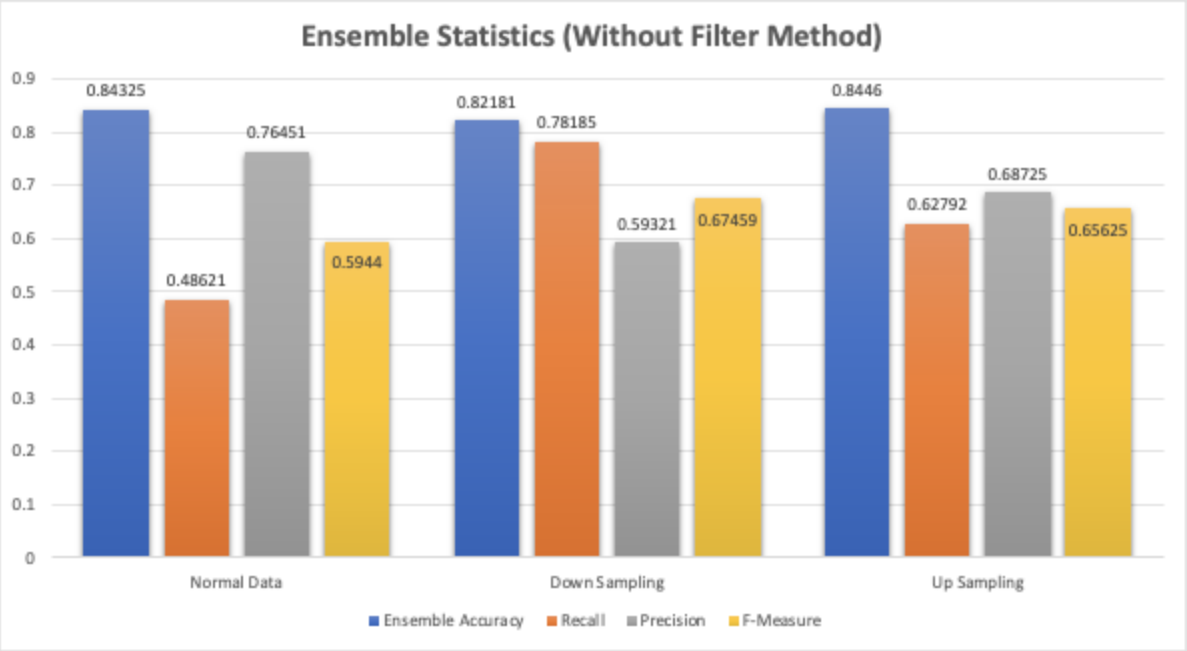
These are the accuracies of each model on each data set.

WITHOUT FILTER METHOD (Feature Selection)	Normal Data	Down-Sampling	Up-Sampling
Decision Tree	0.79055	0.76021	0.79577
Random Forest	0.85387	0.8231	0.85203
Naive Bayes	0.75953	0.76094	0.75947
SVM	0.84036	0.80646	0.82888
KNN	0.83717	0.78035	0.78496

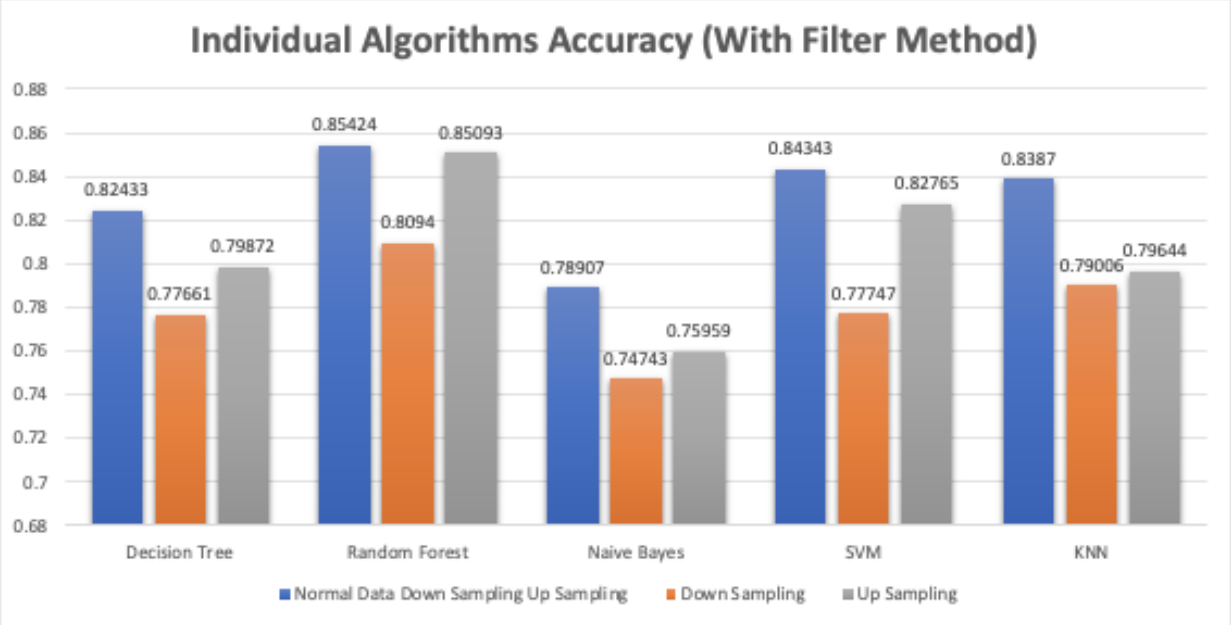


These are the statistics generated by each of the data sets.

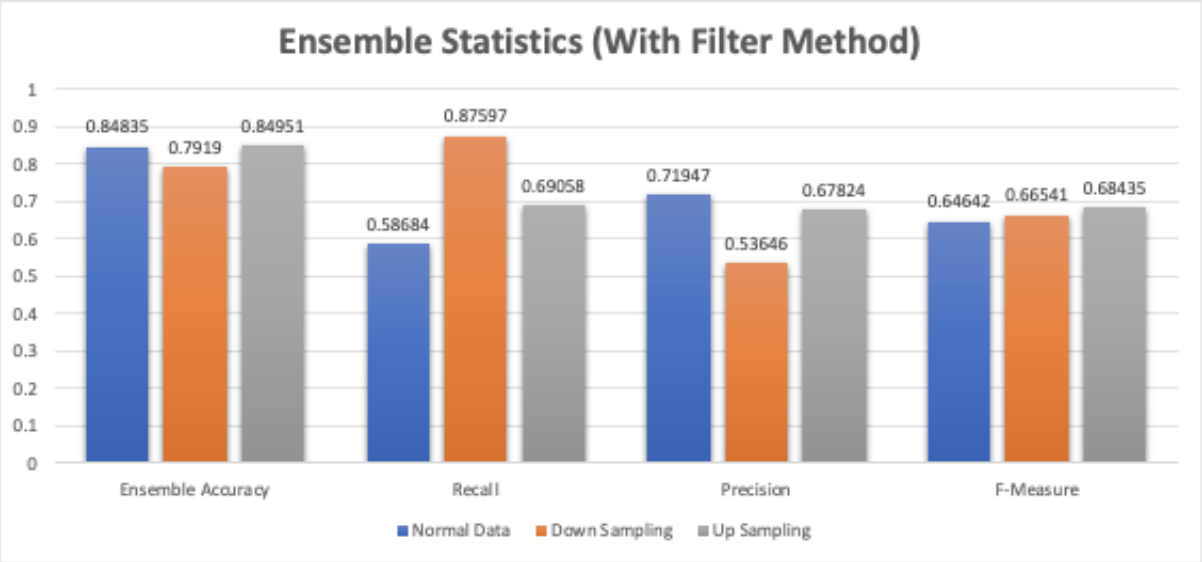
WITHOUT FILTER METHOD (Feature Selection)	Normal Data	Down-Sampling	Up-Sampling
Ensemble Accuracy	0.84325	0.82181	0.8446
Recall	0.48621	0.78185	0.62792
Precision	0.76451	0.59321	0.68725
F-Measure	0.5944	0.67459	0.65625



WITH FILTER METHOD	Normal Data	Down-Sampling	Up-Sampling
Decision Tree	0.82433	0.77661	0.79872
Random Forest	0.85424	0.80940	0.85093
Naive Bayes	0.78907	0.74743	0.75959
SVM	0.84343	0.77747	0.82765
KNN	0.83870	0.79006	0.79644



WITH FILTER METHOD	Normal Data	Down-Sampling	Up-Sampling
Ensemble Accuracy	0.84835	0.79190	0.84951
Recall	0.58684	0.87597	0.69058
Precision	0.71947	0.53646	0.67824
F-Measure	0.64642	0.66541	0.68435



ALGORITHM

- We decided to go about getting our results using the ensemble method and classifying predictions with a majority vote bagging classifier. Ensemble models in machine learning combine the decisions from multiple models to improve the overall performance. The main causes of error in learning models are due to noise, bias and variance. Ensemble methods help to minimize these factors. Ensemble methods are designed to improve the stability and the accuracy of Machine Learning algorithms.
    - Bagging is the method by which we create random samples of the training data with replacement and classify each sample through the model. Then by using the majority vote, we can classify the data.
    - By obtaining samples of the training data and then using a collective output, we avoid the problem of overfitting the data. Therefore, bagging is a great way to reduce variance.
- We used five algorithms for the ensemble method:
    - Decision trees:
    - Gaussian Naive Bayes
    - Random Forest
    - K-nearest Neighbors
    - Support Vector Machines.
- Accuracy for each algorithm was calculated using the normal, up-sampled, and down-sampled data sets.
- We then proceeded to compute the confusion matrix for the ensemble results of this data and got the overall accuracy, precision, recall, and f-measure computed with the false positive, false negative, true positive and true negative statistics

CONCLUSION AND FUTURE

- Our results predicted values with relatively high accuracy. This project gave us a better understanding of the classification models used to predict data. We also enjoyed the learning process of implementing different ways of preprocessing the data and using different implementations of the algorithms to produce different accuracies of the results.
- In the future, we would like to try out a new way of balancing the data such as Adaboost and SMOTE sampling. Furthermore, we would also like to approach feature selection with the Wrapper Method and measure test accuracies with selecting those features produced.