CMPE 255 - Data Mining

Income Classification

Team 4

Submitted to

Gheorghi Guzun

on

12/01/2020

Bhavana Banglore Sathyaprakash	014597245	bhavana.bangaloresathyaprakash@sjsu.edu
Raaga Pranitha Kolla	011824410	raagapranitha.kolla@sjsu.edu
Namratha Bhat	014597258	namratha.bhat@sjsu.edu

TABLE OF CONTENTS

1. Introduction

In today’s world, reducing inequalities and ensuring no one is left behind are integral to achieving the Sustainable Development Goals. Hence most of the political and economic agendas are targeted on hightning the income of a person and country as a whole. Income classification has found its permanent niche in these places. We wanted to take up income classification modeling and apply our knowledge of data mining to test our skills on this data.

1.2 Objectives

The Objective of our project is to classify and successfully predict the income of a person(if he earns above 50000 or less than 50000) based on features such as education level, country of birth, race, etc available in the dataset.

2. System Design and Implementation Details

2.1 Algorithms Experimented With

1. Linear Regression
2. DecisionTreeClassifier
3. Gaussian NB
4. KNeighborsClassifier
5. RandomForestClassifier
6. LinearDiscriminantAnalysis
7. ExtraTreesClassifier

2.1.1 Algorithms selected for final analysis

The algorithms selected for final analysis are Decision Trees, Random Forests and ExtraTreesClassifier because after hyper-parameter tuning these algorithms gave good results. The details about results and parameters are explained in detail in the later sections.

2.2 Technologies,Tools and Libraries Used

1. Pandas

Pandas was used to handle dataset, read data, to create a dataframe, and perform operations on the dataframe (like cleaning the data)etc.

2. Python 3
3. Jupyter Notebooks
4. Scikit-learn

Various Scikit-learn libraries were used to perform classification and regression on the data. Example DecisionTreeClassifier, GaussianNB, KNeighborsClassifier, LogisticRegression.

5. Seaborn

It was used to create a visual analysis of the data, for comparison of the models and understanding the data during exploration.

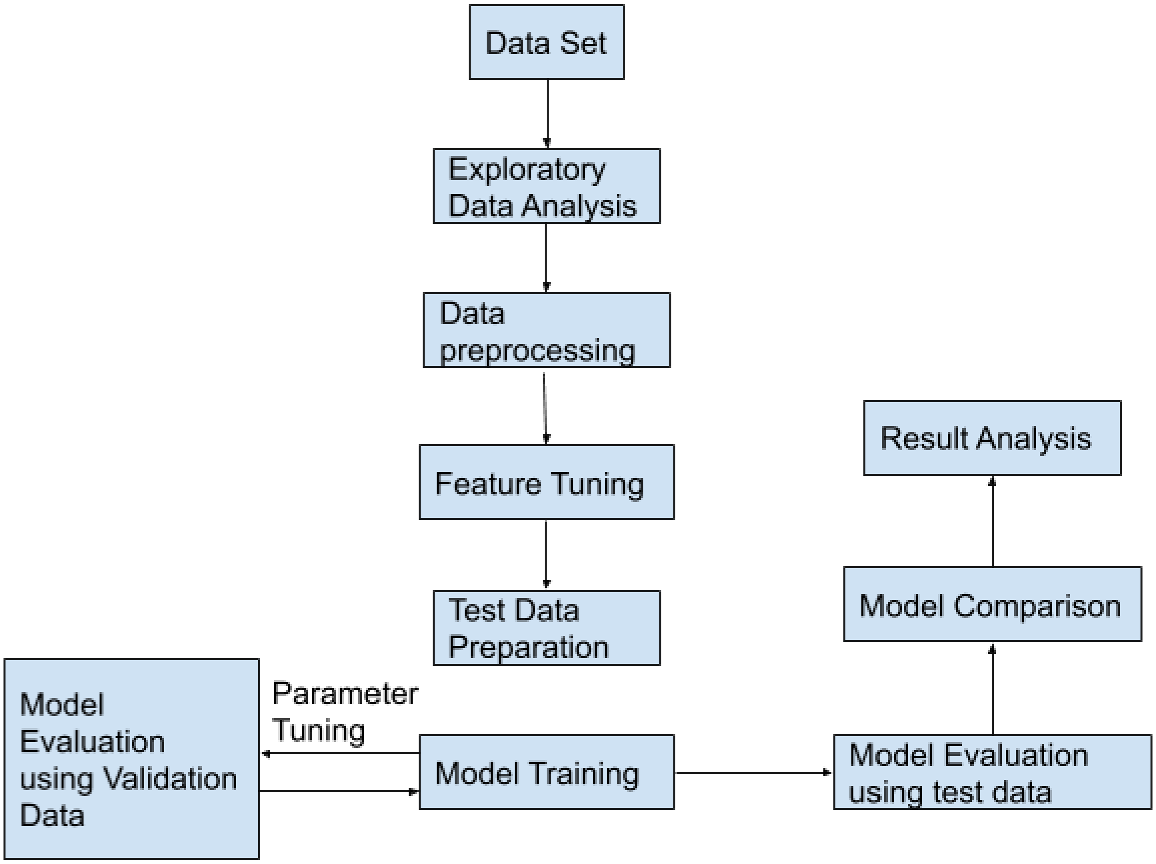
6. Matplot Library

It was used to create a visual analysis of the data, for comparison of the models and understanding the data during exploration.

7. Imblearn

Various Imblearn functions like random over sampling, SMOTE were experimented with to balance the imbalance dataset.

2.3 System Design and Architecture



2.4 Use Cases

With this project we will be providing a solution which can be utilized for analysing the income of the majority of the citizens of the country or any subset of the same. So the government or economists can use the results to analyse the economic situation of the targeted group before taking any measures to provide economic support.

It can also be used by the company for targeted advertisements to sell their products to a particular class of people. Or understand if the product can have success in the current economic situation of the country.

3. Experiments

3.1 DataSet

The dataset used for the project is **Census-Income (KDD) Data Set** and is collected from UCI machine learning repository. The dataset contains separate files for training and testing. The training dataset contains 199523 records with 42 attributes and the test dataset contains 99762
records. The dataset contains continuous and ordinal attributes.

3.1.1 Description

The dataset contains 3 csv files. The census-income.data and census-income.test contains the train and test records respectively. The census-income.names contains descriptions about features in the dataset.

3.2 Data Preprocessing

3.2.1 Feature Selection

Based on the correlation between columns in the train dataset some features are removed and the total number of unique values in the columns are calculated and redundant columns were removed. Some columns were removed based on intuition like native country of mother and native country of father were removed because native country of self column is already giving enough information.

3.2.2 Exploratory Data Analysis

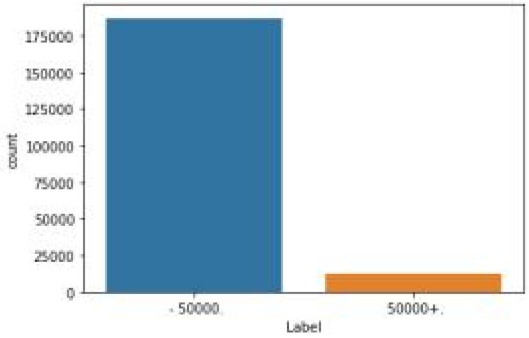


Fig1.Data distribution of income in trains dataset

Data analysis on census data provided some interesting results. The first important observation is the dataset is heavily imbalanced with records having income more than 50,000 being 12382 and records with income under 50000 being 187141.

The proportion of missing values(3393) to the total dataset(199523) is small, so the missing value rows are removed from the dataset. After feature selection, the dataset contains various categorical columns, the categorical valued columns are aggregated into categories so that the number of one-hot encoded labels for each column is limited.

3.2.3.Data Cleaning

The categorical column 'class of worker' contains 9 unique values, there are grouped into 5 unique values aggregating 'state government', 'local government', 'federal government' into government. The 'self-employed no incorporated' and 'self employed incorporated' are aggregated into 'self-employed' category. Similarly the 'never worked' and ' without pay' are aggregated into the 'no pay' category.

- The categorical column 'education' contains 17 distinct values, these are grouped into 7 distinct values by grouping 'children' and all grade values into 'No school'. And also aggregating 'Some college but no degree' ,'Associates degree-occup /vocational','Associates degree-academic program' into 'College'
- The categorical column 'country of birth self' is grouped into values of continents as the country values are 43 unique values.
- The 'marital status' column is aggregated into 'Married', 'Not married', 'Separated' by aggregating 'Married-civilian spouse present','Married-spouse absent', 'Married-spouse present' into 'Married' and 'Divorced' ,'Separated' into 'Separated'.
- The labels column is label encoded to 0 and 1 with 0 being income less than 50000 and 1 being income greater than 50000

3.4 Methodology

- The data has now been cleaned and feature selection completed, However as we see that the data is highly imbalanced, and using the dataset as it is would give us a poor performance on the minority class. The approach to address imbalanced datasets is to oversample the minority class. New examples can be synthesized from the existing examples. This type of data augmentation called SMOTE is used for balancing out the dataset in our case. The size of the dataset before SMOTE: (196130, 12) After SMOTE the dataset was balanced with dimension: (368090, 12).

- Since there is no one perfect classification algorithm that can be used, multiple models were created and trained on the balanced dataset and their accuracies were compared. The following classification models were created and the F1 macro score calculated on the test data:

Models	F1 Score - macro	F1 Score - micro	F1 Score - weighted
Decision Tree Classifier	0.690407726	0.924139452	0.926048523
Gaussian Naïve Bayes	0.673021838	0.888965738	0.905791533
K Nearest Neighbours	0.629956875	0.842735711	0.87575976
Logistic Regression	0.607996159	0.79784888	0.846969891
Random Forest	0.716113242	0.947946112	0.944408608
Linear discriminant analysis	0.621228761	0.807271306	0.853765311
Extra Trees Classifier	0.737802822	0.942402919	0.940694165

Table 1: Models vs F1 scores obtained

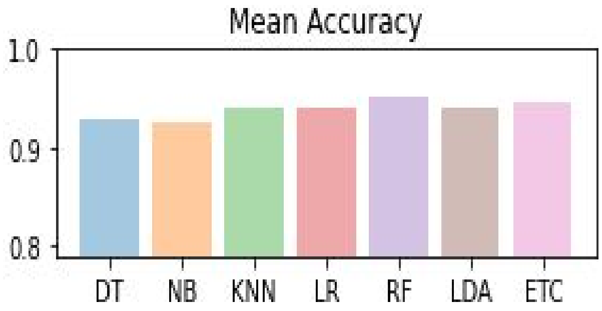


Fig2.Mean Accuracy of models created

- The top 3 values for F1 Macro score were obtained by classifiers Decision Tree, Random Forest, and Extra Trees Classifier. Therefore hyperparameters are implemented by creating pipelines with multiple possibilities of parameters to test for the highest possibilities of F1 Score and obtain the best possible results.
- On obtaining the hyperparameters for these 3 models we rerun the models on the test dataset and got the ideal results and F1 score.

3.5 Results

Graphs showing different parameters/algorithms evaluated in a comparative manner, along with some supportive text. (as applicable)

• Analysis of results :
On running the 3 models Decision Tree, Random Forest, and Extra Trees Classifier. With the calculated hyperparameters, we get the following results.

RandomForestTree Classifier:
F1 macro Score - 0.713922517516442
DecisionTree Classifier:
F1 macro Score - 0.716498549452453
ExtraTrees Classifier:
F1 macro Score - 0.7451752909381619

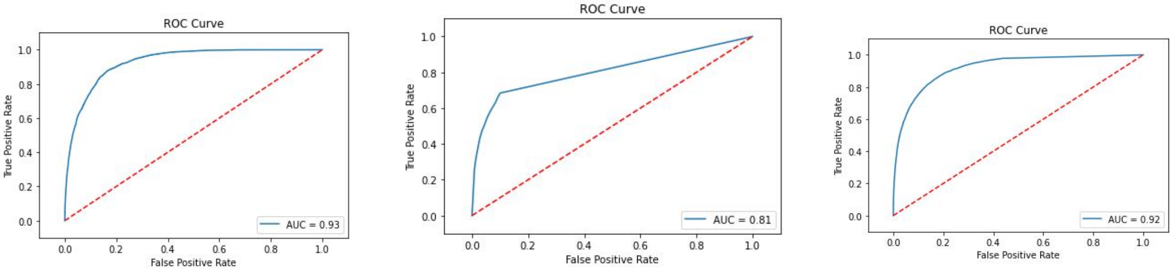


Fig3.ROC Curves for 3 models created using hyperparameters

Based on the above result we conclude that the ExtraTrees Classifier provides the highest value for F1 score with the following parameters:
model = ExtraTreesClassifier(class_weight='balanced_subsample', max_features=None,min_samples_split=5, n_estimators=400, random_state=5)

4. Discussion & Conclusions

4.1 Difficulties Faced and Decisions Made

- Initially the number of features in the train and test dataset was 42 which needed to be reduced to both reduce the computational cost of modeling and to improve the performance of the model. To improve this we performed feature selection and reduced the features to 12.

- Another challenge faced during the process was that the data was highly imbalanced, and using the dataset as it is would give us a poor performance on the minority class. To overcome this challenge we performed SMOTE to create a balanced dataset.
- Model Selection - Finding the single classification model with appropriate parameters was another challenge for which we ran multiple models and then created a pipeline to test hyperparameters for the top 3 models .

4.2 Things that worked

Using hyperparameters and SMOTE helped increase the models accuracy substantially.

4.3 Things that didn’t work well

We attempted PCA to reduce dimensions however the dimensions this did not give us an effective result

4.4 Conclusion

We conclude that the ExtraTrees Classifier provides the best result (F1 macro score = 0.74) with the following parameters:

model = ExtraTreesClassifier(class_weight='balanced_subsample', max_features=None,min_samples_split=5, n_estimators=400, random_state=5)

5. Project Task Distribution

Task	**Responsibility**
Data Collection	All
Data Pre-processing	All
Exploratory Data Analysis	All
Data Cleaning	All
Research on models	All
Decision Trees, Naive Bayes	Bhavana

Random Forest, KNN	Raaga Pranitha
Extra Trees Classifier, Logistic Regression	Namratha
Project Report	All
PPT	All

6. References

1. https://pandas.pydata.org/docs/
2. https://scikit-learn.org/stable/modules/neighbors.html
3. https://scikit-learn.org/stable/modules/naive_bayes.html
4. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
5. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html
6. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
7. **Source Code:** https://github.com/raagapranitha/cmpe255-project