Predicting Income using Demographic Information

Rohit Barve
Khoury college of computer sciences,
Northeastern University,
Boston, MA.
barve.r@northeastern.edu

Vinitha Joyce Marathi
Khoury college of computer sciences,
Northeastern University,
Boston, MA.
marathi.v@northeastern.edu

Lasya Manthripragada
Khoury college of computer sciences,
Northeastern University,
Boston, MA.
manthripragda.l@northeastern.edu

1. Abstract

The main aim of the project is to find the best fit model to predict whether an individual is earning more than 50K or less than 50K per year. The prediction is based on the "Adult dataset" taken from the UCI machine learning repository. This project is modeled as a classification problem, hence, to evaluate the best model, metrics like F1-score, and Area under the receiver operating characteristics curve are used. The dataset contains information like individual's age, education, occupation, gender, race etc., The approach includes data-preprocessing (i.e.,) dealing with null values and feature selection.

2. Statement of contribution

- Rohit Barve: Performed EDA,cleaned data and built models SVM (linear and RBF) ; also optimized the results. Prepared presentation and report.
- Vinitha Joyce Marathi: EDA, cleaned data and built models Gaussian Naïve Bayes; also optimized the results. Prepared presentation and report.
- Lasya Manthripragada: Performed EDA,cleaned data and built models Logistic regression, and Random forest; also optimized the results. Prepared presentation and report.

3. Introduction

Our dataset [1] consists of 14 features and one output, which is the prediction of income if it exceeds 50K or not. If income does exceed 50K it is considered a positive class, and negative class otherwise. The 14 input variables which are affecting our target variable (dependent variable) are considered independent variables. The 14 input variables are as follows (It includes the target variable as well- income):

- Age: Age of the individual, it is an integer.
- work_class : It represents the employment status of the individual.
    - Private, Selfempnotinc, Selfempinc, Federalgov, Localgov, Stategov, Withoutpay, Never-worked
- final_weight : Sampling weights
- education: The highest level of education of the individual.
- edu_num: No. of years of education the individual had.
- marital_status: Marital status of the individual. Married-civ (married a civilian), married-AF (married a spouse in armed forces)
    - Marriedcivspouse, Divorced, Nevermarried, Separated, Widowed, Marriedspouseabsent, MarriedAFspouse.
- Occupation: Occupation of the individual.

    - Techsupport, Craftrepair, Otherservice, Sales, Execmanagerial, Profspecialty, Handlers-cleaners, Machineopinspct, Admclerical, Farmingfishing, Transportmoving, Privhouseserv, Protectiveserv, ArmedForces.
- Relationship: Represents the relationship of the individual to others.
    - Wife, Ownchild, Husband, Notinfamily, Otherrelative, Unmarried.
- Race: Represents race of the individual.
    - White, AsianPacIslander, AmerIndianEskimo, Other, Black.
- Gender: Gender of the individual.
    - Female, male
- cap_gain: Captial gain of the individual.
- cap_loss: Capital loss of the individual.
- work_hours: No. of hours the individual works per week.
- Country: Country of origin of the individual.
    - UnitedStates, Cambodia, England, PuertoRico, Canada, Germany, OutlyingUS(GuamUSVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, DominicanRepublic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, ElSalvador, Trinadad&Tobago, Peru, Hong, HolandNetherlands.
- Income: Label, whether an individual earns more than $50,000 dollars annually.
    - >50K (1) , <=50K (0)

The dataset contains 32560 entries with 15 variables as discusses above. The dataset contains both numerical and categorical data. Features- Native country, occupation, and workclass contain null values. (Missing values handling will be discussed more elaborately in section 4) Since, the target variable column is a string we converted into binary, income more than 50K belongs to class 1, class 0 otherwise.

To gain more insights on features and their correlation between the input variable and target, we visualized the data. We plotted a bar graph, by considering only categorical data, against the mean of the income column. By these graphs we could deduce some correlation between them.

Figure 3.1 represents occupation. The bar graph suggests that individuals who come under the category 5 and 7 (i.e.,) prof-speciality and exec-managerial are more likely to have an income which exceeds 50K.

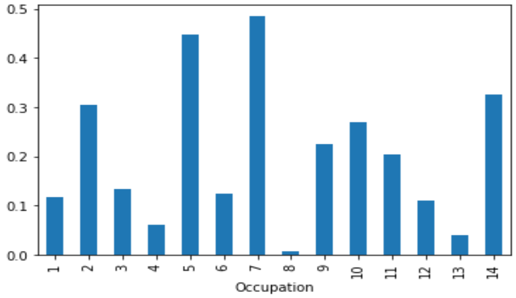


Figure 3.1 Occupation vs Income

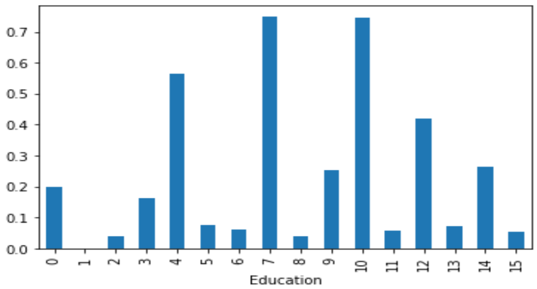


Figure 3.2 Education vs Income

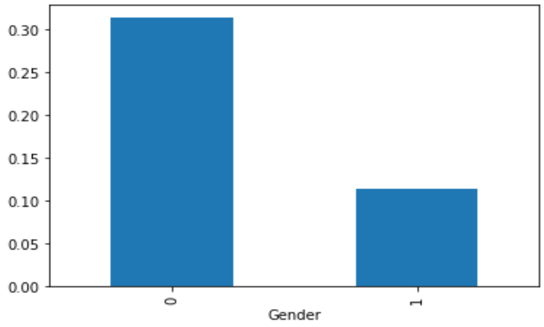


Figure 3.3 Gender vs Income

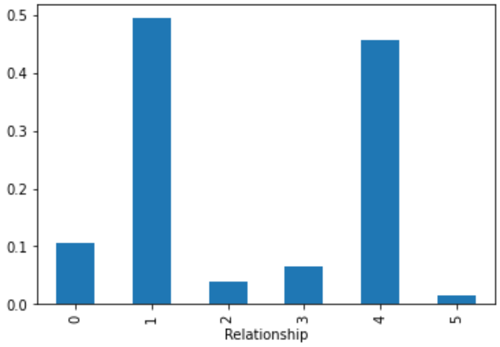


Figure 3.4 Relationship vs Income

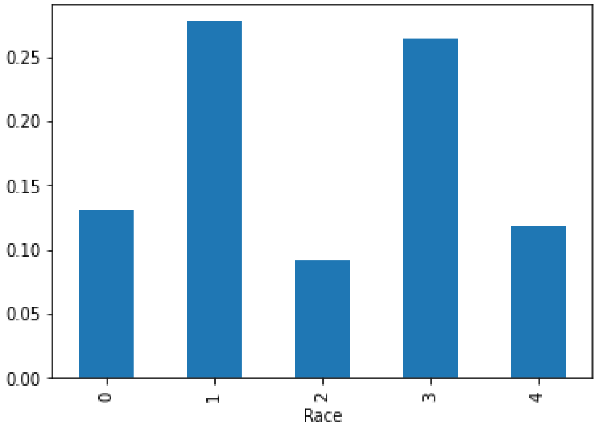


Figure 3.5 Race vs Income

According to figure-3.2, individuals who come under the category 7 and 10 (i.e.,) individuals who have a prof-education, and a doctorate are more likely to earn a higher income. In same fashion, figure 3.3, represents gender, which significantly implies that males have a higher chance of having a higher income. Figure 3.4 represents the relation of the individual relative to others vs the individuals' income, and it implies that individuals who come under 1 and 4 are more likely to have a higher income than other individuals. Essentially, a married couple has a better chance of having an income which exceeds 50K. According to figure 3.5, individuals who are white or Asian-pac-Islander has a better chance of earning a higher income.

There are entries that has null values, and all the input variables are not necessarily correlated to target variable. Section 4 discusses data preprocessing which includes handling null data and feature selection.

4. Feature extraction and data preprocessing

In the exploratory data analysis part, patterns in the dataset were studied with the help of graphs to get an intuitive understanding of the relationship between the target variable and the input features.

As seen in the EDA, some features such as gender, marital status gave a very strong correlation with the target variable. In the pre-processing stage, the main objective was to clean the dataset so that it does not contain any redundant features or any features with a significant number of null values.

The steps involved in pre-processing were:

- Dropping the observations which had null values or the ’?’ character.
- Columns like capital gain and loss were dropped since more than 90% of the observation values in them were zero and did not have any significant effect in predicting the income.
- The target variable i.e. income was mapped to 0 (<=50K) and 1 (>50K).
- Continuous features such as age, work hours etc. were normalized.
- Ordinal features like education were mapped in accordance to their order and higher weights were assigned if the observation had higher education level.
- Nominal features like marital status, occupation were one hot encoded so that each unique entry will have an impact on the target variable.

In the feature selection part, correlation between features and target variable was calculated and the features with very low correlation were dropped. So, features like age, work hours, final weight were dropped. For the countries feature, only the countries with more than 100 observations were selected.Also, features with a uniform distribution with respect to target variable were dropped.Finally for predicting the income, the data comprised of 57 features and 16K observations.

5. Models

Given attributes like Age, work_class, education, etc we are finding if income is above 50k or not. The models we are using will produce a binary result 1(if income greater than 50k) or 0 (if income lesser than 50k).

The models we chose to implement are:

1. Gaussian Naïve Bayes
2. Logistic Regression
3. Random Forest
4. Support Vector Machine

5.1 Gaussian Naïve Bayes

This is a statistical classification technique focusing on determining the probabilities of the features belonging to a particular class. This is done by using the Bayes (conditional) theorem as given below:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

“This method assumes that all the features of a given dataset are independent of each other and hence their joint probabilities factor into individual probabilities. The prior probabilities i.e., the probability of occurrence of each class is obtained by using the number of data points belonging to a particular class divided by the total number of data points. This method is known to suit high dimensional data and has a better performance when the independence condition holds true.”[5]

5.2 Logistic regression

It is one of the easiest, and straight forward algorithms in machine learning. It also works as a baseline for other algorithms. We used scikit [2] to implement logistic regression. The fundamentals of the algorithm are easy to very easy to understand, and as mentioned before it works as baseline for binary classification problem. It works on sigmoid function, which extents between 0 and 1. If the value goes positive infinity, the model predicts positive class, and it goes negative infinity it predicts the negative class. Figure 5.2.1 is the sigmoid function.[8]

$$f(x) = \frac{1}{1+\mathrm{e}^{-(x)}}$$

Figure 5.2.1 Sigmoid function

5.3 Random forest

Random forest is used for both classification and regression. It is a type of bagging ensemble learning model; it chooses the majority vote from multiple decision trees to predict. Random forest does well with high dimensionality data because it works with subsets of data. It is flexible and over comes the problem of overfitting by combining the results of multiple decision trees.[4] Figure 5.3.1 demonstrates how a random forest works [7]

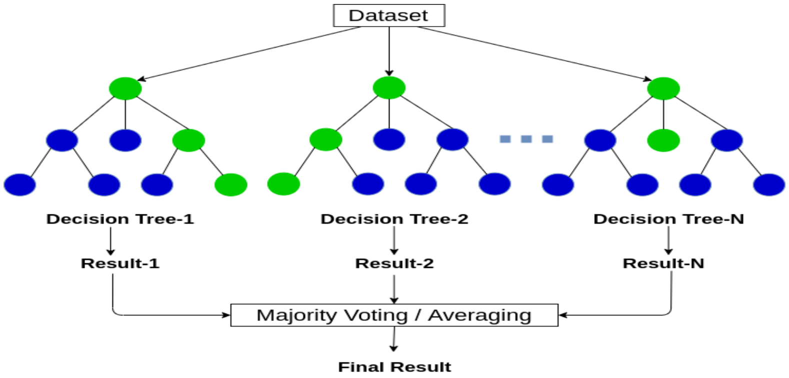


Figure 5.3.1 Understanding of random forest

5.4 SVM

Support vector machines are a type of supervised machine learning algorithms that can perform both classification and regression tasks. They work quite well in a very high dimensional feature space and employ use of kernels like linear kernel, RBF (Radial Basis Function) kernel to separate the data. In some cases for separating data points cannot be separated in that feature space and the feature space has to be expanded for prediction. But the exact nature of the space that can predict the decision boundary is unknown. Kernels provide an optimum solution to this problem in that they represent the expanded feature space as a function of inner product of input features which is much more computationally efficient.

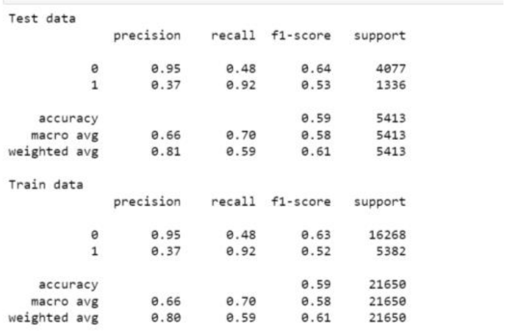
For predicting income, the SVC function of the sklearn library was used. It takes in different parameters like kernel function, C (regularization parameter). The parameter ‘C’ is used for implementing soft-margin SVM if C > 0. The higher the value of ‘C’, the more incorrect datapoints near the decision boundary are accepted.

6. Experiments and Evaluation

6.1 Gaussian Naïve Bayes

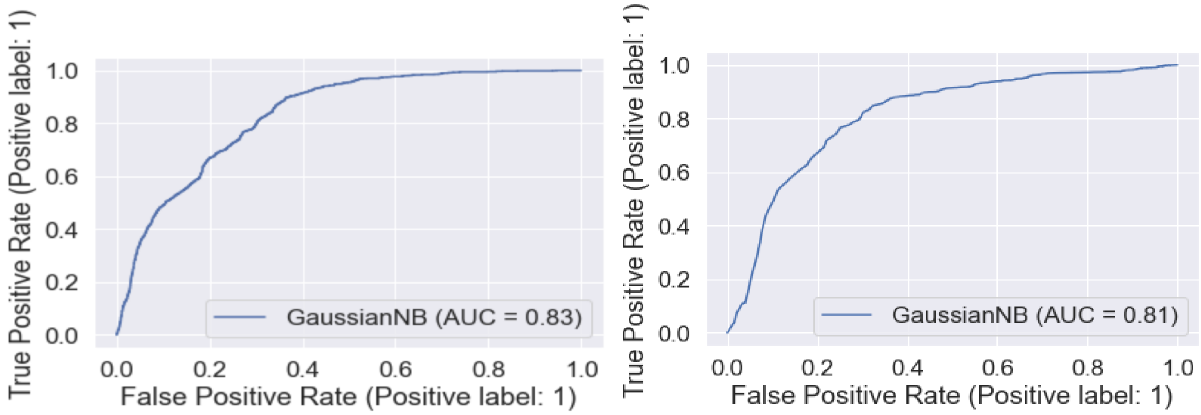
We used scikit to evaluate the model. This function assumes normal distribution for class conditional densitites.

Function used: GaussianNB from sklearn.naive_bayes [5]



```
Test data
              precision    recall  f1-score   support

           0       0.95      0.48      0.64      4077
           1       0.37      0.92      0.53      1336

    accuracy                           0.59      5413
   macro avg       0.66      0.70      0.58      5413
weighted avg       0.81      0.59      0.61      5413

Train data
              precision    recall  f1-score   support

           0       0.95      0.48      0.63     16268
           1       0.37      0.92      0.52      5382

    accuracy                           0.59     21650
   macro avg       0.66      0.70      0.58     21650
weighted avg       0.80      0.59      0.61     21650
```

Figure 6.1.1 Classification matrix for Guassian NB



Fig

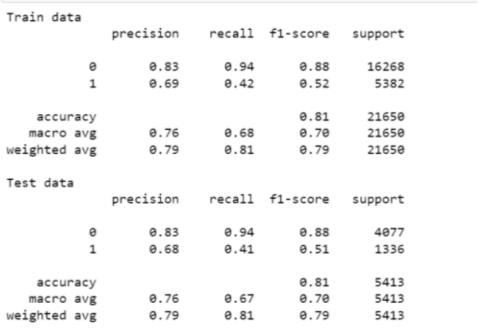
Figure 6.1.2- 6.1.3 AUC curve for test and train data

The accuracy given by Naïve Bayes Algorithm after necessary data cleaning and pre-processing is about 58.5% for train data and 59% for test data which is significantly lower than the other implemented models. The AUC score calculated for the test data is 0.83 and 0.81 for train data. "As expected, the Naive Bayes classifier does not perform well due to the assumption made on class conditional densities. However, they are simple to implement and serve as a good baseline for the other models." [6]

6.2 Logistic Regression

While data pre-processing, as mentioned in section 4, we used one-hot encoding for the categorical data. However, while experimenting, we all also tried changing the feature "NativeCountry" to a binary feature. The reason being, 86% of the data (i.e.,) more than half of the individuals' country of origin is United stated of America. We divided the individuals into two categories- USA (1) and non-USA (0). While training the data it did not give very significant results. The accuracy it gave was 76%. Since it did not do well, we also discarded the column to see how it affects the training of model. Any of these attempts did not seem to go well. The accuracy dropped from the previous attempt and gave 74% accuracy.

However, by implementing the data with the data pre-processing and feature selection discussed in section 4, the model has achieved an accuracy of 80.6% on the test data and 80.7% accuracy on train data.



```
Train data
              precision    recall  f1-score   support

           0       0.83      0.94      0.88     16268
           1       0.69      0.42      0.52      5382

    accuracy                           0.81     21650
   macro avg       0.76      0.68      0.70     21650
weighted avg       0.79      0.81      0.79     21650

Test data
              precision    recall  f1-score   support

           0       0.83      0.94      0.88      4077
           1       0.68      0.41      0.51      1336

    accuracy                           0.81      5413
   macro avg       0.76      0.67      0.70      5413
weighted avg       0.79      0.81      0.79      5413
```

Figure 6.2.1 Classification matrix for test and train data

The performance did not necessarily stand out, but it is a good model to predict the income of the individual. The AUC score calculated to be 0.85 for train data, and 0.84 for test data. The roc_auc score was calculated by using decision_function which estimates how far a datapoint is away from the actual decision boundary.

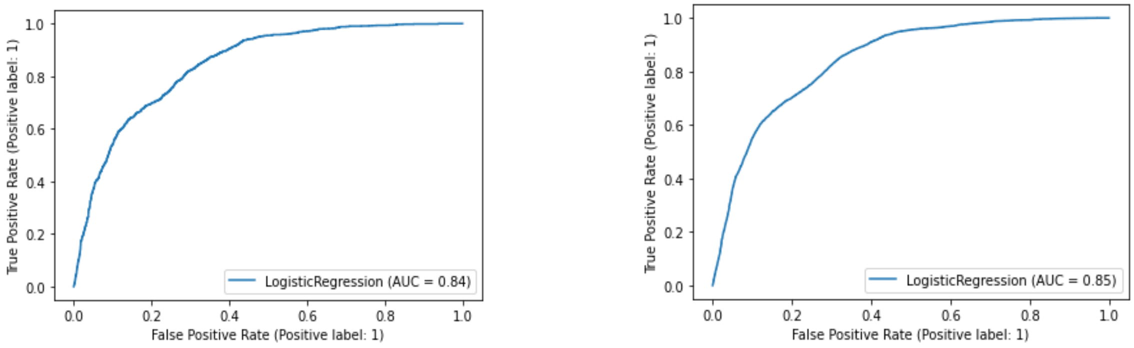


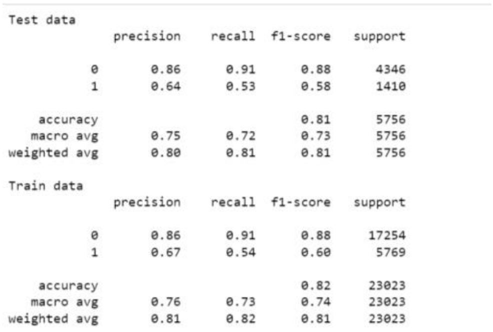
Figure 6.2.2-6.2.3 AUC curve for test and train data

6.3 Random Forest

We used scikit [3] to implement the algorithm. It has important parameters like max_depth, max_features, n_estimators, and min_sample_leaf. The no.of trees used to build the model is given by n_estimators, and the dept the of the tree is given by max_depth.

Max_features determine the no.of features an individual tree can consider.The no.of minimum samples a leaf node can hold is given by min_sample_leaf.

The accuracy of the train and test dataset after implementing random forest calculated to be 81.8% and 81.3%.



```
Test data
              precision    recall  f1-score   support

           0       0.86      0.91      0.88      4346
           1       0.64      0.53      0.58      1410

    accuracy                           0.81      5756
   macro avg       0.75      0.72      0.73      5756
weighted avg       0.80      0.81      0.81      5756

Train data
              precision    recall  f1-score   support

           0       0.86      0.91      0.88     17254
           1       0.67      0.54      0.60      5769

    accuracy                           0.82     23023
   macro avg       0.76      0.73      0.74     23023
weighted avg       0.81      0.82      0.81     23023
```

Figure 6.3.1 Classification matrix of train and test data

The optimal no.of estimators is 200 for this dataset. While we experimented with other values of parameters, when the value was more than 500 towards 1000, the model started to overfit the data, and when it was lesser than 200, the model started to underfit the data.

We have used the following parameters for our model:

RandomForestClassifier(criterion='entropy', max_depth=20, min_samples_split=50, n_estimators=200, oob_score=True, random_state=42

The AUC curve is given below in figure 6.3.2 and 6.3.3 for train and test dataset:

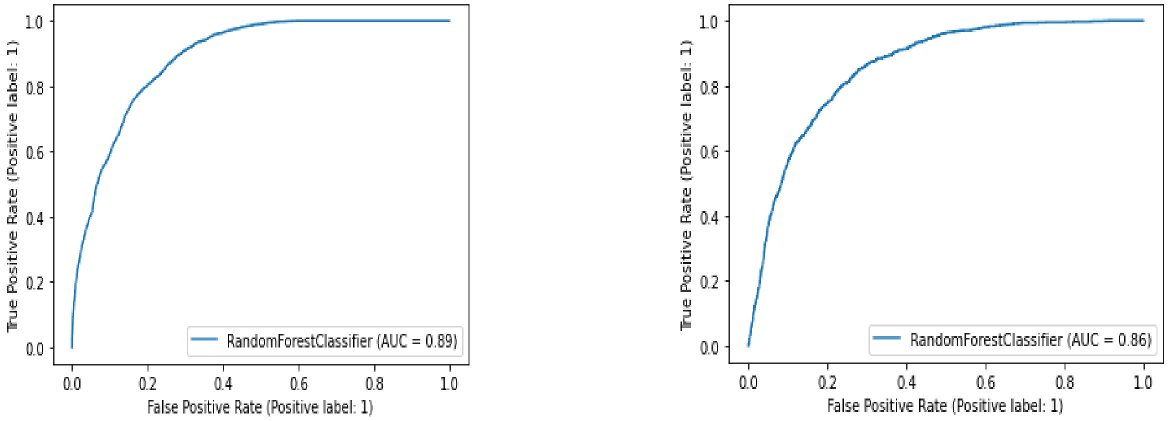


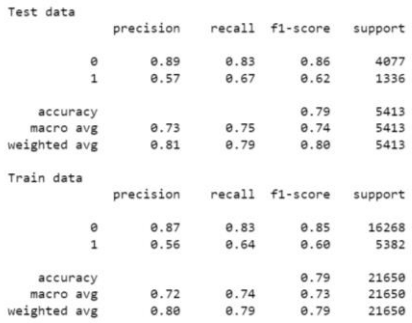
Figure 6.3.2-6.3.3 AUC for train and test dataset

6.4 SVM

Different values for regularization parameter were tested and different types of kernels were used. The values for ‘C’ were 0.1, 1 (default), 10, 100. The higher values allowed incorrect datapoints while the lower values punished the model for including wrong datapoints.

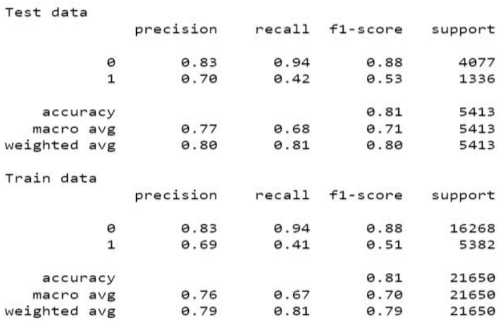
The accuracy for training data with “linear kernel” was 78.5% and for testing data 79.3%. The kernel that seemed to give the best results was the ‘RBF’ kernel since it can generalize the function to an infinite feature space.

The accuracy for training data was 80.6% with an ‘RBF’ kernel and 81.3% for testing data. This indicates that there is very low variance between the training and testing results. AUC score was 0.82 for training data an 81% for testing data.



```
Test data
              precision    recall  f1-score   support

           0       0.89      0.83      0.86      4077
           1       0.57      0.67      0.62      1336

    accuracy                           0.79      5413
   macro avg       0.73      0.75      0.74      5413
weighted avg       0.81      0.79      0.80      5413

Train data
              precision    recall  f1-score   support

           0       0.87      0.83      0.85     16268
           1       0.56      0.64      0.60      5382

    accuracy                           0.79     21650
   macro avg       0.72      0.74      0.73     21650
weighted avg       0.80      0.79      0.79     21650
```

Figure 6.4.1 Classification matrix for SVM Linear kernel



```
Test data
              precision    recall  f1-score   support

           0       0.83      0.94      0.88      4077
           1       0.70      0.42      0.53      1336

    accuracy                           0.81      5413
   macro avg       0.77      0.68      0.71      5413
weighted avg       0.80      0.81      0.80      5413

Train data
              precision    recall  f1-score   support

           0       0.83      0.94      0.88     16268
           1       0.69      0.41      0.51      5382

    accuracy                           0.81     21650
   macro avg       0.76      0.67      0.70     21650
weighted avg       0.79      0.81      0.79     21650
```

Figure 6.4.2 Classification matrix for SVM RBF kernel

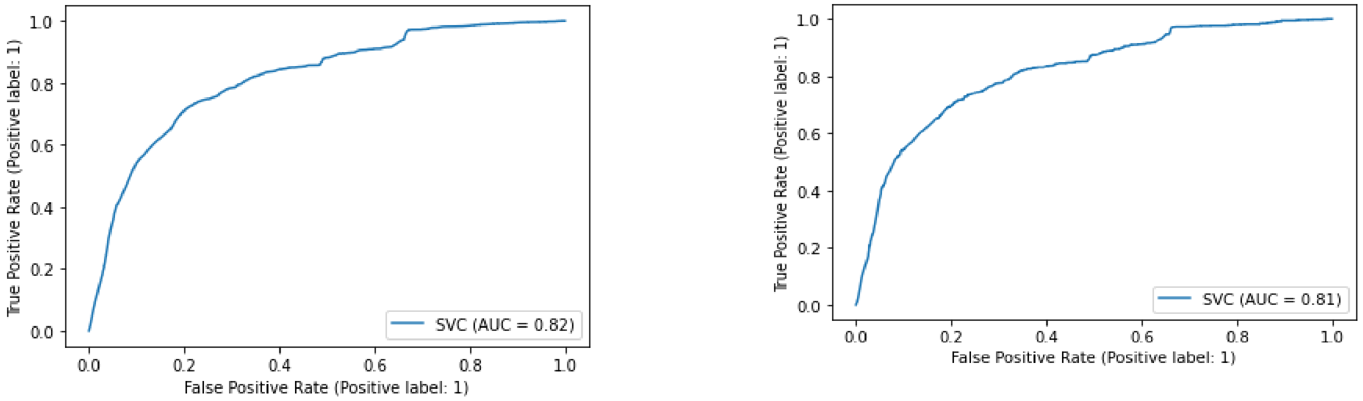


Figure 6.4.3-6.4.4 AUC for train and test data

7. Results

Classifier	Train accuracy	Test accuracy	Precision	Recall	F1-score	ROC_AUC Score
Gaussian NB	0.585	0.59	0.924	0.68	0.78	0.83
Logistic Regression	0.807	0.806	0.827	0.93	0.875	0.84
Random Forest	0.818	0.813	0.85	0.904	0.876	0.86
SVM (RBF)	0.806	0.813	0.87	0.89	0.88	0.82

Table 7.1 Results for different models used.

By evaluating the results, the training data and testing data are in the same range which indicates that there is no overfitting. Naïve Bayes did not perform as suspected in section 5, it doesn't do well with imbalanced data. Logistic regression, Random Forest, and SVM RDF kernel performed well on the dataset.

8. Conclusion and Future Scope

After the necessary data pre-processing, and feature selection; input variables which had a little or no correlation with the target variable were dropped, and entries with null values were discarded as well. After implementing our models to fit the data, Gaussian NB did not perform well. However, SVM, Logistic regression, and Random Forest performed moderately well, but they performed significantly better than Gaussian NB. By evaluating the results using metrics ROC_AUC score, random forest is the best model to predict income for an individual. Adult dataset is highly imbalanced. For example, 'Native country' has 86% of USA for the individuals, which implies that there is not enough data from other countries. For future developments, more accurate, and well-structured data could be gathered. Since it has many dimensions, we can use PCA for dimensionality reduction.

9. References

[1] https://archive.ics.uci.edu/ml/datasets/census+income

[2] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[4] https://medium.com/analytics-vidhya/tuning-random-forest-algorithm-to-predict-inco

[5] https://scikit-learn.org/stable/modules/naive_bayes.html

[6] https://github.com/VaishnaviKrishna/adult-dataset

[7] https://ai-pool.com/a/s/random-forests-understanding

[8] https://medium.com/@gabriel.mayers/sigmoid-function-explained-in-less-than-5-minutes-ca156eb3049a