Prediction of the salary class of a person from the census data

Marketing of consumer goods requires information like age, location, spending power etc. Goods like cars might be purchased by people above a certain salary class. So, marketing of cars can be targeted only towards the class of people who can afford it. Filtering the population based on salary class is helpful to marketing campaigns for signing up to educational programs, selling of automobiles, to find clients for banking products like term deposits, credit cards etc. With the given set of attributes from the census data, which are Age, Work class, Final weight, Education, Education-num, Marital-status, Occupation, Relationship, Race, Sex, Capital-gain, Capital-loss, Hours-per-week, Native-country, we have to predict if the person's salary class belong to >50k per year class or <=50k per year class.

Target Audience

This analysis will be useful to marketing companies to segregate the people based on the class and target certain campaigns to certain people.

The dataset being used consists of 15 attributes which are Age, Workclass, Finalweight, Education, Education-num, Marital-status, Occupation, Relationship, Race, Sex, Capital-gain, Capital-loss, Hours-per-week, Native-country, Class. It contains continuous and nominal attributes. It has 48842 instances of data with some missing values.

Dataset is donated by

Donor:

Ronny Kohavi and Barry Becker
Data Mining and Visualization
Silicon Graphics.
e-mail: ronnyk '@' live.com for questions.

The dataset was extracted by Barry Becker in 1994 from census data of the United States. A set of reasonably clean records was extracted using the following conditions: ((AAGE>16) && (AGI>100) && (AFNLWGT>1)&& (HRSWK>0))

Methodology

The target variable, Salary Class, referred to as Class, has label assigned to it in the data. So, this problem falls under Supervised learning. Since the model has to classify the person as earning the salary of, ‘Greater than 50k/year’ or ‘Less than or equal to 50k/year’, this is Classification problem. I will model the data on two algorithms, Support Vector Machine and Random Forest Classifier and compare their performance. The metric being used to measure the performance are Confusion matrix and ROC-AUC.

Deliverables

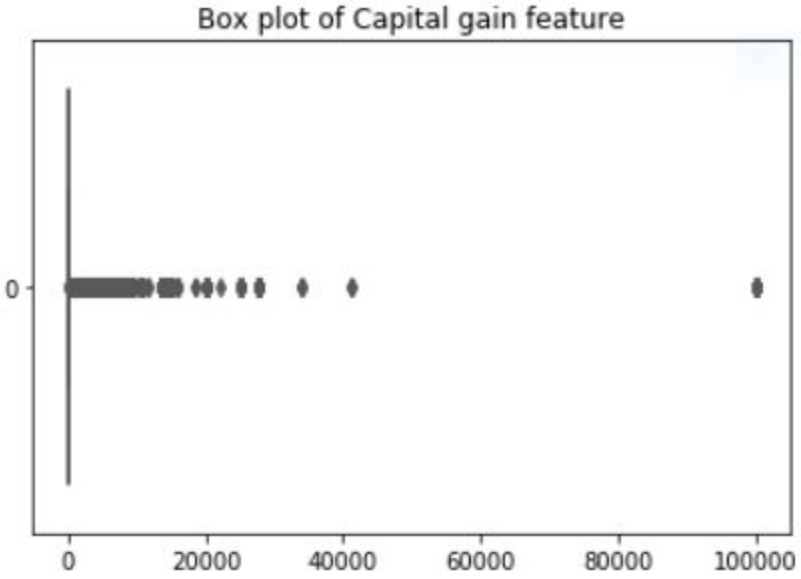
The project report, visualization of the data analysis and the source code will be uploaded on the public github repository.

Data Wrangling

The dataset consists of 48842 rows and 15 columns. Columns are of type categorical and numeric. Categorical columns being 'Workclass', 'Education', 'Marital-status', 'Occupation', ‘Relationship', 'Race', 'Sex', 'Native Country','Class' and numeric column being 'Age', 'Fnlwgt', 'Education-num','Capital-gain', 'Capital-loss' and 'Hours-per-week'.

The dataset is inspected for invalid entries by checking for NANs. There are NAN values in the column Workclass, Occupation, Native Country. There are around 2800 rows NANs out of around 48842 rows, so I am dropping the null values from the dataframe. Trying to forward fill wouldn't make sense for these categorical values. Also, at this point I don’t have much information about the column, ‘Fnlwgt’, so I will not be considering it for this analysis.

The columns, Capital gain and Capital loss have 41432 and 43082 rows of zero value respectively. Since more than 90% of the people have 0 Capital gain and 0 Capital loss, I will not consider those features in this prediction problem.



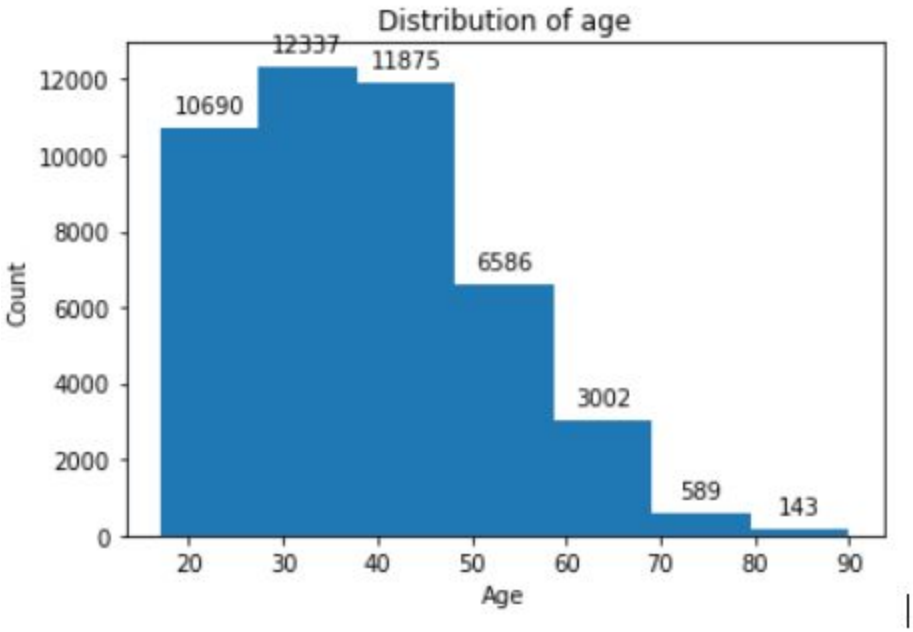
	Age	Education-num	Capital-gain	Capital-loss	Hours-per-week
count	45222.000000	45222.000000	45222.000000	45222.000000	45222.000000
mean	38.547941	10.118460	1101.430344	88.595418	40.938017
std	13.217870	2.552881	7506.430084	404.956092	12.007508
min	17.000000	1.000000	0.000000	0.000000	1.000000
25%	28.000000	9.000000	0.000000	0.000000	40.000000
50%	37.000000	10.000000	0.000000	0.000000	40.000000
75%	47.000000	13.000000	0.000000	0.000000	45.000000
max	90.000000	16.000000	99999.000000	4356.000000	99.000000

Exploratory Data Analysis

We will explore the data to see any patterns, how they are related to other features, check if there are any trends. We will explore the features individually and in relation to another feature.

Age

From the data we find out that,
Oldest person's age in the dataset is 90
Youngest person's age in the dataset is 17

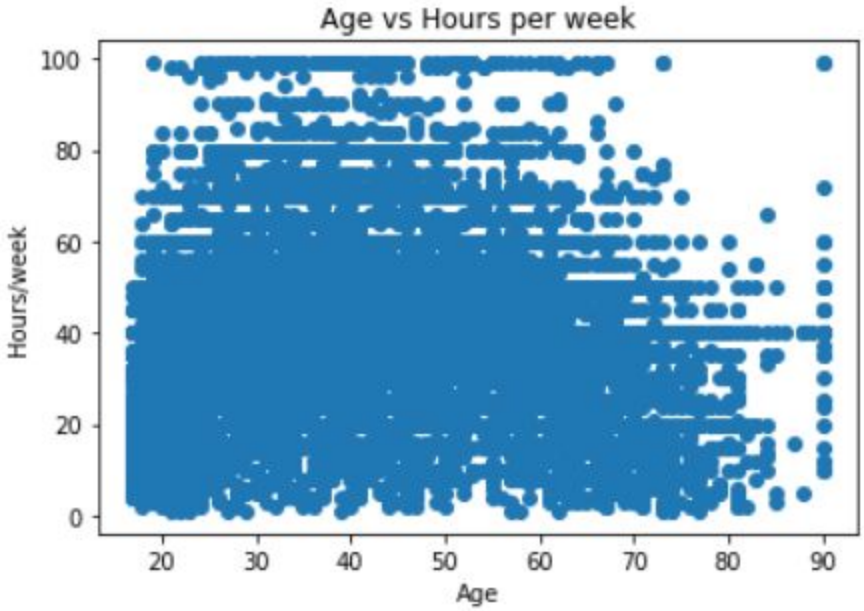


From the plot above, we can see that maximum number of people in the dataset are between 30-40 yrs and least number of people in the group 80-90 years.

Hours per week

Average number of hours worked in the dataset is 40.9 hrs. There are a number of outliers which are number of hours worked less than 32hrs and more than 51hrs which do have predictive power in predicting the income class of the person.

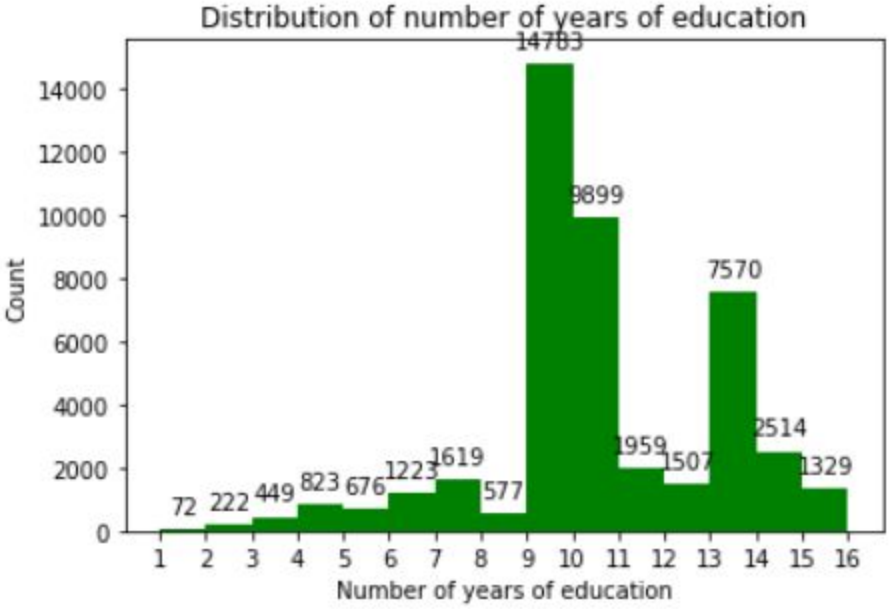
Now, we will see how the person's age affects the number of hours worked per week.



Maximum number of hours per week, which is 90 hours, is worked by people in the age group between 20 and 70. The standard number of hours per week which is 40 h0rs is put in by the

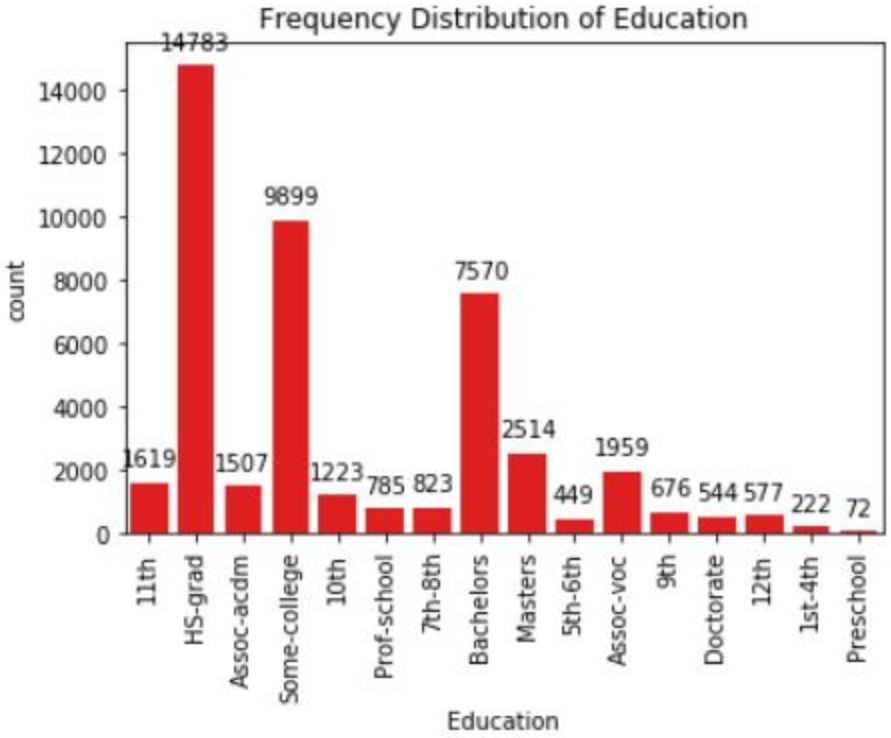
people of all age group in the dataset. The plot also shows that most people are in age group between 20-65 and have worked between 20 - 50 hours per week. More precise numbers can be obtained by analysing the data further.

Number of years of Education



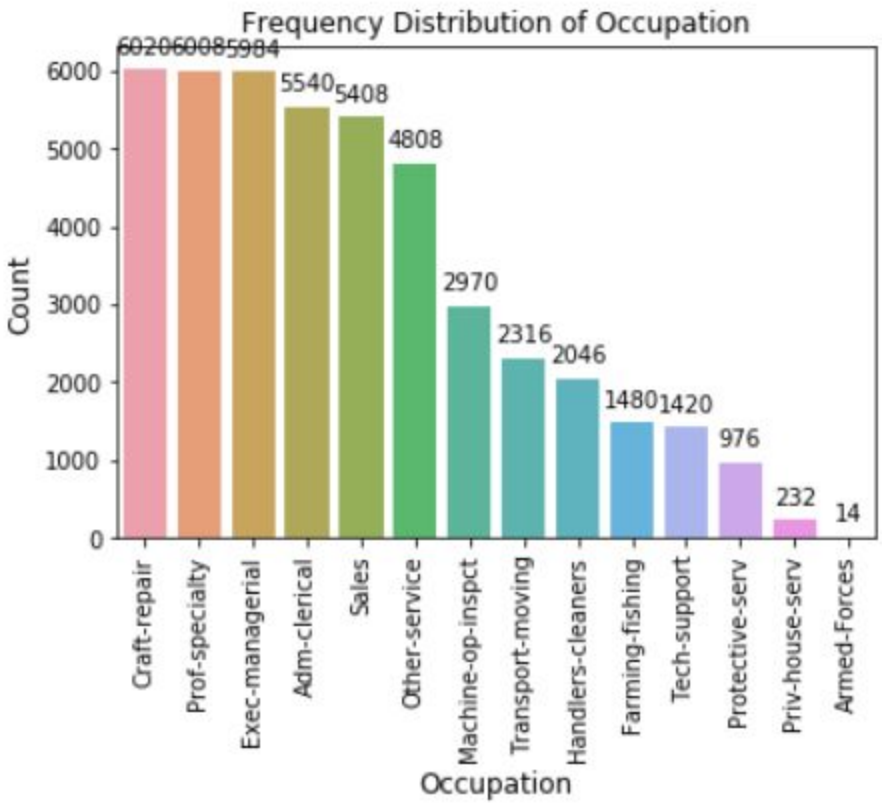
The number of years of education in the dataset ranges between 1 - 16 years. Maximum number of people have 9 years of education.

Education

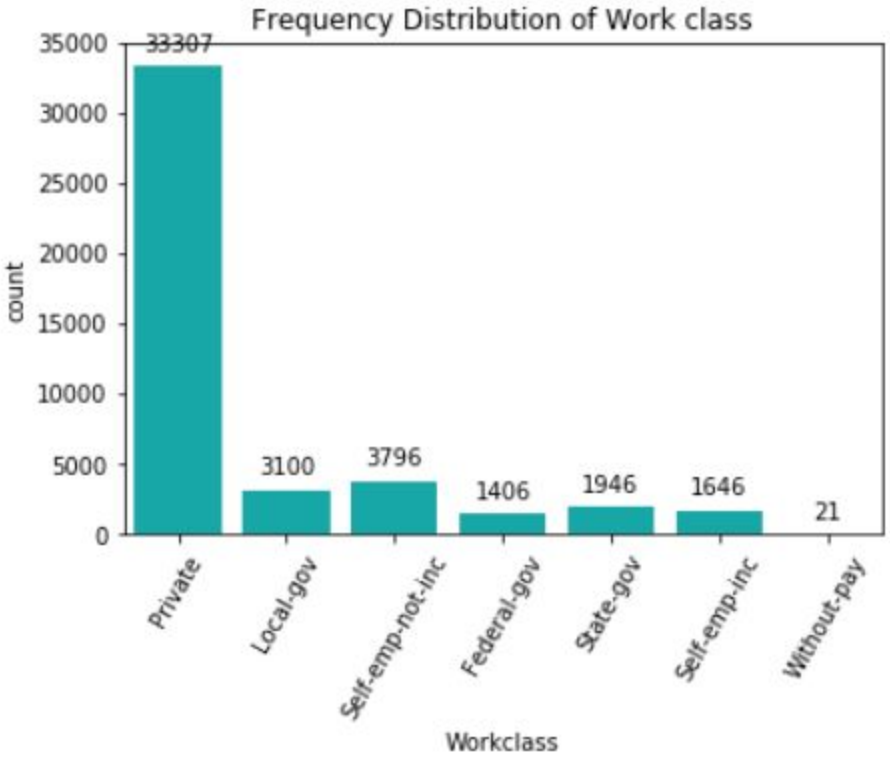


Maximum number of people, 14783, are high school graduates. People with just elementary school education form the minority group.

Occupation

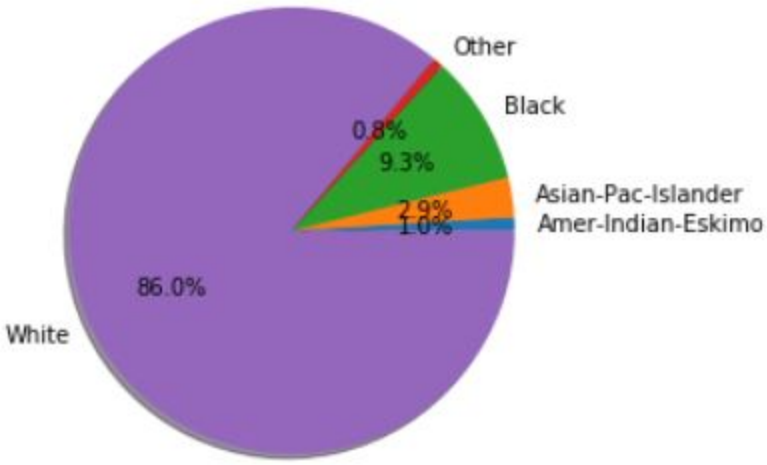

Frequency Distribution of Occupation
6020
6008
5984
5540
5408
4808
2970
2316
2046
1480
1420
976
232
14
Count
0
1000
2000
3000
4000
5000
6000
Craft-repair
Prof-specialty
Exec-managerial
Adm-clerical
Sales
Other-service
Machine-op-inspct
Transport-moving
Handlers-cleaners
Farming-fishing
Tech-support
Protective-serv
Priv-house-serv
Armed-Forces
Occupation

Work class

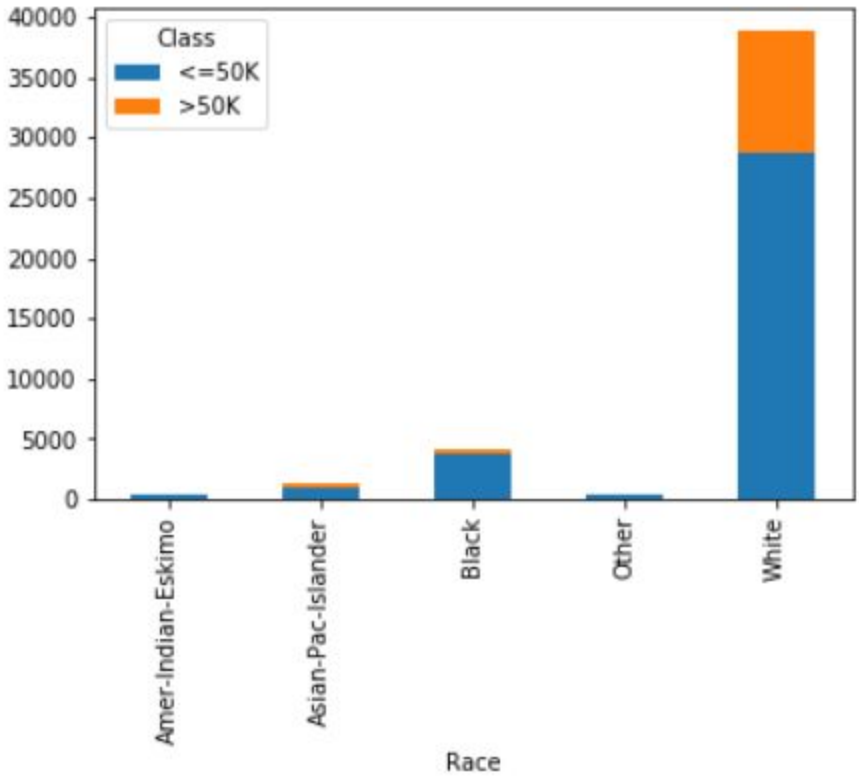


Majority of the given population works for Private sector. There are 21 people without pay.

Race

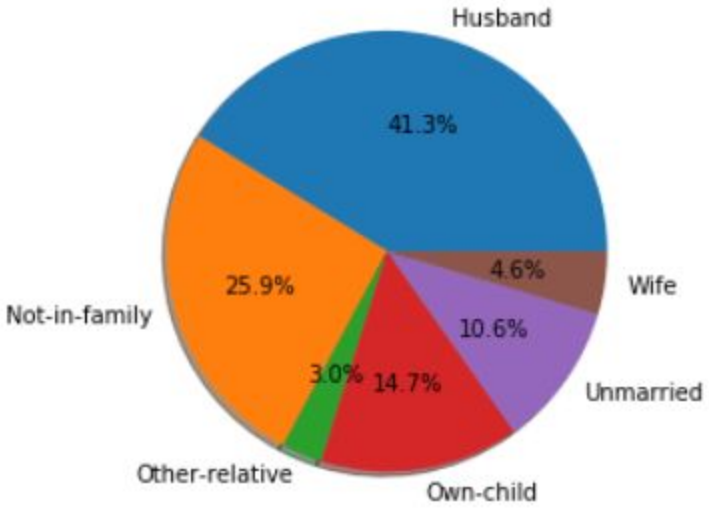


Who are the majority in the >50k income class?



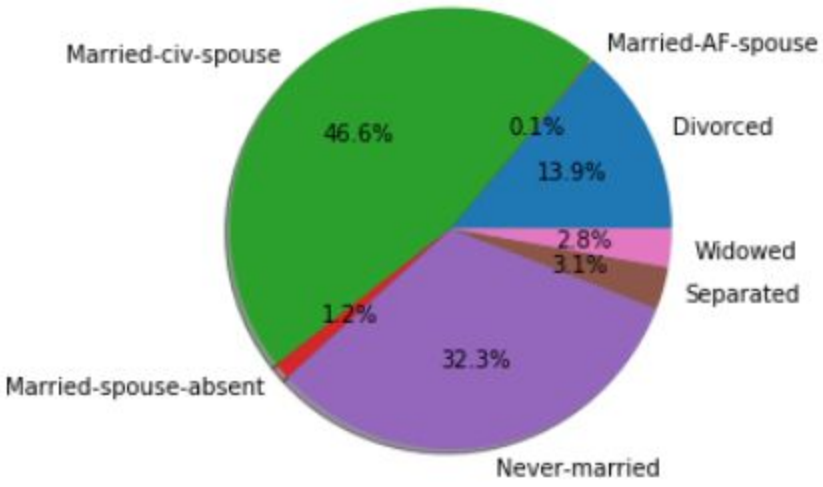
The white race seem to have dominated the income class of more than 50k. But then the given population itself is dominated by white.

Relationship



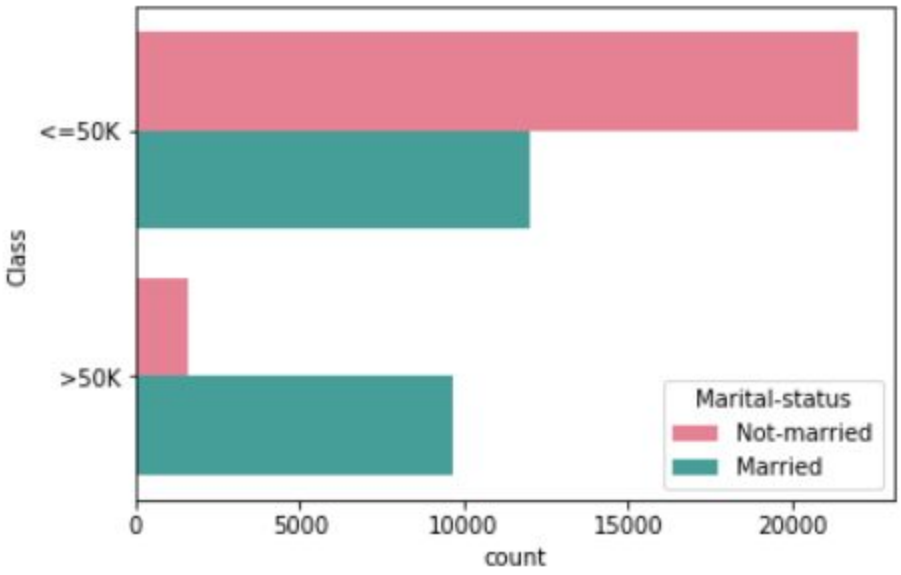
Given data set seems to be having lot more men than women.

Marital Status



There are so many categories in Marital status feature. For simplicity sake I will group it into 2 groups. Status as divorced, separated, widowed and never married as "Not married" and status as married-civ-spouse, married-spouse-absent, married-AF-spouse as "Married".

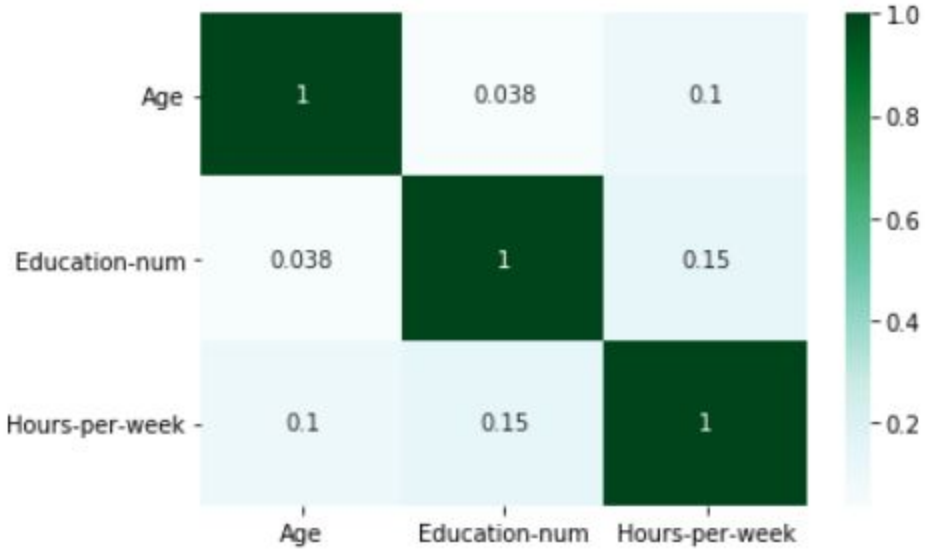
Does the income class of >50k have more married people or not married people?



Majority of the income class of >50k has married people. But the income class of <=50k has more Not-married people than married people.

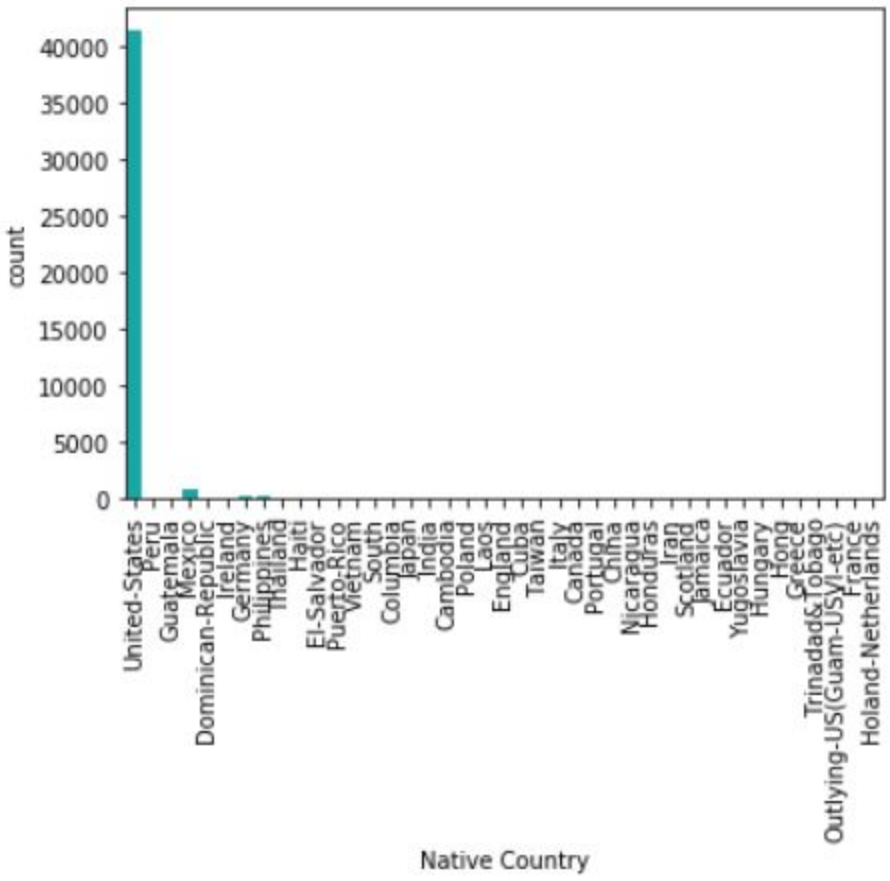
Maybe the responsibility of being married and having to maintain the household will push the person to earn more.

Correlation between Age, Final Weight, Number of education years, Hours/week worked

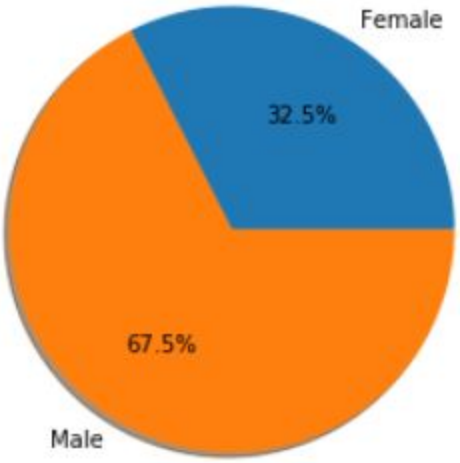


None of the features are strongly positively correlated or negatively correlated. Weak correlation between the numerical features.

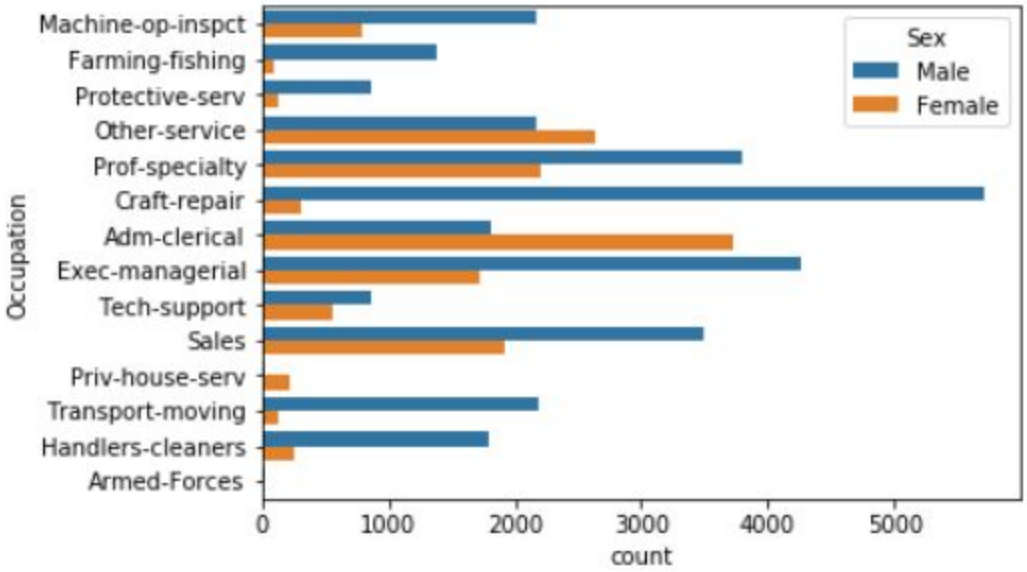
Native Country



Sex

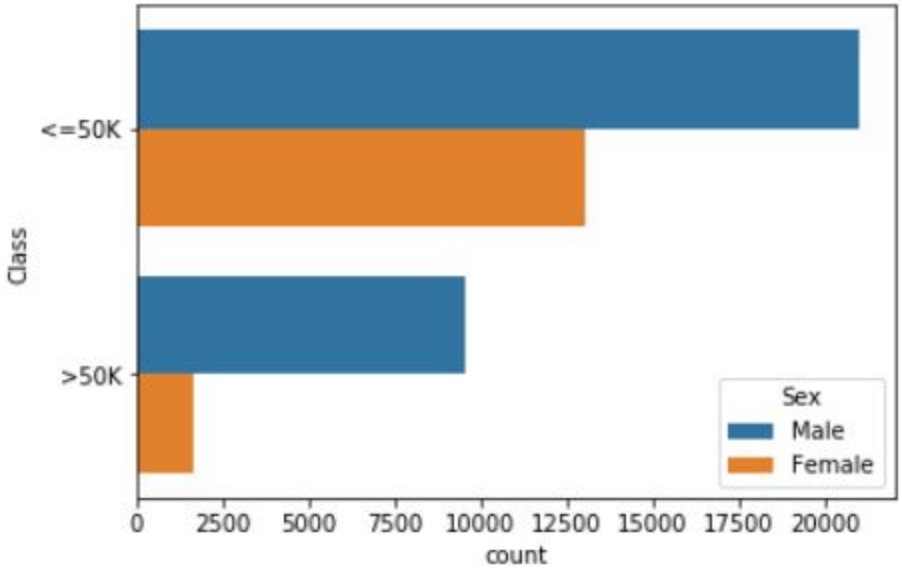


Distribution of men v/s women in various occupation



There are significantly more men than women in transport-moving occupation, which can be explained by the need of physical strength for that occupation. Admin-clerical occupation has more women than men.

Does the income group of >50k have more men or women ?



Men comprise of 2/3rd and women of 1/3rd the given population.

1/3rd of the population is women, but only 1/8 of the income class of >50k is women. Most of the people in the income class of >50k are men.

Modeling the data

I have chosen to model the data on Support Vector Machine and Random Forest Classifier algorithm. Support Vector Machine is considered for relatively clean and less noisy data. Random Forest Classification Algorithm is useful when the dependent variables are mix of categorical and numeric values.

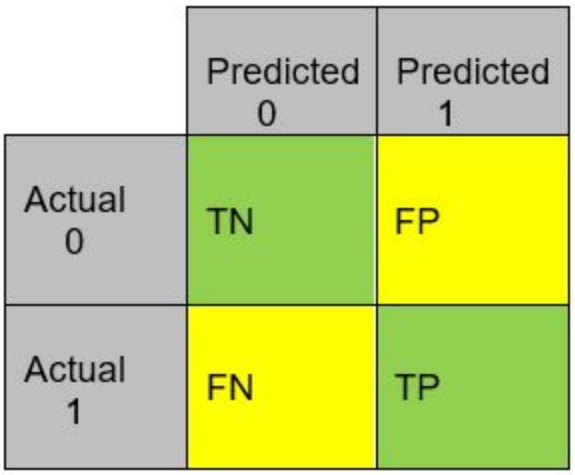
The data is wrangled and ready to be modelled. We will split the data into training set and train different models on it and test the model on the testing set. We will split the data set into 80% training and 20% testing data set. The data has 25% “>50K” class and 75% “<=50k” class. In order for the training and test set to have the same proportion of target variable in the split data, we set the parameter stratify to the target variable.

Performance Metrics

For evaluating the performance of the model, we will use Confusion matrix and Area under the ROC curve.

Confusion matrix

Assess the accuracy of the predicted model. It gives the number of True positive, False Positive, False Negative, True Negative cases.



TN – True Positive
FP – False Positive
FN – False Negative
TP – True Positive

$$Accuracy = \frac{TN + TP}{TN + FP + FN + TP}$$

Accuracy is the proportion of correct classifications from overall number of cases.

$$Precision = \frac{TP}{(FP + TP)}$$

Precision gives proportion of correct positive classifications from cases that are predicted as positive.

$$Sensitivity(Recall/TPR) = \frac{TP}{(TP + FN)}$$

Sensitivity or Recall gives the proportion of correct positive classifications from cases that are actually positive.

$$Specificity = \frac{TN}{(TN + FP)}$$

Specificity gives proportion of correct negative classification from cases that are actually negative.

$$False\ Positive\ Rate(FPR) = 1 - Specificity$$

F1 score = 2((Precision*Recall)/(Precision+Recall))
F1 score is needed when we seek a balance between Precision and Recall.

Ideal systems will have recall, precision as 1, which is impossible.

Area under the ROC

ROC(Receiver Operating Characteristic) is the probability curve and the AUC(Area under the curve) gives the measure of separability. ROC plots TPR on Y-axis and FPR on X-axis while varying the threshold. AUC denotes how well the model is distinguishing between

classes. Higher the AUC, better the model is distinguishing between classes. A perfect model has TPR 1 and FPR at 0 which would give AUC as 1 which means it is separating the classes perfectly. AUC close to 0 means model is not performing well.

Data Modeling

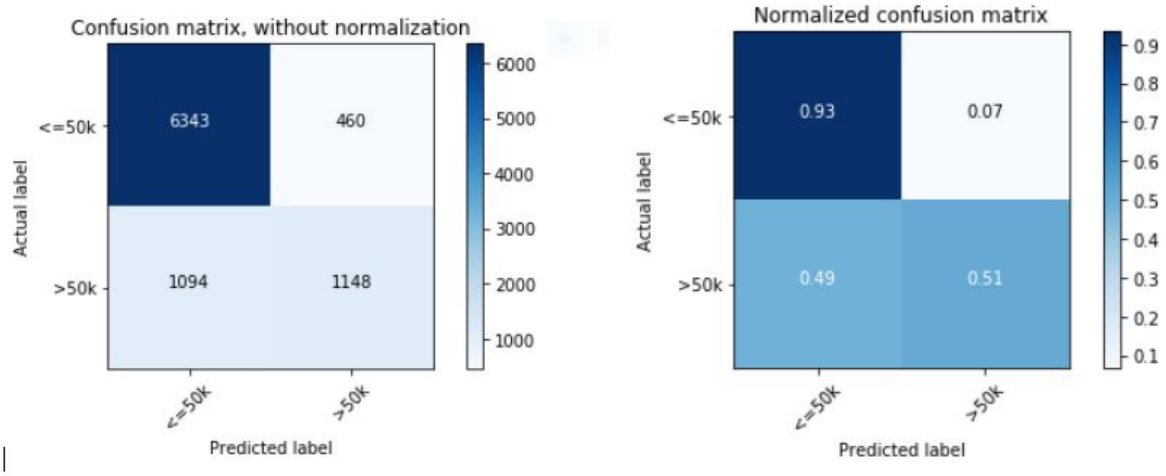
The results obtained for SVM and Random Forest algorithm with default parameters and after tuning the parameters are given below in the form of Confusion matrix, Classification report. AUC is given for the better model for each algorithm.

Support Vector Machine

A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Support Vector Machine with default parameters

Confusion matrix



Here negative denotes the class <=50k, positive denotes the class >50k

From the confusion matrix, we learn that out of 9045 samples,

6343 samples were True negatives meaning those samples were classified as <=50k and they actually belong to <=50k class

1148 samples were True positives meaning those samples were classified as >50k and they actually belong to >50k class

1094 samples were False negatives meaning those samples were classified as <=50k but they actually belong to >50k class

460 samples were False positives meaning those samples were classified as >50k but they actually belong to <=50k class

Classification Report

Accuracy score for test set 0.8281923714759536

	precision	recall	f1-score	support
0	0.85	0.93	0.89	6803
1	0.71	0.51	0.60	2242
avg / total	0.82	0.83	0.82	9045

From the classification report, we learn that 82.8% of the test data was correctly classified.

Hyper tuning the parameters for SVM Algorithm

We can improve the performance of the model by tuning it's parameters. We will tune the parameter C (regularization parameter) and gamma and check the performance of the SVM model.

C (Regularization parameter)

The C parameter tells SVM optimization how much you want to avoid misclassifying each training example. For large values of C, the optimization will choose a

smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassified more points.

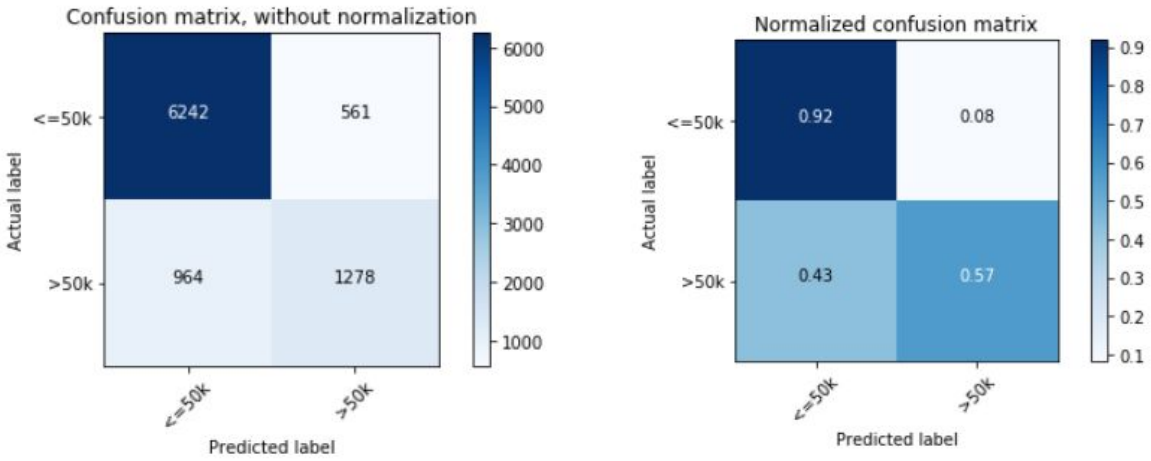
Gamma Parameter

The gamma parameter defines how far the influence of a single training example reaches, with low values meaning far and high values meaning close. With low gamma value, points far away from plausible separation line are considered in calculation for the separation line. Where as high gamma value means the points close to plausible line are considered in calculation.

The best parameters obtained through GridSearchCV are Gamma=0.1, C=1. The performance with optimum parameters is shown below

Support Vector Machine With Gamma = 0.1

Confusion matrix



Classification Report

	precision	recall	f1-score	support
0	0.87	0.92	0.89	6803
1	0.69	0.57	0.63	2242

avg / total 0.82 0.83 0.83 9045

Accuracy score for test set 0.8313985627418463

92% of the Class 0(<=50k) is classified as Class 0(<=50k).

Only 57% of the Class 1(>50k) is classified as Class 1(>50k). Performance of the model on classifying Class 1(>50k) is lower than it's performance classifying Class 0(<=50k).

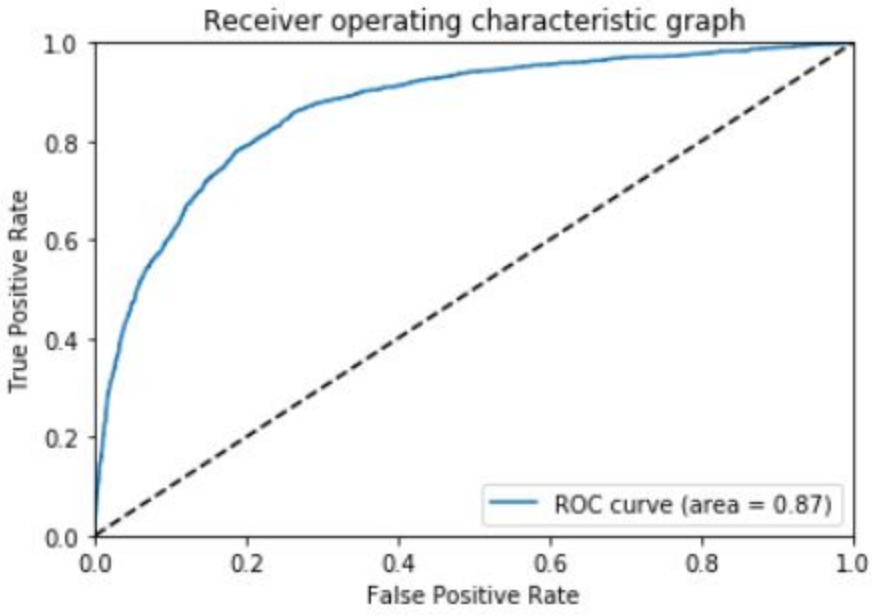
When Class 0(<=50k) is classified as Class 0(<=50k), it is correct 87% of the time. Which also means Class 1(>50k) is wrongly classified as Class 0(<=50k) 13% of the time.

When Class 1(>50k) is classified as Class 1(>50k), it is correct 69% of the time. Which also means Class 0(<=50k) is wrongly classified as Class 1(>50k) 31% of the time.

If this model is used for marketing purpose, 13% of the '>50k' class of people will not receive the campaign material targeted for them. And 31% of '<=50k' class would receive the campaign material which would be irrelevant to them, which would be waste of marketing resources.

With optimal parameter values of gamma = 0.1 and C = 1, we can see slight improvement of accuracy score of 83.13% when compared to base model performance with accuracy score of 82.6%

Area under ROC

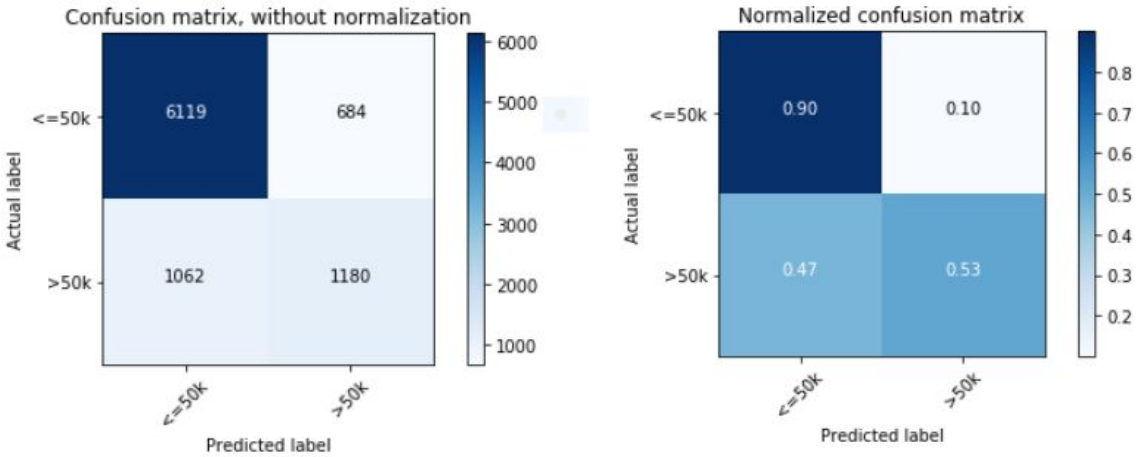


Area under the ROC curve : 0.8671

Random Forest Classifier

Random Forest Classifier with default parameters

Confusion matrix



Classification Report

	precision	recall	f1-score	support
0	0.85	0.90	0.88	6803
1	0.63	0.53	0.57	2242
avg / total	0.80	0.81	0.80	9045

Accuracy score for test set 0.8069651741293532

GridSearchCV

The Random Forest Classifier's parameters can be tuned to increase the predicting power of the model. Some of the features that I have picked to tune here are max_features, n_estimators, min_sample_leaf

.

max_features

This is the maximum number of features the algorithn is allowed to try in an individual tree. Increasing the max_features improves the performance of the model as there are more options to be considered at each node. But this decreases the speed of the algorithm. Optimum number of features should be chosen to balance the performance and speed of the model

n_estimators

This is the maximum number of trees that can be formed by the model before taking the vote. Higher the number of trees, better the predicting power. Again this comes at the cost of the speed.

min_sample_leaf

The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least min_samples_leaf training samples in each of the left and right branches

min_samples_split

The minimum number of samples required to split an internal node.

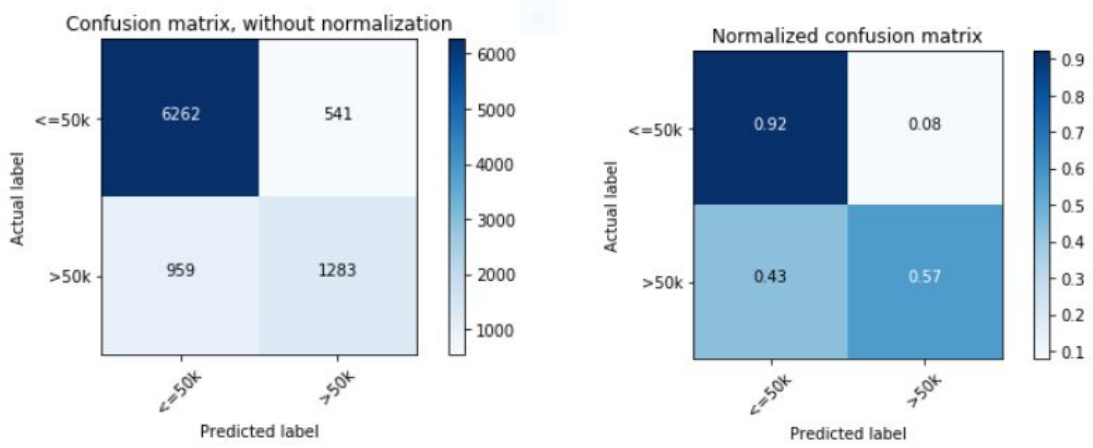
I am going to find the best params values for the above parameters by GridSearchCV.
The param grid is formed with the following parameters

```
param_grid = {
        'max_depth': [3, 4, 5, None],
        'max_features': [4, 6, 11],
        'min_samples_leaf': [3, 4, 5],
        'min_samples_split': [8, 10, 12],
        'n_estimators': [100, 200, 500]
}
```

The best parameters obtained are
{'max_depth': None, 'max_features': 11, 'min_samples_leaf': 3, 'min_samples_split': 12, 'n_estimators': 100}

Random Forest Classifier with the best params obtained by GridSearchCV

Confusion matrix



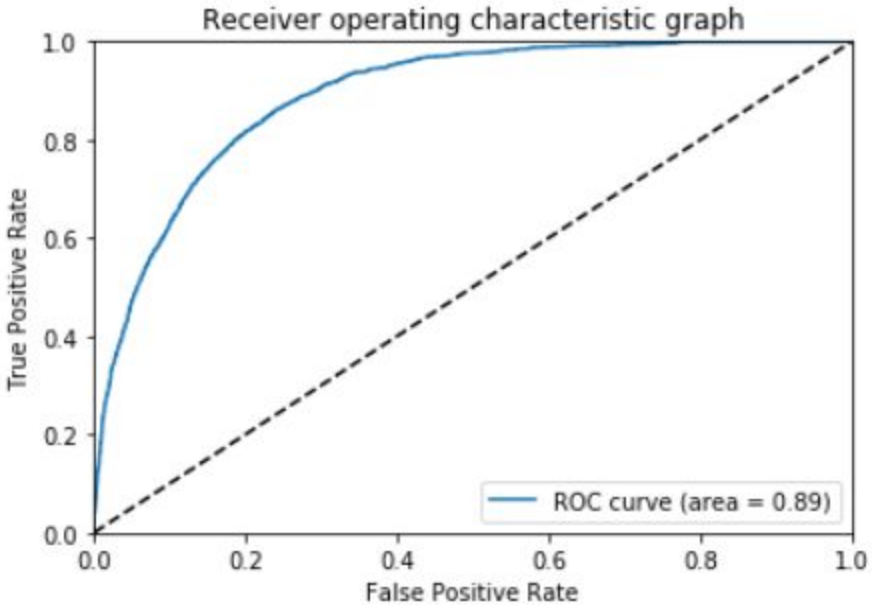
Classification Report

	precision	recall	f1-score	support
0	0.87	0.92	0.89	6803
1	0.70	0.57	0.63	2242
avg / total	0.83	0.83	0.83	9045

Accuracy score for test set 0.8341625207296849

Random Forest Classifier with best params is performing better with 3.4% increase in accuracy as opposed to the Random Forest model with default parameters with accuracy rate of 80%.

Area under ROC



Area under the ROC curve : 0.8903

Comparison of models

Model/Metric	Accuracy score	Precision		Recall		AUC
		<=50k	>50k	<=50k	>50k	
SVM (default Parameters)	0.8281	0.85	0.71	0.93	0.51	n/a
SVM (gamma = 0.1)	0.8313	0.87	0.69	0.92	0.57	0.86
Random Forest (default parameter)	0.8069	0.85	0.63	0.90	0.53	n/a
Random Forest (Best parameters)	0.8341	0.87	0.70	0.92	0.57	0.89

Conclusion

SVM with best param is performing at accuracy rate of 83.13% and Random Forest with best params is performing at accuracy rate of 83.4% showing Random Forest is slightly better.

We can see that SVM has AUC of 0.87 and Random Forest has slightly better AUC of 0.89, meaning Random forest is separating the classes a tad bit better than SVM model.

Further Steps

To improve the model, I should try to get more details on the features, Capital loss and Capital gain and build the model including those features. Also, parameters which are not tuned here can be tuned for better performance.