Department of Physics, Umeå University
October 27, 2025

Write the title of your report here

John Smith[1]*, Jennie Smith[1]

Abstract

Keywords
Optics — Interference — Diffraction

[1] *Department of Physics, Umeå University, Umeå, Sweden*
***Corresponding author**: john@smith.com
***Supervisor**: joe@doe.com

Contents

1. Introduction

2. Data analysis

2.1 Dataset

The dataset we decided to study is a labeled income prediction dataset. This dataset includes 14 features with information about the people in the study and a label with the income as either more than \$50 000 per year or less than or equal to \$50 000 per year. This means that we are looking at a binary classification problem. A lot of the features are discrete where only a set number of options available. This includes features such as marital status, education and working class. The dataset features around 32500 data points.

2.2 Data cleaning and feature engineering

There were a couple of things with our dataset that had to be modified in order for it to be usable in our ML application. We find that some of the features are redundant or not interesting in our project. We romove the redundant feature education since there is another already numerically encoded feature containing the same data. We also chose to remove the feature 'fnlwgt' since it is a already calculated number that is used by the Census Bureau to estimate population statistics. Since we want to estimate the population statistics based on the other features and not the already calculated weight we remove this feature. We have a mix of numerical and non-numerical features in our dataset. Since the machine learning models cannot use non-numerical data we have to encode the non-numercial data into corresponding numbers. This is with the label encoder built into sci-kit learn and used on all non-numerical data.

2.3 Handling missing values

With our numerical version of the dataset we found with the info function in pandas that around 2500 values were NaN values. We reasoned that filling these values with something as the mean of the category does not make very much sense for our application. Since there are many discrete categories a mean value means nothing. Especially since we gave many categories arbitrary numbers the mean means nothing. We therefore decided to only use complete data points. This resulted in removing about 6% of the total amount of data points or about 2500 data points.

2.4 Training, validation and test sets

Before doing any sort of training or analysis on the data, se split it into training, test and validation data. We did this by first splitting a random 20% of the data into test data. This data is reserved for the final testing of the model and will not be touched until the model is finished. Then we did a further split of the rest of the data were 25% was designated as validation data. This data will be used for calibration of the model and hyperparameter tuning. The rest of the data which is 60% of the total data or around 18000 data points will be used to train the model.

3. Model selection

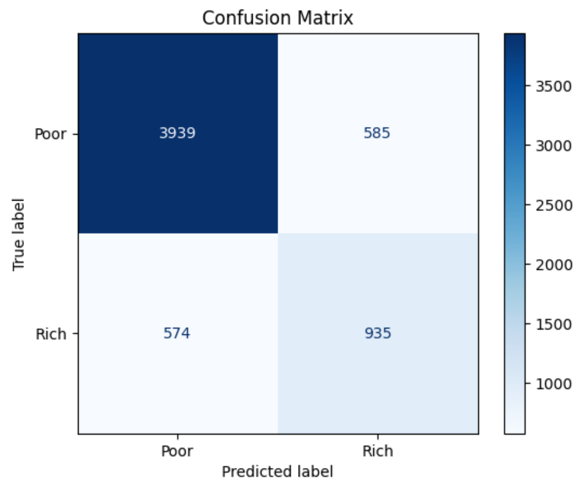
When selecting the model to use for this project we have to limit us to using models that are appropriate to the type of problem that we are trying to solve. The problem is a classification task so all models that are used for regression are immediately invalid. There are plenty of different types of classification models left to choose from. Many of them however, are good for data that has non-discrete features. This includes models such as logistic regression, KNN and other similar types of classification models. Also since we have so many features that are non-numerical and converted into arbitrary numbers these types of models would not be optimal. What is left is the Gaussian Naïve Baye's and the different tree based models. Naïve Baye's can be a bit troublesome for this dataset since we have found that some parameters are slightly correlated. However, this does not necessarliy make in an inappropriate method as it has been found to perform well despite this strict assumption. Therefore we are left with the tree based models such as the decision tree and random forests. We decided to implement two different types of models. We first do a decision tree and see how good we can get that model to work. We then do a random forest which may not be the absolute best model but since it is a continuation on the decision tree it might be interesting to see if it performs better. We then do analysis on both methods and see if these models are good enough and if there is any meaningful difference between the two.

4. Model Training and Hyperparameter Tuning

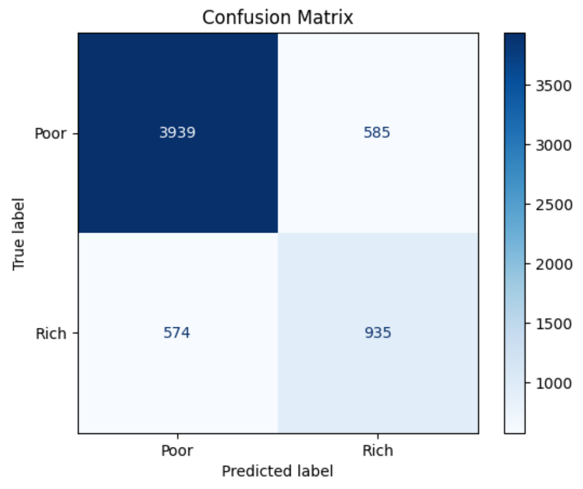
During the model training there are some important changes we can make to improve the accuracy of our model. One thing we implement is cross validation. Since there is a great spread in our data we choose to use randomized search. Another very important part of the model training is finding the optimal hyperparameters. This is an important step in minimizing the risk of overfitting. Some important hyperparameters in our decision trees are the maximum depth and minimum sample split. The maximum depth hyperparameter decides how deep the tree is allowed to go. If a tree is allowed to go very deep there is a high risk of overfitting. We therefore test multiple different depths and see which values give the best training and validation accuracy. This will ensure that we use the most optimal depth for our tree. The minimum sample split states how many data points there has to be for a new split to be created. This is also a good measure against overfitting since if it is very low we risk training the noise of the data instead of the general trend and end up overfitting the data. It is also important that it is not to small since we then loose information and underfit instead. For the random forest there is also the hyperparameter of how many estimators to use. This decides how many trees to choose from.

5. Model Evaluations

There are two interesting parts to look at after our analysis. One part is to analyze how well the actual models performed and compare the difference between the two models we have chosen to study. We fine tuned our models using the validation part of the data. After running it on the test data we can see how well it actually performs. A great way to get a quick overview of how well a model classifies is to look at the confusion matrix.



(a)



(b)

Figure 1. The confusion matricies of the Decision Tree model and the Random Forest model on the test data.

As we can see in the confusion matricies there is not that big of a difference between the models. Both did an overall good job at identifying the two classes. There is a difference in how well the models did in identifying the two different classes. Overall they performed a lot better at classifying the poor people than the rich. This is a very interesting result and maybe not so weird as it first seems. There were a lot more poor people in our training data set than rich people. This would of course train our model to be better at classifying the poor. As well as looking at the classification matricies it is interesting to look at the actual performance metrics that can be calculated from the matricies. These metrics can be seen in table(2). Of note is that all of these metrics are calculated as weighted metrics which means that they account for the class imbalances seen in the confusion matrcies. Looking at the values we see that the difference between our models

Table 1. The performance metrics of the models on the validation data.

Model	Accuracy	Precision	Recall	F1 Score	Total Time
RF	0.8589	0.8535	0.8589	0.8534	150.8154
DT	0.8483	0.8449	0.8483	0.8462	6.7357

Table 2. The performance metrics of the models on the test data.

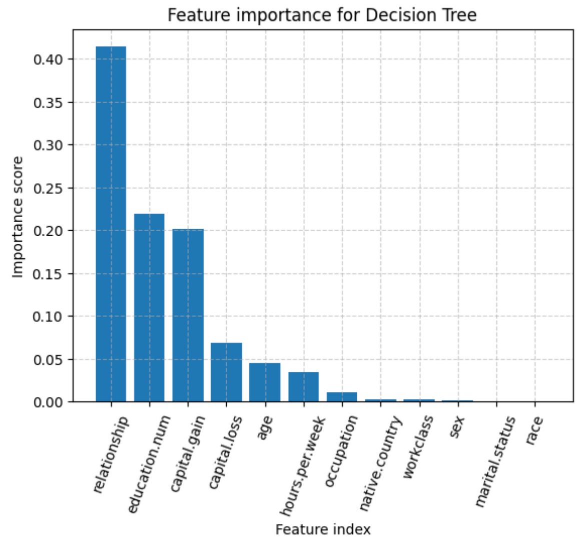
Model	Accuracy	Precision	Recall	F1 Score	Total Time
RF	0.8589	0.8535	0.8589	0.8534	150.8154
DT	0.8483	0.8449	0.8483	0.8462	6.7357

is not that large. The Random forest model is on average about 1 percentage point better than the Decision Tree. We can also see that all metrics are at about 0.85. This means that our models are not very accurate and that the differences between them is not that large at all. Which model that is better depends a lot on what is the priority. While it is clear that the Random Forest has the better performance, even by just a little bit, it is also significanty slower. So for this dataset was it really worth 30x the computational time to get a slightly better result? We are not really sure. The extra computational time is a definite negative but at the size of this dataset we are only talking about a couple of minutes which is not too bad. For another dataset the results may be different and it might be clearer which is really the prefered model.

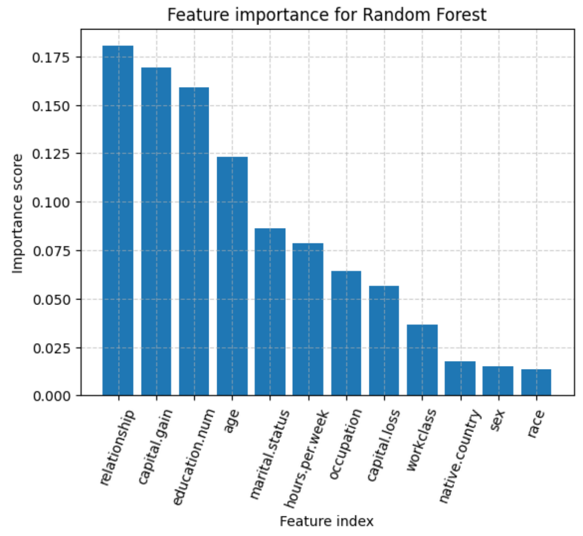
At a first glance at both the confusion matricies and the performance metrics the models do not look to be that good. But what has to be considered is the data that we are analyzing. We are looking at what possible indicators there are for a person to earn more than a certain amount of money. This is real world data and in the real world there is a lot of unique ways of earning money. While there certainly are some indicators that will clearly tell that somebody is earning a lot of money, there are other factors that are not as telling. This means that some features are less important than others. This can be seen in our models int he feature importance graphs in figure(2(a)) and (2(b)). This also means that there will be plenty of outliers in the data. No matter how good the model is, it cannot possibly catch all of these outliers. If it did it would be overfitted. We simply cannot expect a model to have very good accuracy on this type of data set.

Taking a closer look at the feature importance graphs of the two models we notice an interesting difference. The Decision tree which is only one tree focuses has only a few main features where one is the most important. The rest are not used that much or almost not at all. The Random Forest uses a far wider range of features. They also rank the features a bit differently and the best feature for one model is not the best for the other. We considered removing the worst performing features to see if it would make a difference in the performanes. But since they have diffrent worst performing features we reasoned that to keep the comparison as fair as possible it would be more interesting to leave the features as is.

We spent some time tuning the hyperparameters to ensure that we did not overfit. We can also see if we



(a)



(b)

Figure 2. The feature importance graphs for the Decision Tree model and the Random Forest model.

References

[1] Steinhaus, H., Mathematical Snapshots, 3rd Edition. New York: Dover, pp. 93-94, (1999)

[2] Greivenkamp, J. E., Field Guide to Geometrical Optics, SPIE Press, Bellingham, WA, (2004)

[3] Pedrotti, F.L. and Pedrotti, L.S., Introduction to Optics, 3rd Edition, Addison-Wesley, (2006)

[4] UC Davis ChemWiki, Propagation of Error, Available at: https://chem.libretexts.org/Textbook_Maps/Analytical_Chemistry/Supplemental_Modules_

(Analytical_Chemistry)/Quantifying_Nature/
Significant_Digits/Propagation_of_Error, (Accessed:
10th March 2016).