Coen 432 Final Project

Keeano Gerald - 40095571
Hugh McKenzie - 40088023
Concordia University - Coen 432
Presented to: Dr. Nawwaf Kharma

December 3rd, 2022

Contents

List of Figures

1 Problem

In this assignment, we investigated the performance of three machine learning models on the Adult Data Set, from the UCI Machine Learning Repository, to predict whether the income of a certain individual exceeds $ 50 000 per annum. The binary classifiers required to perform this analysis are: K-Nearest Neighbour (KNN), Decision Tree (DT), and Support Vector Machine (SVM). Using Grid Search and K-fold Cross Validation as the optimization methods to find the best parameters for the models, we were able to conclude which model returns the highest accuracy with our given data set, and compare the efficiency of the run times between the models. While a perfect model is ideal, it's almost never obtained. Thus, as the goal of this project was to predict the income of an individual, we aimed to obtain the most accurate model given the different parameters and models.

2 Dataset

The Adult Data Set from the UCI Machine Leaning Repository had been previously split into a training and testing set. Therefore, before starting the preprocessing of our data set, we concatenated the two files and removed all the non-numerical values. As this is a binary classification problem, we changed all the non-numerical data to binary values - using one hot encoding - before splitting the data into 20% testing, 80% training. Finally, since both KNN and SVM are methods that measure distance, we normalized the data by scaling it to lower the distance between the data points. While there are many attributes to this data set, we wanted to be able to predict the income of an individual, so our label was the 'income' column containing 1 if an individual's income was $>$ 50K and 0 if $\leq 50K$.

3 Optimization

Pseudocode for Optimization Grid Search Algorithm :

```
parameter1 = []
parameter2 = []
parameter3 = []

Given a list for a range of parameters,
for i in parameter1:
    for j in parameter2:
        for k in parameter3:
            Build a parameter combination from each i,j,k
            Run model on each combination and obtain accuracies

return best parameter combination
```

Figure 1: *Grid Search Pseudocode*

We used a grid search to find the optimal parameters for our machine learning models. It builds a model on each parameter combination that exists (including with the Cross

Validation Folds) by iterating through every possible combination and fits a model for each combination.

4 Results

To perform the analysis on all three of our machine learning models, we first pre-processed the data, then used a grid search to get the best parameters in order to then plot our data using the Receiver Operating Characteristics (ROC) Curve graph as well as the Area Under the Curve (AUC) for all three machine learning models.

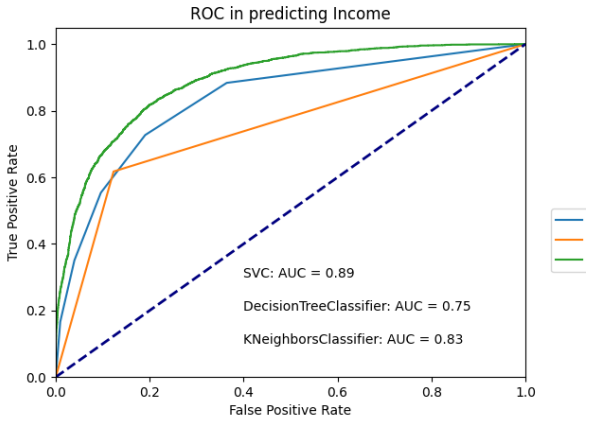

Figure 2: *ROC Curve*

A perfect classifier has a ROC-AUC equal to 1, while a random model would have a ROC-AUC equal to 0.5. Our models, as shown above in figure 2, are close to 1 which shows that the model closely predicts whether a person makes over $50 000 a year. As shown, the SVC returns the highest accuracy, followed by KNN then DT.

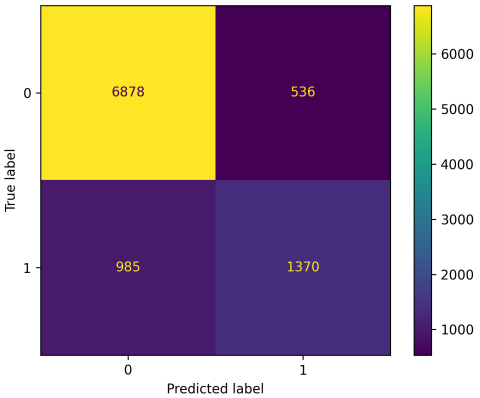

Figure 3: *Confusion Matrix for kNN*

The figure above shows the confusion matrix for KNN. We have most of the predictions condensed in the top left and bottom right. This is a great outcome as it shows that there were more True Positives than false positives, and more true negatives than false negatives i.e better classified data.

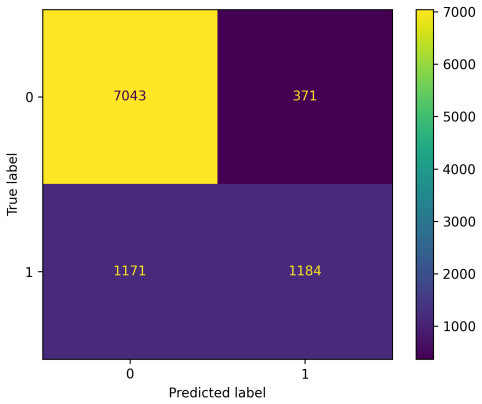

Figure 4: *Confusion Matrix for DTC*

Secondly, the figure above shows the confusion matrix for our decision tree. At first, we believed it would be interesting to do a simple decision tree rather than random forests. The reason for this would be to see if a simple decision tree would cause over fitting, resulting in a low test accuracy. This was not the case. Again, we have a large number of predictions condensed in the top left and bottom right. As seen in the ROC curve, it fits less accurately than the kNN and SVM. Although it is quick to produce, DTC's will not be used as they produce a less accurate ROC than kNN.

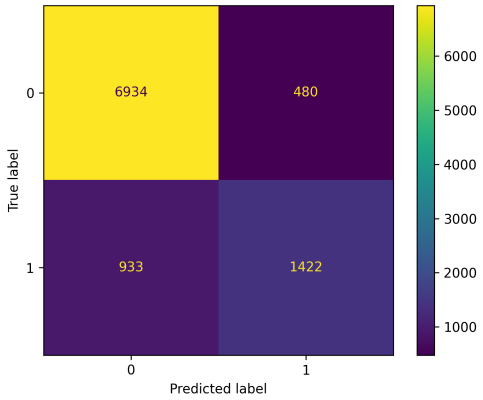

Figure 5: *Confusion Matrix for SVC*

Lastly, the figure above shows the confusion matrix for the support vector machine. Again, a large number are condensed in the top left (true positive) and bottom right (true negative) which is ideal. Furthermore, the ROC curve produced by the SVM is superior to that of the decision tree and kNN. However, with the time it takes to produce the model for SVM, kNN seems to be more suitable.

Our choice for parameters was mainly centered around runtime efficiency depending on each model and each model used a 10 fold Cross Validation. For the KNN grid search, as the only parameter to test was K - the number of nearest neighbours - we chose a range of 15 as as it had an fairly efficient run time as it fit the model 150 times. The final accuracy obtained for our testing dataset with tuning using KNN is 86.70% with {'n_neighbors': 5} chosen as the best K value.
For the DT grid search, we chose 'max_leaf_nodes', 'min_sample_split" and 'max_depth' as our three parameters and used a combination of values. We chose max depth as the the deeper the tree, the more information it can capture about the data. Next, we chose min_sample_split as when it increases, we have to consider more samples at each node. Finally, we chose 'max_leaf_nodes' to simply minimize the number of leaf nodes in the tree. For DT, the Grid search performed fairly quick with the combination of parameters at 10x Cross Validation so we were able to have a bigger range and more parameters. The final accuracy for our testing data set with tuning using DTC is 84.22% with {'max_depth': 5, 'max_leaf_nodes': 8, 'min_samples_split': 3} as the best parameters.
Lastly, for the SVM grid search, we had to keep the range and values for the three parameters fairly small since otherwise the run time would take too long. In fact we were unable to test the model using 10x Cross validation as using even 5x Cross Validation exceeded 90 minutes. Thus, the following accuracy for our testing data set with tuning using SVC and 3x Cross Validation is 85.54% with {'C': 1, 'gamma': 1, 'kernel': 'linear'} as the best parameters. The choice of parameters heavily depended on the run time of fitting the model, more precisely, in the choice of C and gamma. Likewise we chose a linear kernel instead of the RBF, polynomial, and sigmoid kernels for the same reason.

5 Summary

After comparing the training and testing accuracy, as well as the ROC curves of our three machine learning models, we were able to conclude that KNN was our most successful model. More specifically, we came to this conclusion based on the run time, as well as the accuracy. The run time of KNN was similar to that of the decision tree, and far superior to that of the SVM while creating only a slightly less accurate ROC curve than the SVM. Due to this, we were able to conclude that KNN was the most appropriate model to classify the income of individuals. Overall, we obtain a satisfactory result from our model and were able to predict the desired result at a rate that meets our expectations. We achieved an ROC-AUC close to 1 indicating an accurate model with an accuracy of 86%. Given these results, we can conclude that our model is quite adequate at predicting if an individual exceeds a $50 000 yearly salary based on our given attributes from the adult data set from the UCI Machine Learning Repository.