Data Mining: Prediciting High Earners

Daniel Ray
18053479
Department of Computer Science
Manchester Metropolitan Univesity
Manchester, United Kingdom
18053479@stu.mmu.ac.uk

***Abstract*— Contained in this report are the several necessary steps needed to carry out the process of data mining. The process beings with exploring the dataset to infer and analyse key features of the data. Upon completing EDA, features are then engineered and selected for the Data Mining algorithms to fit on. The parameters of said algorithms are then tuned in order to attain the best model. Finally, results are analysed to evaluate the best performing algorithm.**

Keywords*— *Data Mining, Machine Learning and Adults Dataset

I. Introduction

"A computer once beat me at chess, but it was no match for me at kick boxing" – Emo Philips

Data Mining is the process of generating new information from pre-existing datasets. In this report, details of each step of the data mining process will be documented then analysed. Details of the data mining algorithms and how they have already been used within the field and on the Adults dataset will be given. From that, an exploratory data analysis task will be carried out in order to gain a deeper understanding of the data and how it relates to each attribute.
Once enough information has been generated via the EDA, features will be extracted in order to run the data mining algorithms on. The parameters of each algorithm will be tuned in order to attain the best-tuned algorithm for the dataset. Finally, the results will be analysed to determine the best-suited algorithm for the Adults dataset.

II. Background

A. OneR

One Rule algorithm like than name implies generates one rule for each of the features in the dataset by constructing a frequency table for each feature split on the target variable. The algorithm then selects the rule to classify the data points by computing the total error for each feature, which measures the features contribution. The feature with the smallest total error is selected, as it produces the highest contribution to the predictability of the model. One R is an effective classification algorithm producing interpretable results however lacks the accuracy of more intricate classification algorithms.

B. Decision Tree (C4.5)

Decision trees are classification models that take the form of a tree, containing decision and leaf nodes. Decision nodes have two or more branches whilst leaf nodes represent the classification for that branch. Decision trees are built from the top down with the first decision node being the root node corresponding to the best predicting feature. To generate the final tree, the dataset is partitioned into subsets with similar values.
The core algorithm ID3, which C4.5 is built upon, uses information gain to evaluate the homogeneous of each feature. To compute Information gain, the following formula is used:

$$Gain(T, X) = Entropy(T) - Entropy(T,X)$$

The most homogenous feature is selected as the decision node, leaf nodes are then created by calculating the entropy. If the entropy is zero then that branch becomes a leaf node, whilst an entropy more than zero requires more splitting. Thus needing different functions for both entropy types.
To compute the entropy of two attributes:

$$E(T, X) = \sum_{c \in X} P(c)E(c)$$

To compute the entropy of one attribute:

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

C4.5 improves on ID3 as it accepts both continuous and discrete values, handles incomplete data and solves the common issue of overfitting with decision trees by pruning the tree using a bottom up approach. Decision trees produce interpretable results which can be mapped into if statements or select case statements. Below are the parameters that will be tuned in order to achieve the best model:

a) Minimum Number of Objects: Minimum number of instances per leaf.

b) Confidence Factor: Used for pruning.

C. K-Nearest Neighbour (IBk)

K Nearest Neighbour is a classification algorithm that is able to classify new data points based on a similarity measure. A data point is assigned a class based on the most occurring class amongst that data point's neighbours. Within the KNN algorithm, there are several methods to compute the similarity measure; Euclidean:

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

As well as, Manhattan:

$$\sum_{i=1}^{k}|x_i - y_i|$$

Both produce the distance of a data points nearest neighbours. K is in reference to the number of neighbours used to evaluate that data points class. Generally, larger k values produce more precise classifications due to its noise reduction capabilities. Below are the parameters that will be tuned in order to achieve the best model:

a) KNN: the amount of neighbours to evaluate against.

b) Distance Function: The similarity measure.

D. Naïve Bayes

Based on Bayes' theorem, Naïve Bayes is a classifier that uses the assumption, class conditional independence, that the effect of the value of the feature on a given class is independent from the values of other features. The posterior probability is computed using:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Where, P(c) is the prior probability of class, P(c|x) is the likelihood or the probability of the feature given class and P(x) is the prior probability of the feature.
Naïve Bayes is an effective data mining algorithm that outperforms more sophisticated algorithms. Below are the parameters of the algorithm which will be tuned:

a) Use Supervided Discretization: converts numerical attributes to nominal ones.

E. Random Forest

Random forests are an extension of decision trees such that they build multiple decision trees and merge the results together in order to attain an accurate classification. Random Forests whilst growing the trees, instead of searching for the most important feature when splitting on a node, it creates random subsets of the dataset and finds the best feature within the subsets. This process adds more randomness to the model producing better results. Overfitting is somewhat eliminated if enough trees are in the random forest however this takes time causing model development time to be slower. Below are the parameters of the random forest algorithm that will be tuned:

a) Max Depth: the maximum depth of the tree

b) Num Features: Set the number of randomly selected attributes

III. Related Work

Within the Data Science Community, the UCI Adults dataset is one of the most recognisable due to its 'real world' nature allowing aspiring data scientist to run experiments and gain a deeper understanding of Data Mining techniques and workflow. Due to the notoriety it has received, the dataset has a substantial list of citations [1].

One of the most notable citations is R. Kohavi's investigation into scaling up the accuracy of the Naïve Bayes Classifier using a decision tree hybrid. In the paper, a discussion of how the Naïve Bayes algorithm struggles to run as accurately when scaled up comparing it to a decision tree which does. Kohavi theorised and implemented a cross breed, called NB Tree where the decision-tree nodes contain univariate splits as regular decision-trees, but the leaves contain Naive-Bayesian classifier. NB Tree when run on larger datasets outperformed both original algorithms [2].

Rosset produced an examination of model selection based on the Area under Curve (AUC) metric. A comparative study was conducted comparing AUC to the empirical misclassification error with a goal to minimise future error rates. The study showed that AUC is a valuable metric to evaluate models performance. [3]

Bharath University's work with Rapid Miner saw the development of a new analysis and data mining tool called rapid miner. The tool used in both business and industry for quick prototyping supporting all steps of the data mining cycle. [4]

One of the main issues with the Adults dataset is the disparity of class sizes, as there are nearly 3 times more low earners as high. Zadrozny investigated how selection bias affects data mining algorithms proposing a solution to the problem. [5]

Cohen and Singer used the dataset to create a rule learner called SLIPPER that generates the ruleset using a repeated process of boosting a simple greedy rule generator.

Slipper is a highly scalable and effective learner scaling no worse than O(n log n), n being the number of examples. Compared to RIPPER and C4.5, slipper achieved lower error rates of 20 times and 22 times respectively [6].

IV. Dataset

The Adults UCI dataset first came into existence by Barry Becker when he extracted a subset of records of the 1994 US Census Bureau Database [1]. The dataset contains 48,842 entries with 15 columns, with differing levels of measurement. Looking at the last column in the dataset, 'Income', we see there are two values, <=50k and >50k, which represents whether that person earns more than 50k or not which will be used to classify that entry. As the target variables are contained in the dataset, supervised learning will be employed to predict the class of given data points.

A. Nominal Data

Nominal data is the first level of measurement being labels that have no order or quantitative value, in the dataset there are six instances of nominal data along with the target variable.

There are two columns related to work; work class and occupation. Work class has 8 values with Private being the mode with 33,906 entries. Splitting on the target variable we see that both high and low earner majority work in the private sector whilst jobs in government tend to have low earners. The mode occupation is Prof-specialty with 6,172 entries however Craft-repair, Exec-managerial, admin clerical and sales all range between 5,550 -6,000. High earners tend to work in Exec-managerial roles where as low earners tend to work in service jobs such as admin cleric and craft-repairs.

Looking at Marital status, low earners tend to have never married gaining over 14,000 entries whilst high earners tend to be married with a civilian spouse. People who are divorced, separated or widowed tend to be low earners.

Race and Native country show that the dataset contains majority white American people as both high and low earners tend to be white however, there tends to be more non-white people who are low earners compared to high earners. Looking at native country it's clear that both high and low earners tend to be American however there are more low earners who are from Mexico, El-Salvador and Philippines.

Gender shows us that both low and high earners tend to be Males showing the dataset to be male heavy having nearly twice as many males as females. That being said, high earners are four more times likely to be male than female.

B. Ordinal Data

Ordinal data is categorical data where an order is implied. Within the adult's dataset, there are only two instances of ordinal data, Education and Educational-Num both of which are depicting the same information just encoded differently. The most occurring education level is High School graduates with 14,974 entries. The highest level of education, Doctorate only saw 576 entries.

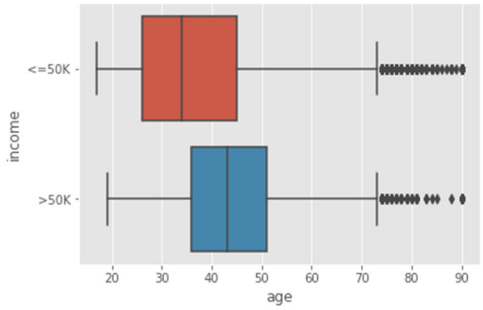
Splitting the data on the target variable, high earners tend to have higher education levels with Bachelors being the most occurring whilst for low earners, High School Grads occurred most.

C. Ratio Data

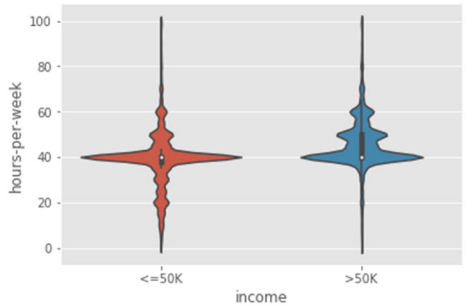
Ratio data is the highest level of measurement having all the characteristics of nominal and ordinal as well as having equivalent distance between categories but differing from interval data as it has a meaningful zero. There are four instances of ratio data in the dataset.

Looking at the distribution of age, there is a skew to the left, a mean of 38.55 and an IQR of 28-47. Splitting age on income, it is clear that high earners tend to be older than low earners with high earners having a 99% CI of [43.76 – 44.26] whilst low earners has a 99% CI of [36.56 – 36.93]

showing a clear difference in means of age dependant on income. Looking at the boxplot, there are many outliers on the upper boundary due to people above 70 inclining to retire however some still remain in the workforce.



Hours per week is another instance of ratio data. Looking at the distribution its apparent that it adheres somewhat to a normal distribution with an IQR of 40-45 and a range of 1-99. The average hours worked per week is 40.95 with a standard deviation of 12. Splitting hours worked per week on income, it's clear that high earners tend to work more hours having a 99% CI of [45.43-45.95] whilst low earners have a 99% of [39.21-39.55]. Splitting hours per week on genders shows a difference of means, males have a 99% CI of [42.69-43.03] whilst females have a 99% CI of [36.70-37.20].



The last two instances of ratio data is capital gain and loss both of which have interesting distributions, both have an IQR of 0-0 whilst capital gain has a range of 0-99,999 and capital loss has a range of 0-4,356. Capital loss has a mean of 88.71 and a standard deviation of 405.27 whilst capital gain has a mean of 1,114.09 and a standard deviation of 7588.77. This shows that all data points above zero are outliers for both capital gain and loss.

D. Missing Data

In order to get a complete dataset, missing data has to be dealt with using several methodologies. The first instance being work class that has 2,799 entries with missing data. As work-class and occupation are related, it would be easy to infer the work class from the occupation however, for all 2,799 entries; both work-class and occupation are missing so these entries were removed.

The next instance of missing data was occupation, having seven more missing entries however, the work class for those entries is 'never worked' so the occupation must be unemployed.

The final instance of missing data is Native Country that has 811 entries of missing data. In order to replace the missing data the mode Native Country being United States was used.

V. EXPERIMENTAL METHODOLOGY

A. Feature Engineering

Feature Engineering is the process of transforming raw data in such a way that represents the dataset to be better understood by the predictive models thus producing better result. Below are several methodologies employed to engineer the features:

a) Extracting Embedded Information: Within the Marital Status attribute there are two categories that depict married couples, a civil partnership or an armed forces one. In order to keep this information a new boolean attribute is created called Armed Forces Family which stores a 1 if that entry is part of an armed forces family or a 0 if not.

b) Map Categorical Attributes to Numerical: Due to the nature of Data Mining Algorithms, text attributes which represent categorical values are somewhat meaningless therefore these values have to be mapped to numerical ones, for example, Gender has two attributes; Male and Female which can be mapped to 0 and 1 retaining the information but represented differently. This method was employed on the target variable as well.

c) Decompose Nominal Attributes: Due to the nature of integers, for example 2 > 1, mapping numerical values to nominal attributes would imply an order, e.g. if Marital status was mapped with a range of 1-4 then category 1 would be less than category 3 which isn't the case. In order to keep the nominal attributes nominal, a technique called Dummy variables can be employed which extracts each variation of that attribute, creating a boolean feature for each. This method was used on Race and Occupation. Before using it on Work Class, government jobs were grouped together. Before using it on Martial Status, the attributes were mapped to 3 categories; Married, Seperated and Never-Married. Finally, Native Country was mapped to seven regions to reduce noise before creating dummy variables for each.

d) Noise Reduction: The ratio data in the dataset contains varying amounts of noise; the most noisy attributes being Captial Gain and Loss. In order to combate this issue, binning can be employed to transform the ratio data to ordinal retaining the pattern of the attribute but elimintating noise. Captial Gain and Loss were split into 4 bins each using the maximum value to create the cut points. Age and Hours per week employed floor division to create 10 bins for each attribute.

B. Training and Testing Strategy

Looking at the distribution of the target variable, there are 3 times more low earner entries than high which when fitting the data to models raises issues such as under or over fitting. Under fitting is in reference to the algorithm not capturing enough patterns in the data thus performing poorly whereas Overfitting is in reference to the model capturing noise and patterns in the data that do not generalise to unseen data points.

A strategy used to combat these two issues along with the inequality of the target variable is to employ a 10-fold Cross Validation training testing set. In a normal Training testing split the whole dataset is split into 3 groups, one to train the model on, one to validate the model on and finally on to test the model on. This method allows for over and under fitting as patterns in the data may not be contained in the training set so the model wouldn't be able to learn said patterns.

In 10 fold CV, every data point is in the training set nine times and in the validation set once. The model trained then validated using each of the 10 folds and the average accuracy is taken. This significantly reduces overfitting as most of the data is used to validate the model as well as reducing under fitting as most of the data is contained in the training set.

C. Parameter Tuning Experiments

a) Experiment 1: Preliminary

Aim*:* Run a base line experiment to evaluate the algorithms without any parameter tuning.

Method*:* Using Weka's experiment evironment, each algorithm will run producing results for comparison.

Results:

Algorithm	OneR	J48	IBk	NB	RF
Accuracy	77.22	83.07	80.99	80.74	82.15
Significance		(1/ /)	(1/ /)	(1/ /)	(1/ /)

Discussion: The results are conclusive that the J48 algorithm performed best, without parameter tuning, gaining an accuracy of 83.07. Random Forest ranks second whilst Naïve Bayes and K-NN saw an accuracy close to 81. Using one rule algorithm as the base line, its clear that all four algorithms performed significanly better using a paired t-test to test a difference in means of two samples.

b) Experiment 2.1: J48 Minimum Number of Objects

Aim*:* To achieve the best preforming MNO for the J48 Algorithm.

Method*:* Using a range of values (2-30) for the Minimum Number of Objects, stastically test performance to evaluate the best value for MNO.

Results:

Parameter	2	5	10	15	20	25	26	30
Accuracy	83.07	83.04	83.11	83.15	83.15	83.16	83.15	83.13
Significance		(/1/)	(/1/)	(/1/)	(/1/)	(/1/)	(/1/)	(/1/)

Discussion: The results for the Minimum Number of Objects parameter tuning shows a small increase in accuracy. Comparing to the orginal MNO to the most accurate model shows an increase of 0.09 accuracy being 25 MNO. All of the experiments show no statisically significant change from 2 MNO.

c) Experiment 2.2: J48 Confidence Factor

Aim: Discover the best Confidence Factor for the decision tree.

Method*:* Using a range of values (0.01-0.25), stastically test performance to evaluate the best value for Confidence Factor.

Results:

Parameter	0.01	0.05	0.1	0.15	0.2	0.25
Accuracy	83.10	83.35	83.50	83.52	83.44	83.39
Significance		(1/ /)	(1/ /)	(1/ /)	(1/ /)	(1/ /)

Discussion: A confidence factor of 0.01 statistically performed worse than all other CF's tested. Looking at the accuracy, 0.15 gained the highest with 83.52. The default 0.25 got 83.39 accuracy which sees a difference of 0.13.

d) Experiment 3.1: IBk Number of Neighbours

Aim*:* Discover the best performing number of neighbours in the KNN algorithm.

Method*:* To run the algorithm using a range of values for the number of neighbours.

Results:

Parameter	1	3	5	7	9
Accuracy	80.47	81.75	82.21	82.53	82.68
Significance		(1/ /)	(1/ /)	(1/ /)	(1/ /)

Discussion: The results show that there is a relationship between k and accuracy, as k increases so does accuracy. 9 neighbours achieved an accuracy of 82.68 which is the highest. All numbers of neighbours achieved a signficicantly better accuracy than the base of 1.

e) Experiment 3.2: IBk Distance Function

Aim*:* To establish the best distance function to model the data on using the KNN algorithm

Method*:* using a 10-fold cross validation, each distance function will be run to evaluate the best performing.

Results:

Parameter	Euclidian	Manhattan
Accuracy	80.47	80.48
Significance		(/1/)

Discussion: Both distance functions performed with 0.01 variance in accuracy. The Manhattan distance function performed the best gaining an accuracy of 80.48 however there is no statistical signifcance difference between the two distance functions.

f) Experiment 4.1: Naïve Bayes Use Supervided Discretization

Aim*:* Evaluate whether using the supervided discretization improves performance of the model.

Method*:* Run the Naïve Bayes using both the supervided discretization and without.

Results:

Parameter	False	True
Accuracy	80.60	80.26
Significance		(/ /1)

Discussion: There is a clear decrease in accuracy when using the supervided discretization with a decrease of 0.34. Using a paired t-test the results are significantly worse.

g) Experiment 5.1: Random Forest Max Depth

Aim*:* Determine which maximum depth performs best for the random forest algorithm.

Method*:* Using a range from 1-50, using infinite as a base line.

Results:

Parameter	∞	1	10	30	50
Accuracy	82.17	75.19	83.20	82.33	82.17
Significance		(/ /1)	(/ /1)	(1/ /)	(/1/)

Discussion: The results show that 30 is the most optimal max depth. Comparing to the base of no limit, 1 max depth was significanlty worse whilst 10 and 30 were significantly better. A max depth of 50 showed no significant change.

h) Experiment 5.2: Random Forest Num Features

Aim*:* Evalute which number of randomly selected features performs best.

Method*:* Using a range of values from 0 to 20 for the number of random feature to select.

Results:

Parameter	0	1	5	10	20
Accuracy	82.17	81.97	82.17	82.17	82.09
Significance		(/ /1)	(/1/)	(/1/)	(/1/)

Discussion: The results are conclusive that an increase in number of features isnt significantly different to the base line of 0. One feature has a significantly worse result, gaining 81.97. Both 5 and 10 gained the highest accuracy of 82.17.

VI. Experimental Results

A. Evaluation Criteria

In order to compare the algorithms performances different measures have to be computed in order to gauge how effective that algorithm was at modelling the dataset. Below I will detail the various measures I will use to evaluate performance as well as evaluating the best algorithm:

a) Accuracy: the ratio of number of correct predictions to the total number of data points. Accuracy is a good measure of performance however lacks robustness when the dataset isnt equally balanced.

b) Area Under Curve: Used for binary classification, the area under the curve is equivelent to the probability that the classifier will rank a randomly selected positive example higher than a negative one. It takes the range of 0 – 1 where 1 is the best performing model.

c) F-Measure: A harmonic mean between precision and recall. It is a measure of how precise and robust the model is at fitting to the data. High precision and lower recall produces a very accurate model but misses a large number of instances that are difficult to classify. Takes a range of 0-1, where the greater the f measure the better the performance.

B. One Rule

One Rule uses the lowest total error to find one feature to classify the dataset on. Upon running the algorithm on the dataset, it found that education number was the feature with the lowest total error. The model produces states that if the educational number is less than 13, the data point is classed as a low earner whilst 13 and above saw a class of high earner.

The time taken to build the model was very quick, taking only 0.48 seconds to build, classifying 35,555 data points correctly whilst only getting 10,488 data points incorrect. Looking at the evaluation metrics:

Metric	Accuracy	ROC Area	F-Measure
Low Earners	0.956	0.586	0.863
High Earners	0.215	0.586	0.319
Weighted Avg.	0.772	0.586	0.728

The results show that One R performed better on the low earner class than the high earner class having an accuracy of nearly 96% for low earners whilst only getting 21.5% for high earners, with a weighted average of 77.2%. The area under the curve gained a score of 0.586 showing that one rule has some issues with the dataset. Finally, the f measure shows a similar trend to that of accuracy where the model is able to predict low earners better than high earners however gaining a weighted average of 0.728, which shows the algorithm, is somewhat precise and robust.

C. C4.5 Decision Tree (J48)

Upon running the tuned J48 algorithm, using a confidence factor of 0.15 and 25 Minimum Number of Objects, a tree of size 123 with 62 leaves is produced in 11.32 seconds which is a lot longer than the one rule algorithm. Classification wise, J48 was able to classify 38,454 data points correctly only miss classifying 7,589 entries. Looking at the evaluation metrics:

Metric	Accuracy	ROC Area	F-Measure
Low Earners	0.919	0.870	0.893
High Earners	0.580	0.870	0.636
Weighted Avg.	0.835	0.870	0.830

The results show a similar trend to One R in that low earners performed better gaining an accuracy of 91.9% whilst high earners got 58.0% with a weighted average of 83.5%. The area under the curve gained 0.87 that is relatively high, suggesting the algorithm performed well. The f measure again is higher for low earners getting a score of 0.893 whilst high earners achieved 0.636. Overall, it is clear that the J48 performed well at fitting the data to a model in order to predict high earners.

D. K-Nearest Neighbours (IBk)

Using the optimal parameters of the Manhattan distance function and nine nearest neighbours a model was built in 0.02 seconds, which compared to the other algorithms performed the quickest. The IBk algorithm successfully classified 38,132 entries whilst incorrectly classifying 7,911 entries. Looking at the evaluation metrics:

Metric	Accuracy	ROC Area	F-Measure
Low Earners	0.908	0.871	0.888
High Earners	0.586	0.871	0.629
Weighted Avg.	0.828	0.871	0.824

The results show a similar trend to previous algorithms in that low earners performed better gaining an accuracy of 90.8% whilst high earners got 58.6% with a weighted average of 82.8%. The area under the curve gained 0.871 that is relatively high, suggesting the algorithm performed well. The f measure again is higher for low earners getting a score of 0.888 whilst high earners achieved 0.629. Overall, it is clear that the IBk performed well at fitting the data to a model in order to predict high earners whilst also being the quickest built model.

E. Naïve Bayes

Naïve Bayes, based on Bayes' Theorem computing the posterior probability. The model took 0.38 seconds to build, correctly classifying 37,107 entries whilst only miss classifying 8,936. Looking at the evaluation metrics:

Metric	Accuracy	ROC Area	F-Measure
Low Earners	0.838	0.867	0.867
High Earners	0.708	0.867	0.644
Weighted Avg.	0.806	0.867	0.811

The results show an increase in performance on high earners gaining an accuracy of 70.8% whilst low earners got 83.8% with a weighted average of 80.6%. The area under the curve gained 0.867 that is relatively high, suggesting the algorithm performed well. The f measure again is higher for low earners getting a score of 0.867 whilst high earners achieved 0.644. Overall, it is clear that the Naïve Bayes performed best at fitting the high earner data however didn't achieve the best metrics compared to others.

F. Random Forests

Running the tuned Random Forest algorithm, using a Maximum Depth of 10 a model was built in 26.62 seconds being the slowest to build. Classification wise, Random Forest was able to classify 38,312 data points correctly only miss classifying 7,731 entries. Looking at the evaluation metrics:

Metric	Accuracy	ROC Area	F-Measure
Low Earners	0.941	0.888	0.894
High Earners	0.502	0.888	0.597
Weighted Avg.	0.832	0.888	0.820

The results show a similar trend to the majority of algorithms in that low earners performed better gaining an accuracy of 94.1% whilst high earners got 50.2% with a weighted average of 83.2%. The area under the curve gained 0.888 being the best ROC out of the five, suggesting the algorithm performed well. The f measure again is higher for low earners getting a score of 0.894 whilst high earners achieved 0.597. Overall, it is clear that the Random Forests performed well at fitting the data to a model in order to predict high earners.

G. Comparison of Algorithms

In order to evaluate which algorithm performed best, Statistical tests were run on each of the evaluation metrics to test for similarity. Below are each of the metrics for all five algorithms:

a) Accuracy:

Algorithm	J48	OneR	IBk	NB	RF
Accuracy	0.84	0.77	0.83	0.81	0.83
Significance		(/ /1)	(/ /1)	(/ /1)	(/ /1)

The results show that the J48 had the best accuracy as the four other algorithms are significantly worse using the paired student's t-test. We see the decision tree gained an accuracy of 84%. K-Nearest Neighbours and Random Forests both gained an accuracy of 83% whilst One R got the lowest of 77%. Statistically, it is correct in saying that J48 was the most accurate algorithm for the dataset.

b) Area Under Curve:

Algorithm	J48	OneR	IBk	NB	RF
ROC	0.87	0.59	0.87	0.87	0.89
Significance		(/ /1)	(/1/)	(/1/)	(1/ /)

The results show that Random forests were significantly better than the other four algorithms, gaining 0.89. J48, IBk and Naïve Bayes all achieved a ROC of 0.87 having no significant difference whilst oneR was significantly worse gaining 0.59.

c) F-Measure:

Algorithm	J48	OneR	IBk	NB	RF
F-Measure	0.83	0.73	0.82	0.81	0.82
Significance		(/ /1)	(/ /1)	(/ /1)	(/ /1)

Finally, the results of the F-measure show that J48 performed best, gaining 0.83 which is significantly better than all remaining algorithms. Random Forests and K-Nearest Neighbours both gained 0.82 whilst Naïve Bayes got 0.81. One Rule performed the worse gaining 0.73.

VII. CONCLUSION

Examining the metrics for each algorithm it's apparent that J48 was the best suited to the problem, gaining the highest Accuracy and F-measure showing precision and robustness. Random Forests outperformed the J48 algorithm on the Area under Curve Metric showing that Random Forests are also a well-fit algorithm for the task.

One Rule, being the baseline, saw the worst results that is due to the simplicity of the algorithm, however, based on the results education number seems to be the feature with the smallest total error.

K-Nearest Neighbours saw competitive metrics, having 0.83 in accuracy and 0.87 AUC. Adjusting the parameters saw an increase in accuracy of 3%. Naïve Bayes, although

being a very interpretable algorithm the results didn't compare to those of j48 or Random forest

Thorough exploratory data analysis allows for better fitting algorithms. Having knowledge of the distribution of each attribute as well as knowing how each attribute relates to each other is hugely beneficial as features can then be engineered and selected in a method that increases performance of each algorithm.

An extension of the work carried out in this report could take the form of exploring more data mining algorithms such as Multi-layered Perceptrons and NB Trees or newly developed tools such as SLIPPER.

REFERENCES

[1] "UCI Machine Learning Repository: Adult Data Set", *Archive.ics.uci.edu*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/adult. [Accessed: 27- Mar- 2019].

[2] R. Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifers: a Decision-Tree Hybrid", *Robotics.stanford.edu*. [Online]. Available: http://robotics.stanford.edu/~ronnyk/nbtree.pdf. [Accessed: 27- Mar- 2019].

[3] S. Rosset, "Model Selection via the AUC", *Tau.ac.il*. [Online]. Available: https://www.tau.ac.il/~saharon/papers/auc-fixed.pdf. [Accessed: 27- Mar- 2019].

[4] D. Hanirex and K. Thooyamani, "An analysis on adult dataset in a decision tree using rapid miner tool", *Jchps.com*. [Online]. Available: https://www.jchps.com/issues/Volume%209_Issue%202/CSE%209.pdf. [Accessed: 27- Mar- 2019].

[5] B. Zadrozny, "Learning and Evaluating Classifiers under Sample Selection Bias", *http://citeseerx.ist.psu.edu*. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.92.170&rep=rep1&type=pdf. [Accessed: 27- Mar- 2019].

[6] W. Cohen and Y. Singer, "A Simple, Fast, and Effective Rule Learner", *Aaai.org*, 2019. [Online]. Available: http://www.aaai.org/Papers/AAAI/1999/AAAI99-049.pdf. [Accessed: 27- Mar- 2019].