# Classifier Tuning for the UCI Adult Salary Dataset

David Webb
Faculty of Science and Engineering
Manchester Metropolitan University
Manchester
david.webb@stu.mmu.ac.uk

***Abstract*—Understanding wealth distribution and the earning potential for different social groups is a vital tool for governments looking to create equal opportunity societies. This study looks to create a classifier model to help guide such decisions through assessment of the UCI Adult Salary Dataset which details demographics of US citizens and their current salary band. A decision tree classifier with a Gini split quality and maximum node number pruning strategies was found to be the most effective model, producing a classification accuracy of 85.3%.**

***Keywords—machine learning, classification, logistic regression, decision tree***

## I. Introduction

Understanding national pay structure data and the underlying demographic distributions can provide valuable insights to societal issues. In recent times the UK has used such data to drive policies addressing the gender [1], [2] and the ethnicity pay gaps [3], [4], and to highlight potential routes out of poverty [5], with similar trends seen around the world [6]–[8]. Processing the vast amounts of data required to guide these decisions is however a costly and time-consuming endeavour.

Automated data processing and machine learning methods are invaluable tools in the modern world for analysing and making use of large data sets. Such tools allow for efficient data cleaning to reduce the time required preparing the data for analysis and the generation of classifiers producing predictive results from previously unseen data. Used in conjunction with pay structure data these tools allow for analysis of key factors leading to pay scales for particular demographics and the prediction of likely pay for new demographic blends.

This paper looks to utilise Python 3 coding language coupled with the scikit learn machine learning library [9] to demonstrate data exploration and machine learning techniques for a national pay structure dataset. The data used will be the Adult Salary Dataset taken from the open-source UCI machine learning library [10], which provides salary band classification for a sample of US citizens along with key demographic figures. This mix of data is an ideal blend for generating a predictive model to determine the salary of an individual from their demographic attributes and will allow us to demonstrate universal techniques that can be applied to similar data sets from other countries.

## II. Data Exploration

The UCI Adult Salary Dataset comprises of a subset of data from the 1994 US census database, with each entry representing a surveyed individual. The dataset contains 14 multivariate attributes with various descriptive data and one output label variable indicating whether the individuals salary was more or less than $50k per year. The data contains 48842 individual entries and is separated into training and testing sets. For the purposes of this paper we have recombined the data into one set for initial exploration and cleaning before later re-splitting into train/test sets for final machine learning.

Summary statistics for the data are generated using the Python 3 'describe' command along with the skew and kurtosis of the continuous data variables. Full statistics of the samples are displayed in "Table I" with additional details commented on below. Also noted is that a lot of the variables contain either extreme outlier or low frequency outlier valuers, these are described and addressed in the later Data Cleaning and Feature Engineering section.

### A. Numerical Data

5 of the 14 attribute variables can be classed as numerical as identified in "Table I". The numerical data can also be analysed for normality using the generated summary statistics. Normality statistical tests can be readily performed in Python but these tests are typically designed for small data sets and are too sensitive to handle large data distributions, often spuriously producing negative results [11]. Instead we use the general rule of thumb for data that a distribution with skewness > ±2, or a kurtosis > ±3 should be treated differently to a normal distribution [12]. With this rule of thumb we are free to treat the 'age' and 'hours-per-week' attributes as normal distributions, but as all other variables require caution due to non-normality we shall use more robust methods for all if manipulation is required.

The 'capital-gain' and capital-loss' attributes also appear to require special attention. As can be seen both attributes have very high kurtosis values and the median values lie significantly lower than the mean values. Visual assessment of the distribution of these attributes show that both have very high numbers of entries at 0 value followed by a distribution of data at higher values as in "Fig 1", known as sparse data, which can cause issues with outlier removal using mathematical methods and create issues with some classifier types such as k-Nearest Neighbour [13]. One potential solve is to reclassify the attributes as binary where entries have or don't have any capital gain/loss, however this would still leave a sparse dataset.

One final observation on the numerical data is the magnitude of the units which varies significantly between the attributes, in particular the largest magnitude ('fnlwgt') is 4 orders of magnitude larger than the lowest

TABLE I: SUMMARY STATISTICS OF THE UCI ADULT SALARY DATASET ATTRIBUTES

| Numerical Data | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Variable** | **Data Type** | **Summary Statistics** | | | | | | | |
| | | ***Count*** | ***Mean*** | ***Median*** | ***Standard Deviation*** | ***Min*** | ***Max*** | ***Skew*** | ***Kurtosis*** |
| age | continuous | 48842 | 38.6 | 37.0 | 13.7 | 17 | 90 | 0.6 | -0.2 |
| fnlwgt | continuous | 48842 | 189664.1 | 178145.0 | 105604.0 | 12285 | 1490400 | 1.4 | 6.1 |
| capital-gain | continuous | 48842 | 1079.1 | 0.0 | 7452.0 | 0 | 99999 | 11.9 | 152.7 |
| capital-loss | continuous | 48842 | 87.5 | 0.0 | 403.0 | 0 | 4356 | 4.6 | 20.0 |
| hours-per-week | continuous | 48842 | 40.4 | 40.0 | 12.4 | 1 | 99 | 0.2 | 3.0 |

| Categorical Data | | | | | |
|---|---|---|---|---|---|
| **Variable** | **Data Type** | **Summary Statistics** | | | |
| | | **Count** | **Unique** | **Modal Class** | **Modal Count** |
| workclass | nominal | 46043 | 8 | Private | 33906 |
| education | ordinal | 48842 | 16 | HS-grad | 15784 |
| education-num | ordinal | 48842 | 16 | 9 | 15784 |
| marital-status | nominal | 48842 | 7 | Married-civ-spouse | 22379 |
| occupation | nominal | 46033 | 14 | Prof-specialty | 6172 |
| relationship | nominal | 48842 | 6 | Husband | 19716 |
| race | nominal | 48842 | 5 | White | 41762 |
| sex | binary | 48842 | 2 | Male | 32650 |
| native-country | nominal | 47985 | 41 | United-States | 43832 |
| salary | binary | 48842 | 2 | <=50K | 37155 |

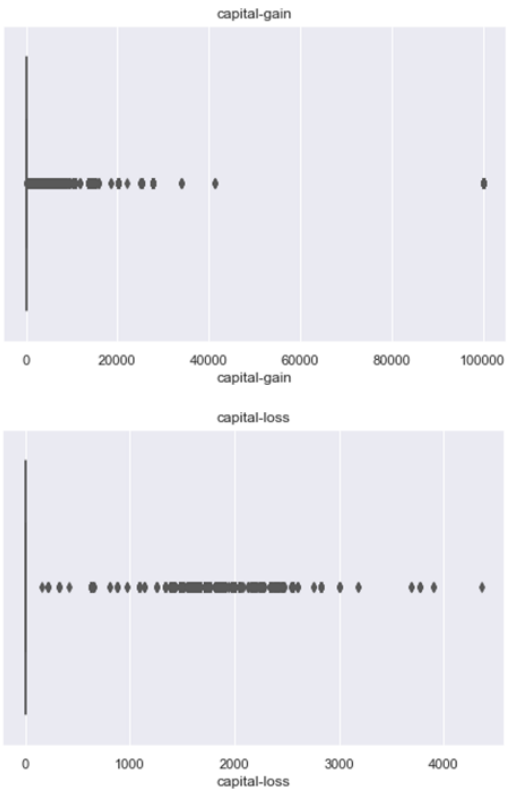


Fig 1. Box and whisker plots of the 'capital-gain' and capital-loss' attributes from the UCI Adult Salary Dataset. Note that the majority of the data is zero value with anything else classed as an extreme outlier.

('age' / 'hours-per-week'). Such large swings in magnitude can cause issues for machine learning algorithms that rely on the distance between data points, in particular k-Nearest Neighbour and clustering methods, and will require scaling before such models are applied.

### *B. Categorical Data*

9 of the 14 attribute variables can be described as categorical data as shown in "TABLE I". Categorical data is assessed for modal values and the frequency of the modal class to assess the spread of the data. One initial observation is that 3 of the attributes ('workclass', 'occupation' and native-country) appear to be incomplete in terms of data, this will be addressed in the Data Cleaning and Feature Engineering section later in this paper.

On assessment of the data the 'education' and 'education-num' attributes appear to contain the exact same information simply relabelled, with the 'education-num' providing an ordinal score for the 'education' attribute. This allows us to drop the 'education' attribute for modelling purposes as it is simply repeated information and contains less information than 'education-num' which also contains the ordinal ranking.

We can also observe 2 attributes which have very dominant modal features with >80% of the data with lots of other potential classes ('race' and 'native-country'). Attributes such as this are prime targets for minor class merging to eliminate noise in the data and will again be investigated as such in the later Data Cleaning and Feature Engineering section.

A final note on the categorical data is on the processing ability in our chosen coding language Python. Python is

unable to utilise raw categorical data for most machine learning algorithms barring those based on decision trees. To eliminate this issue before algorithm training all categorical features can be split into binary dummy variables which can be accepted by the code (one-hot encoding [14]). This processing however will largely increase the sparsity of the data, providing a further disadvantage for algorithms which do not work well with data of this type.

## III. Literature Review and Model Selection

For our purposes the 'salary' attribute will be used as the target for machine learning. This attribute is a binomial type with an approximate 3:1 split in class distribution. This leaves the data an ideal candidate for classification type machine learning algorithms to build a predictive model. This section looks to review work already done in this field to narrow down a list of potential algorithms for use.

The Adult Salary Dataset is widely used for machine learning algorithm demonstration and has many references in both formal academic paper and blog form. "TABLE II" summarises the results of a literature search identifying 5 models and a baseline strategy for potential use with advantages, disadvantages and examples of their use either directly on the Adult Income dataset or very similar salary classification data. With this the table also provides the average classification accuracy found for each classifier from all the sources, alongside accuracy and training time figures from an initial run of each classifier with the raw training data and no parameter tuning in Python.

### *A. Rejected Models*

3 of the 5 selected models have been rejected for further consideration as explained below.

**k-Nearest Neighbour** was rejected despite its reasonable accuracy score within the found literature [15], [16] due to its difficulty in dealing with sparse datasets. As we will be utilising Python for our research and require one-hot encoding to reduce the nominal attributes to multiple binary attributes, this will increase the sparsity of our data drastically which will limit the usefulness of this algorithm.

**Naïve Bayes** is a model which provides many of the same advantages as linear models such as logistic regression in terms of being easy to interpret, quick to train and requiring minimal data pre-processing. It comes however with reduced accuracy due to only relying on attributes in isolation (i.e. no attribute interaction) for class prediction. During our literature search we found several references comparing machine learning models directly on the adult dataset [17]–[19] which all found the k-Nearest Neighbour algorithm to be significantly worse in accuracy than other models. For this reason we will reject the model and consider potentially higher accuracy algorithms.

**MultiLayer Perceptron (MLP)** classifiers come with a very high level of tunability and can be used to produce very accurate classifiers. Our literature search could only find one example with the UCI Adult Dataset [18] but this

TABLE II: LITERATURE REVIEW SUMMARY FOR CLASSIFIERS USED WITH THE UCI ADULT SALARY DATASET

| **Classifier** | ***Advantages*** | ***Disadvantages*** | ***Examples*** | ***Average Accuracy in Literature, %*** | ***Initial Trial Accuracy, %*** | ***Initial Trial Training Time, s*** |
|---|---|---|---|---|---|---|
| Zero-R | - default baseline accuracy model | - reports all data as the modal class | N/A | N/A | 76.1 | 0.055 |
| k-Nearest Neighbour | - easy to understand<br>- fast to train<br>- can handle multiclass datasets<br>- few training hyperparameters<br>- good with numerical data | - poor with large or many featured datasets<br>- requires data pre-processing<br>- poor with sparse datasets (lots of zero values) | [15], [16] | 79.0 | 81.7 | 0.926 |
| Logistic Regression | - fast to train and predict<br>- works with large and sparse datasets<br>- can extend to multiclass datasets (default is binary)<br>- few training hyperparameters | - requires data preprocessing | [17], [20], [21] | 85.2 | 84.5 | 0.255 |
| Naïve-Bayes | - easy to understand<br>- fast to train (fastest of algorithms listed)<br>- can handle multiclass datasets<br>- few training hyperparameters | - compromised accuracy as decisions are based on attributes in isolation | [17]–[19] | 79.2 | 81.3 | 0.107 |
| Decision Tree | - very easy to understand<br>- minimal data preprocessing required<br>- good with a mix of categorical and numerical data<br>- can handle non-linearly seperable data | - easy to overfit the data | [17]–[20], [24] | 84.6 | 81.0 | 0.522 |
| MultiLayer Perceptron | - highly tunable<br>- can handle non-linearly seperable data | - very slow to train and predict<br>- poor explainability of the model<br>- requires data preprocessing | [18] | 86.3 | 84.4 | 47.549 |

did produce the highest accuracy of all the searched classifiers. The tunability and accuracy of this classifier however comes at a high price of training time, as seen in "TABLE II" the initial test trial results show that the MLP algorithm took >50x longer to train, which would add significant time delays during tuning experiments. Additionally as MLP models are so complex they are very difficult to break down to see where the classification decisions are being made. This can cause serious issues if such a model would be used to guide governmental policies as suggested as the lack of transparency would mean important decisions would have to be made in an uniformed manner. For these reasons we have decided to reject this model for study.

### B. Selected Models

From our paper research 2 models have been selected for further consideration with justification outlined below.

**Logistic Regression** is a linear classification model which is very fast to train and simple to understand [13]. Our literature review and initial testing showed it to be within the top 2 models for accuracy [17], [20], [21] as well as being the second fastest to implement (not including Zero-R baselining). Due to its speed of training and simple implementation it will provide an ideal modelled baseline classifier for our adult dataset. For this algorithm there are 2 main hyperparameters for tuning as follows:

1. *The C value* (often referred to as alpha in other linear models) – controlling the magnitude of the weightings on the model, lower C values create simpler models and a balance between over/underfitting needs to be found [22]. Typically C values are searched for on a logarithmic scale during experimentation.
2. *L1/L2 regularisation* - L1 denoting least squares method of regularisation of the dataset attributes, while L2 utilises a lasso method [23]. L2 regularisation will potentially lead to an easier to understand model but may cut out useful features if all are of similar importance to the classifier, hence both can be trialled to assess for any differences in accuracy.

**Decision Tree (DT)** classifier solutions were the most common found during our literature search [17]–[20], [24]. They carry a key advantage in that they are very easy to understand particularly for non-data science professionals, allowing for an easy visualisation of the classifier (providing the model is relatively small). They also require minimal data pre-processing as they allow for many data type attributes and large data ranges within the same model with no accuracy penalty. On the downside however decision trees are very prone to overfitting data and great care needs to be taken to eliminate over complex models which do not work on unseen generalised data. Additionally out initial test run with the DT algorithm produced the lowest classification accuracy of the modelled algorithms despite our literature search showing an accuracy comparable to logistic regression and MLP classifiers. This indicates that there are potential large gains to be made through tuning of the classifier with the data. For this reason coupled with the easy explainability of the model we have decided to pursue this algorithm for further study.

As DT algorithms are so prone to overfitting many of the parameters are designed to combat this, utilising different 'pruning' strategies. We shall focus on the 3 main pruning strategies along with the split quality metric which are identified as the features that can provide the most tuning efficacy [13], as follows:

1. *Max_depth* – representing the maximum number of node layers the decision tree can create, larger depth increases the complexity of the tree and can lead to overfitting. From our literature search we have found values of 5 – 10 used for this hyperparameter [17], [18] and we shall focus on a similar range.
2. *Max_leaf_nodes* – represents the maximum number of nodes the decision tree can create in total, higher numbers can lead to overfitting of the data as more data separation steps are used. We shall explore a very wide range of maximum leaf nodes (up to 200) and monitor the effects on the model accuracy.
3. *Min_samples_leaf* – represents the minimum number of samples each leaf can contain within the decision tree. Low numbers can lead to overfitting of the data as the model creates complex structures which cater to very small numbers of samples within the training data. We shall explore a very wide range of minimum samples per leaf node (up to 200) and monitor the effects on the model accuracy.
4. *Gini/Entropy split quality* – denoting the split method utilised by the tree to either maximise purity (Gini) or information gain (entropy). Both are valid methods that yield slightly different results so shall both be investigated for our dataset.

## IV. EXPERIMENTAL METHODOLOGY

### A. Experimental Set Up

Due to the speed of running algorithms in Python we will be able to test a very wide range of parameters simultaneously on out chosen classifiers. For the logistic regression classifier we shall test a wide range of C values for both the L1 and L2 regularisation methods. For the DT algorithm we shall test each of the pruning strategies over a wide range of values for both the Gini and entropy splitting strategies, before comparing the pruning strategies to devise the best one.

To train each algorithm we shall split the data into training and test sets using a 65:35 % split strategy. The training set shall be used to train each of the parameter tuning steps where a 5-fold cross validation methodology will be used to test the efficacy of each hyperparameter tuning step. Due to the random nature of the data and the incremental gains seen when tuning the hyperparameters a graphical method shall be used to choose the optimum values (statistical methods will struggle to identify

significance when high granularity of data is used), accuracy scores shall be plotted before running a smoothing algorithm to select the best value. The test set shall be reserved for a final test using all the tuned algorithms for comparison of results, along with a standard Zero-R baseline strategy where a simple model predicting all data as the most common class is used [25]. This method will allow us to perform a final comparison test on unseen data for all the tuned algorithms whilst still allowing us to test performance on data not used for training during the hyperparameter tuning stage.

Throughout training we shall focus on the classification efficacy of the models as the primary indicator of performance to allow focus on only one metric. Other metrics will also be explored during the final algorithm comparison.

### *B. Data Cleaning and Feature Engineering*

During experimentation and model tuning with the data, cleaning and feature engineering was also implemented to solve the data noise issues identified in section II, remove duplicate data, tune parameters to better fit the selected models and reduce model training times. As with the hyperparameter tuning experimentation this was done using an iterative methodology looking for gains in the accuracy and training times of the classifiers. A summary of the final data modifications is presented below along with accuracy and training time of our baseline logistic regression classifier with default settings in the Scikit learn package. Any attributes which are not listed were not modified. In addition to the modifications mentioned all numerical values were normalised using the min/max method [26] to reset the values between 0 and 1 avoiding any scaling issues between variables, and all categorical data was one-hot encoded [14] to allow running through scikit learn. Also noted is that all data modification was done using the pipeline methodology in Python [27] and tuned on the training dataset before applying to the test set to avoid bias in the data.

#### *1) Missing Data Values*

As identified earlier three of the attributes within the dataset contained missing values:

- The **'workclass'** and '**native-country'** missing values were replaced by the modal values of 'private' and 'United-States' respectively which accounted for >70% of the data in both attributes
- The **'occupation'** attribute showed several classes of similar frequency (~12%), as we were unable to assign a new class without biasing the data missing value entries were removed from the dataset

Once the missing data values were dealt with the baseline logistic regression classifier check returned an accuracy of **84.7%** and a training time of **1.52s**.

#### *2) Numerical outlier data*

All of our numerical fields were classed as having outlier data, which was dealt with in the following way:

- The **'age'** and **'fnlwgt'** attributes had outliers removed using the robust box and whisker method where any data more than 1.5x the inter quartile range above the 75 percentile or below the 25 percentile were removed [28].
- The **'capital-gain'** and '**capital-loss'** attributes were treated in a similar method but had the zero-value data removed first before calculating the outliers to ensure that the calculation was not influenced by the sparsity of the data. Once the outliers had been removed the zero-value data was returned to the dataset.
- The **'hours-per-week'** data was reclassified into an ordinal attribute rather than simply removing the outliers. This was done as most of the data could be seen to represent people working a typical western work pattern of ~40 hours per week. Deeper analysis revealed that a significantly higher portion of the longer hour workers were earning >50k than the rest of the dataset. To ensure this data wasn't lost by removing instances of outliers for people working outside these hours, the data was instead reclassified as 'part-time' (<35 hours), 'full-time' (35-48 hours) and 'long-hours' (>48h), with these classifications guided by the Rones et al. 1997 study into working hour patterns [29].

These modifications to the data gave us a new baseline accuracy of **84.9%** and a training time of **1.51s**.

#### *3) Categorical data*

Categorical features were handled by combining outliers into other categories where appropriate, or by removing them altogether as follows:

- The **'native-country'** attribute was simply reclassified as 'United-States' or 'Other' due to the overwhelming majority (>90%) of the data belonging to the 'United-States' class.
- Any 'without-pay' class data was removed from the '**workclass'** attribute as the class only accounted for a tiny portion of the data (<0.05%) and didn't make sense to be included in a dataset looking at classifications for paid work.
- The **'educational-num'** attribute was reclassified to condense the multitude of classes available. This was guided by the work from Tamborini et al which assessed earning potential based on education level in the US [30]. New classifications with corresponding ordinal rank were as follows: post-graduate (5), bachelors (4), some-college (3), high-school (2), less-than-high-school (1).
- From the **'occupation'** attribute the 'Priv-house-serv' and 'Armed-Forces' classes were removed as they accounted for <<1% of data, potentially meaning they should be considered separately.
- In the **'marital-status'** attribute the 'married-AF-spouse' and 'married-civ-spouse' classes were combined into one 'married' class as both referred to married individuals and this negated the noise created by the very small 'married-AF-spouse' class
- The 'Asian-Pac-Islander' and 'Amer-Indian-Eskimo' classes were combined into the 'Other' class in the **'race'** attribute, to reduce the amount of noise from

smaller populated classes in an attribute where >90% of the data was contained in the two most dominant classes

These modifications to the data gave us a new baseline accuracy of **84.1%** and a training time of **0.42s**.

Overall data cleaning and feature engineering has slightly reduced the accuracy of our baseline classifier by 0.7%, but we have also seen a training time reduction of 72%. This was deemed acceptable as the modifications should give us a more generalised model and allow for speedy processing if future larger datasets were used.

## V. Results and Discussion

Each of our selected algorithms shall be discussed separately before a final comparison between the optimised algorithms is undertaken.

### A. *Logistic Regression*

"Figure 2" and "TABLE III" show the results for the logistic regression classifier experiments testing a range of C values using both the L1 (least squares) and L2 (lasso) regularisation methods. From the data we can see that both regularisation methods produce very similar results for both low ($C \leq 10^{-4}$) and high ($C \geq 10^{-2}$) complexity, with the L2 performing better for the values in between. As the regularisation looks to limit the strength of the least influential attributes we can see that the lasso method produces superior results when assigning attributes to be dampened. Also observed is that low complexity models produce an accuracy score very close to the Zero-R baseline indicating that they are most likely simply classifying the whole dataset as the majority class.

The best results from the model for both regularisation techniques is seen with higher complexity models with a plateau reached in the data for C values $\geq 10^{-1}$. For our champion model we shall choose a C value of 1 with L2 regularisation as this produced the joint highest accuracy with the lowest complexity and training time, allowing for less risk of over fitting and speedy processing of larger datasets.

Table III: Full experimental results from hyperparameter tuning of a logistic regression classifier applied to the UC Adult Salary Dataset.

| C | L1 regularisation | | L2 regularisation | |
|---|---|---|---|---|
| | *Accuracy, %* | *Time, s* | *Accuracy, %* | *Time, s* |
| $10^{-6}$ | 0.754 | 0.380 | 0.754 | 0.371 |
| $10^{-5}$ | 0.754 | 0.415 | 0.754 | 0.417 |
| $10^{-4}$ | 0.754 | 0.385 | 0.754 | 0.418 |
| $10^{-3}$ | 0.754 | 0.552 | 0.803 | 0.450 |
| $10^{-2}$ | 0.839 | 0.867 | 0.838 | 0.539 |
| $10^{-1}$ | 0.847 | 2.499 | 0.845 | 0.702 |
| $10^{0}$ | 0.846 | 2.924 | 0.847 | 0.812 |
| $10^{1}$ | 0.846 | 2.900 | 0.846 | 0.896 |

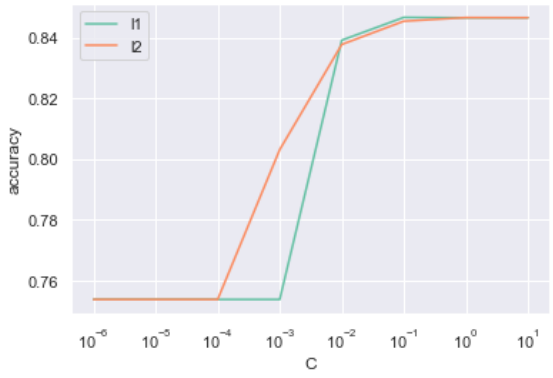


Figure 2: Experimental results from hyperparameter tuning of a logistic regression classifier applied to the UC Adult Salary Dataset depicting the classifier accuracy vs the C values.

### B. *Decision Tree Classifier*

"Figure 3:" shows the results from the Decision Tree classifier experiments testing the three pruning methods with both the Gini and Entropy node splitting strategies. As the data set is very large for these experiments only the champion data from each set is presented in "TABLE IV", note that 2 of the pruning parameters are presented twice as they had different champion results for the 2 splitting strategies.

For the max tree depth and max leaf node splitting strategies we can see that accuracy rapidly drops off for simplified models (low numbers) indicating underfitting, with a less rapid drop off as the model becomes more complex. The opposite is true for the min samples per leaf strategy which sees a rapid drop off for the higher complexity (low number) models.

The max tree depth and min samples per leaf strategies both show marginal differences between the Gini and entropy split quality strategies. While the max leaf node strategy shows a more significant difference, particularly as it nears optimum accuracy. The entropy splitting strategy tends to penalise mixed leaf nodes more harshly than Gini, indicating that this pruning strategy is most likely providing a better split for the data than the other two, which is reflected in the comparative maximum accuracy scores.

Our optimum model for the decision tree algorithm was using the Gini splitting strategy coupled with the max leaf node strategy with the value set to 96, which provided the optimum classification accuracy for the lowest processing time. Note that combinations of different pruning strategies were also tested but these only served to reduce the classification accuracy of the model compared to when the strategies were used individually.

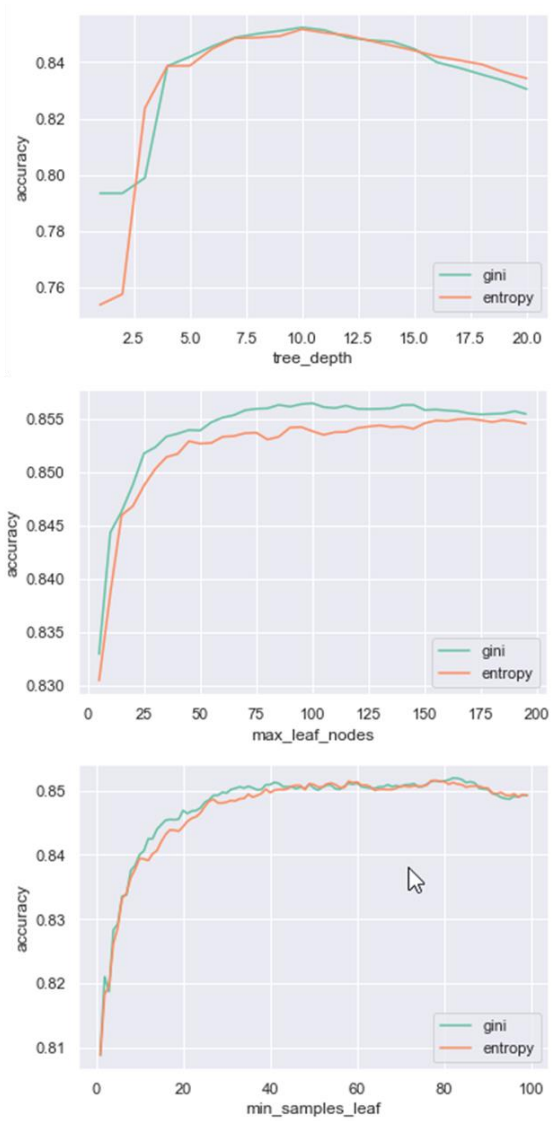


Figure 3: Plots of the experimental data tuning the 3 different pruning strategies (top – tree depth, middle – max leaf node, bottom – min samples per leaf) for a decision tree classifier applied to the UCI Adult Salary Dataset.

Table IV: Optimum pruning strategy hyperparameter values for a decision tree classifier applied to the UCI Adult Salary Dataset

| Parameter | Value | Gini | | Entropy | |
|---|---|---|---|---|---|
| | | *Accuracy, %* | *Time, s* | *Accuracy, %* | *Time, s* |
| Max depth | 10 | 0.852 | 1.013 | 0.852 | 0.826 |
| Max leaf nodes | 96 | 0.856 | 0.752 | 0.854 | 0.909 |
| Max leaf nodes | 190 | 0.856 | 0.801 | 0.855 | 0.985 |
| Min Samples leaf | 82 | 0.852 | 0.724 | 0.851 | 0.874 |
| Min Samples leaf | 77 | 0.851 | 0.720 | 0.852 | 0.817 |

### C. *Final Model Comparison*

Final model comparison data was generated by running the trained classifiers on the previously unseen reserved test data. Results of this final experiment are shown in "TABLE V" along with the corresponding results from a Zero-R baseline classifier.

TABLE V: CLASSIFICATION DATA REPORT FOR THE FINAL TUNED CLASSIFIERS ON THE UCI ADULT SALARY DATASET

| Metric | Class | *Zero-R* | *Logistic Regression (L2 regularisation, C value = 1.0)* | *Decision Tree (Gini splitting, max leaf node = 92)* |
|---|---|---|---|---|
| Accuracy | N/A | 0.755 | 0.842 | 0.853 |
| Precision | <50k | 0.755 | 0.877 | 0.881 |
| Recall | | 1.000 | 0.921 | 0.931 |
| Precision | >50k | 0.000 | 0.710 | 0.741 |
| Recall | | 0.000 | 0.596 | 0.611 |

From the results we can see that the classification accuracy has dipped slightly (~0.5%) for both models, this is expected with unseen data and the reason models were kept as general as possible during training. Another observation is that the precision and recall for both models is far lower for the >50k class data than the <50k class, in particular the recall for both is quite poor (~60%) indicating that the models tend to over classify as <50k for unseen data. Depending on the question asked of the data this could be an important issue, for example if we wish to look at which demographic factors increase the likelihood of a high >50k salary we may miss some key features if too much data is wrongly classified.

Both classifiers perform significantly better than the Zero-R baseline strategy model. Statistical testing using a 1-tail hypothesis test shows that the 1.1% difference in accuracy between the models is significant at the 95% confidence level (p-value << 0.05) allowing us to conclude that the decision tree algorithm does indeed have a superior classification accuracy. The decision tree algorithm also showed significantly better recall and precision accuracies for the less classified >50k category indicating its higher probability of relevant recall for this lower populated class.

## VI. CONCLUSIONS AND FURTHER WORK

In conclusion we have demonstrated a robust method for training of a machine learning algorithm for demographic data predicting salary bands. A decision tree classifier algorithm was selected for its speed of processing and explainability to a wide audience. Pruning method and splitting strategy optimization returned a classification accuracy of 85.3% on previously unseen data, beating the performance of other applications of this classifier on the UCI Adult Salary Dataset found during a literature review.

Future work around this dataset would look at a breakdown of the factors which lead to higher salaries for individuals. Decision trees are ideal for this as diagrams of the model process are simple to interpret. With our model

however we have seen that there are up to 92 potential node splits in the data which may still be difficult to interpret important factors from. Instead a more robust method such as SHAP value analysis [31] on the data may provide more transparent insights to the key demographic features.

Additionally we may also explore more complicated decision tree models on the data such as random forest classifiers which utilize an ensemble of trained decision trees [32] or gradient boosted trees which utilize an ensemble of weak learner single split trees [33]. However it is noted both of these methods will increase training time and reduce explainability of any resulting models.


[1] 'Gender pay gap reporting', *GOV.UK*. https://www.gov.uk/government/collections/gender-pay-gap-reporting (accessed Apr. 14, 2021).

[2] 'Why equal pay matters | Equality and Human Rights Commission'. https://www.equalityhumanrights.com/en/advice-and-guidance/why-equal-pay-matters (accessed Apr. 14, 2021).

[3] Simonetta Longhi and Malcolm Brynin, 'The ethnicity pay gap.', Institute for Social and Economic Research, University of Essex, 108, 2017. Accessed: Apr. 14, 2021. [Online]. Available: https://nls.ldls.org.uk/welcome.html?ark:/81055/vdc_100060606741.0x000001.

[4] 'Ethnicity pay reporting', *GOV.UK*. https://www.gov.uk/government/consultations/ethnicity-pay-reporting (accessed Apr. 14, 2021).

[5] Peter Kemp, Jonathan Bradshaw, Paul Dornan, Naomi Finch, and Emese Mayhew, 'Routes out of poverty', Joseph Rowntree Foundation, Nov. 2004. Accessed: Dec. 17, 2020. [Online]. Available: https://www.jrf.org.uk/report/routes-out-poverty.

[6] Daniel Ornstein and Jordan Glassberg, 'Countries Implement New Gender Pay Gap Measures', *The National Law Review*, Nov. 27, 2018. https://www.natlawreview.com/article/countries-implement-new-gender-pay-gap-measures (accessed Apr. 14, 2021).

[7] CloudPay, 'A Guide to Pay Parity Laws Around the World'. https://www.cloudpay.net/resources/a-guide-to-pay-parity-laws-around-the-world (accessed Apr. 14, 2021).

[8] 'Equity in Education: Breaking Down Barriers to Social Mobility | en | OECD'. http://www.oecd.org/education/equity-in-education-9789264073234-en.htm (accessed Dec. 17, 2020).

[9] 'scikit-learn: machine learning in Python — scikit-learn 0.24.1 documentation'. https://scikit-learn.org/stable/ (accessed Apr. 14, 2021).

[10] 'UCI Machine Learning Repository: Adult Data Set'. https://archive.ics.uci.edu/ml/datasets/adult (accessed Apr. 14, 2021).

[11] G. Gigerenzer, 'Mindless statistics', *The Journal of Socio-Economics*, vol. 33, no. 5, pp. 587–606, Nov. 2004, doi: 10.1016/j.socec.2004.09.033.

[12] P. Westfall and K. S. S. Henning, 'Skewness and Kurtosis', in *Understanding Advanced Statistical Methods*, CRC Press, 2013, pp. 248–259.

[13] Andreas Muller and Sarah Guido, 'Supervised Learning', in *Introduction to Machine Learning with Python*, O'Reilly Media, 2016.

[14] J. Brownlee, 'Why One-Hot Encode Data in Machine Learning?', *Machine Learning Mastery*, Jul. 27, 2017. https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/ (accessed May 05, 2021).

[15] M. Muqorobin, K. Kusrini, S. Rokhmah, and I. Muslihah, 'Estimation System For Late Payment Of School Tuition Fees', *International Journal of Computer and Information System (IJCIS)*, vol. 1, no. 1, Art. no. 1, May 2020, doi: 10.29040/ijcis.v1i1.5.

[16] 'Income data using KNN'. https://kaggle.com/abhijeetjainmca/income-data-using-knn (accessed Apr. 22, 2021).

[17] U. Gunturi, 'Adult Income Prediction using Python', *Medium*, Dec. 13, 2019. https://medium.com/@umagunturi789/adult-income-prediction-using-python-1d716f700b4e (accessed Apr. 22, 2021).

[18] R. Khanna, 'Comparative Study of Classifiers in predicting the Income Range of a person from a census data', *Medium*, Dec. 10, 2018. https://towardsdatascience.com/comparative-study-of-classifiers-in-predicting-the-income-range-of-a-person-from-a-census-data-96ce60ee5a10 (accessed Apr. 22, 2021).

[19] S M Bramesh and B S Puttaswamy, 'Comparative Study of Machine Learning Algorithms on Census Income DataSet', *Journal of Engineering Research and Application*, vol. 9, no. 8, pp. 78–81, Aug. 2019.

[20] C. Perlich, F. Provost, and J. S. Simonoff, 'Tree Induction vs. Logistic Regression: A Learning-Curve Analysis', *Journal of Machine Learning Research*, vol. 4, pp. 211–255, Mar. 2003.

[21] A. Agarwal, 'Logistic Regression classifier on Census Income Data', *Medium*, Apr. 20, 2019. https://towardsdatascience.com/logistic-regression-classifier-on-census-income-data-e1dbef0b5738 (accessed Apr. 22, 2021).

[22] 'sklearn.linear_model.LogisticRegression — scikit-learn 0.24.1 documentation'. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html (accessed Apr. 22, 2021).

[23] Aditya .P, 'L1 and L2 Regularization.', *Medium*, Nov. 11, 2018. https://medium.com/@aditya97p/l1-and-l2-regularization-237438a9caa6 (accessed Apr. 22, 2021).

[24] S. M. Bekena, 'Using decision tree classifier to predict income levels', Jul. 30, 2017. https://mpra.ub.uni-muenchen.de/83406/ (accessed Apr. 12, 2021).

[25] J. Brownlee, 'A Gentle Introduction to Imbalanced Classification', *Machine Learning Mastery*, Dec. 22, 2019. https://machinelearningmastery.com/what-is-imbalanced-classification/ (accessed Dec. 21, 2020).

[26] 'sklearn.preprocessing.MinMaxScaler — scikit-learn 0.24.2 documentation'. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html (accessed May 05, 2021).

[27] 'sklearn.pipeline.Pipeline — scikit-learn 0.24.1 documentation'. https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html (accessed Apr. 26, 2021).

[28] C. Thirumalai, M. Vignesh, and R. Balaji, 'Data analysis using box and whisker plot for lung cancer', in *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, Apr. 2017, pp. 1–6, doi: 10.1109/IPACT.2017.8245071.

[29] P. L. Rones, R. E. Ilg, and J. M. Gardner, 'Trends in Hours of Work since the Mid-1970s', *Monthly Lab. Rev.*, vol. 120, p. 3, 1997.

[30] C. R. Tamborini, C. Kim, and A. Sakamoto, 'Education and Lifetime Earnings in the United States', *Demography*, vol. 52, no. 4, pp. 1383–1407, Aug. 2015, doi: 10.1007/s13524-015-0407-0.

[31] S. Lundberg and S.-I. Lee, 'A Unified Approach to Interpreting Model Predictions', *arXiv:1705.07874 [cs, stat]*, Nov. 2017, Accessed: May 07, 2021. [Online]. Available: http://arxiv.org/abs/1705.07874.

[32] J. VanderPlas, 'In Depth: Decision Trees and Random Forests', in *Python Data Science Handbook: Essential Tools for Working with Data*, O'Reilly Media, Inc., 2016, pp. 421–432.

[33] B. Gardin, *Gradient Boosting Trees: A Beginner's Guide For Gradient Boosting: Machine Learning For Dummies*. Independently Published, 2021.