ADULT DATA CLASSIFICATION DATA SET DISCUSSION

Exploratory Data Analysis:

Part Two data required a lot more data analysis than part one, with most of the details of it being included in the notebook containing the code (graphs and prints of correlation). One of the major pre-processing tasks that needed to happen for this data was cleaning it up in the first place. It had missing values (?), it had string values that needed to be converted to binary details and it also had missing column names. Due to the small percentage of missing values I decided to simply remove the instances with missing data instead of attempting to impute the data, finding through online research that this would still provide me with a strong model. There was also an outlier in capital gain with a value of 99999 which I removed as it seemed unlikely to be a genuine data point given the number (99999) and the next highest value being ~20000.

Before breaking up some of the string features into binary ones I first checked the unique values in the features. For some, such as marital status, there were entries which could be combined to give a more straight forward but still detailed feature and less binary splits when that happened to try and reduce the dependence in the data. Some values such as married-af- and married-civ- were combined in order to simply have a married feature. I did this with education as well, combining all the levels pre-high school graduate into school leaver. Once the data had been cleaned up I split it into binary values, increasing the number of features to 91.

Due to the still relatively small number of features (<100) I decided not to use PCA analysis on this data set in order to keep all the features within the model. This decision was made also in part due to the difficulty of applying the PCA and getting the results from the training set to apply to the test set as well, due to the files being provided separately. The dimension was also not so large that it would cause problems, though there was undeniable dependence between the features based on the binary data split (e.g. Married 1 implies Single 0 for all instances where married is 1).

The general data analysis also showed that some features, such as native country, had a very low impact on the final class while others, like education num, had a high impact and a clear division between the education num itself and the class assigned to that instance. There was also an interesting indicator that could be ethically important, with women belonging more to the low income class than the high one. There were also indicators that higher values of all the numerical features were related to a higher income level. Occupation was also a strong indicator of class, showing a much more even distribution in managerial roles than handlers-cleaners for example.

After the exploration, I standardised the data and ensured that it was ready for the algorithms to run on.

Results of Algorithms:

CLASSIFICATION ALGORITHM	ACCURACY	PRECISION	RECALL	F1-SCORE	AUC
KNN	0.78	0.78	0.78	0.72	0.57

Naïve Bayes	0.26	0.66	0.26	0.13	0.50
SVM	0.75	0.74	0.75	0.74	0.64
Decision Tree	0.75	0.76	0.75	0.76	0.68
Random Forest	0.82	0.80	0.82	0.80	0.68
AdaBoost	0.83	0.82	0.83	0.81	0.69
Gradient Boosting	0.84	0.83	0.84	0.82	0.71
Linear Discriminant Analysis	0.78	0.78	0.78	0.78	0.69
Multi-Layer Perceptron	0.73	0.74	0.73	0.73	0.65
Logistic Regression	0.68	0.71	0.68	0.69	0.62

Note, all classifications were run with a 5-fold cross validation for their scores/metrics. The Precision/Recall/F1-score are all taken as the average / total, not the individual class values

Accuracy

Accuracy is not a good indicator of performance for classification methods. Whenever there is unbalanced data in a data set, i.e. data is not divided evenly across the classes, accuracy becomes in a sense useless. This is because it can easily be tricked into polling high numbers and claiming a large accuracy when the opposite is true. This can be best shown when there is a dataset which may have 90% of class A and only 10% of class B. If the predictor chooses every time for the class to be A, 90% of the time it will be correct, but the number of false positives will be extremely high, while true negatives will be 0. This affects metrics such as precision, recall and specificity, and shows that while accuracy can be useful in cases with balanced data especially in unbalanced data it is one of the least useful metrics.

Best Algorithms and Why.

Ranking of Algorithms

Classification Algorithm	Accuracy	Precision	Recall	F1-Score	AUC
KNN	4	4	4	8	9
Naïve Bayes	10	10	10	10	10
SVM	6	7	6	6	7
Decision Tree	6	6	6	5	4
Random Forest	3	3	3	3	4
AdaBoost	**2**	**2**	**2**	**2**	**2**

Gradient Boosting	**1**	**1**	**1**	**1**	**1**
Linear Discriminant Analysis	4	4	4	4	3
Multi-layer Perceptron	8	7	8	7	6
Logistic Regression	9	9	9	9	9

Across the board the best two algorithms according to the performance metrics were AdaBoost and Gradient Boosting algorithm. They performed the best along each of the metrics, with Gradient Boosting outperforming AdaBoost, while the rest of the algorithms lagged behind and many had different ranks depending on the performance metric they were being run through. Naïve Bayes was the lowest of the algorithms by quite a stretch, the variables very clearly not conditionally independent as it needs to assume to run correctly so having an impact on the metrics for this dataset.

Gradient Boosting algorithm performed the best across the board. It is based on the idea of building an additive model in a forward fashion, pulling in features that are useful as it defines the model to minimise the loss function. In the default settings the loss function is the deviance, which is reduced with the creation of n class numbers of regression tree, in this case one tree added in each stage. This is in a sense a directed boost of decision trees, which are aiming towards the goal of classifying all the instances correctly by building on the mistakes of the prior generation.
Through the first pass of the gradient boosting algorithm a tree is built which due to the low number of >50K class instances will have a low recall, precision and f1 score accordingly, though with each pass / iteration a new regression tree will be boosted from the previous generation and begin to balance out the smaller number of instances by increasing the emphasis put on the smaller classes to increase precision, recall, f1 and overall accuracy, which will boost the ROC curve based on the true positive rate as well.

AdaBoost performed well across the board as it is based on the idea of fitting a classifier on original data before fitting copies of that classifier on the same dataset with the weights of the incorrectly classified instances adjusting so the next classifier will focus more on classifying these difficult cases. This worked well on this dataset as the lack of balance between the classes led to a higher emphasis initially on classifying the instances as the more prominent of the two classes, giving it a high accuracy but decreasing the precision and recall, which impacts directly on the F1 score.
As the classifier continues to run the AdaBoost algorithm will increase the importance of classifying the less prominent class, which would increase the precision and recall, along with the F1 score. This leads to a higher accuracy as the classes are more likely to be correctly identified, increasing the true positive rate and decreasing the false positive rate, along with increasing specificity. As the true positive rate became higher and more accurate the AUC will also increase in size, as the threshold of classification adjusts and the ROC curve adapts accordingly.