A comparative analysis of Naïve Bayes and K-Nearest Neighbors on the Adult Dataset

Motivation & Description of the data

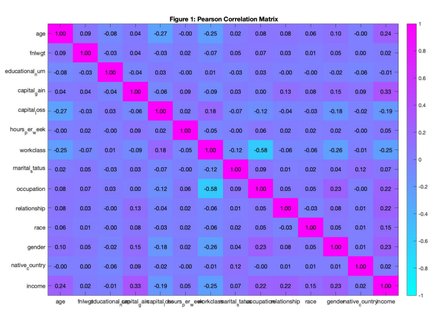
- The problem is to predict whether the individuals earn more than 50K as an income-based on different attributes. Thus, this is a binary classification problem.
- For this analysis, we chose to apply Naïve Bayes and K-Nearest Neighbours algorithms. The aim is to compare and contrast the performance of those methods and understand how they learn from the chosen dataset.

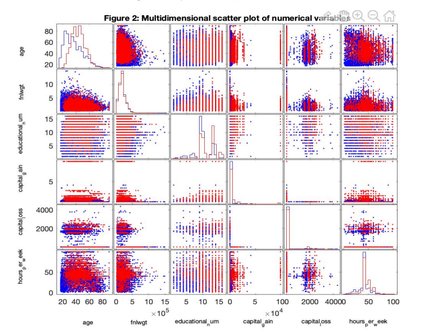
Initial analysis of the data & basic statistics

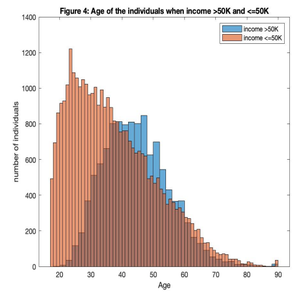
- **Dataset:** Adult from the UCI Repository
- The dataset contains 48,842 rows before cleaning and after cleaning 45,222.
- The dataset has a total of 15 features, one of which is the target variable 'income' with 2 classes: '>50K' and '<=50K'. Among those 15 features, 9 are categorical, and 6 are numerical (continuous).
- From the Pearson Correlation plot (figure 1), we can see that the linear relation between our numerical predictors is not strong. This is confirmed by the Multidimensional scatter plot (figure 2) which does not show a distinctive correlation between the numerical features. The strongest correlation is between 'income' and 'capital_gain' with a value of 0.33. Moreover, we can see a rather strong negative correlation between the 'occupation' and 'workclass' features, with a value of -0.58.
- According to figure 3, there is a class imbalance in the dataset. 75% of the individuals in the census earn an income '>50K' and 25% earn '<=50K'.
- The capital loss and gain features have many '0' values and as can we can see in the boxplots (figure 7 and 8), they have many extreme outliers.

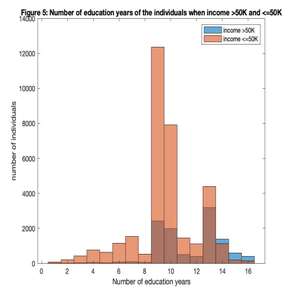
Features \ Statistics	>50K			<=50K		
	Mean	Standard dev.	Skewness	Mean	Standard dev.	Skewness
Age	44.00	10.00	0.49	36.75	13.56	0.73
Education_num	11.60	2.00	-0.32	9.63	2.42	-0.43
Capital gain	3991.80	14616.00	5.83	149.02	927.43	17.05
Capital loss	193.50	593.00	2.80	54.03	312.22	5.91
Hours per week	45.70	11.00	0.85	39.37	11.97	0.31

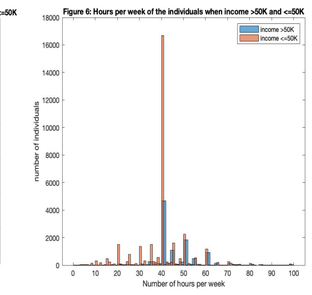
Table 1: Descriptive statistics

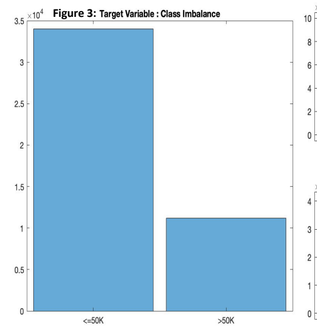


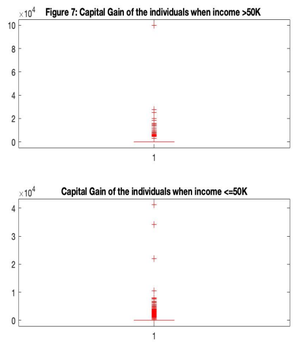


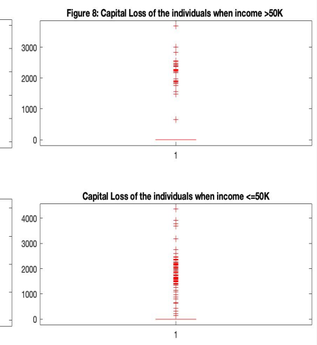












Summary of the two ML models

K-Nearest Neighbors

- K-NN is an example of an instance-based algorithm. Instance-based algorithms use instances seen during the training period and stored in the memory to perform the prediction/classification task.
- Predictions are made for new data by searching the original dataset for similar instances (neighbours).
- K-NN is also an example of a non-parametric and lazy learning algorithm. It is called lazy as it does not learn from a discriminative function but just memorises the training dataset instead. [3]

Pros [3]

- Easy to understand and implement (there is no need to train a model for generalisation)
- The training phase of this model is usually much faster than the one of other classification methods
- Performs well on a dataset that has multiple class labels

Cons [6]

- Selection of k value is tricky and application dependent
- The testing phase of KNN classification is slower and costlier in terms of time and memory
- Requires scaling of data because this algorithm uses the Euclidean distance between two data points to find the nearest neighbours
- Euclidean distance is sensitive to the magnitude
- The features with high magnitude will weight more than the features with low magnitude
- Having a high number of features can be costly/not feasible memory and computation-wise (limited)

Naïve Bayes

- Naïve Bayes is a supervised machine learning method, which based on Bayes Theorem, that can be used to classify data into two or more classes. This means that this classifier is trained to analyse a dataset for which the classes are labelled. [6]
- Naïve Bayes is an example of a Generative algorithm. It is based on the joint probability of the input data and the target. This classifier assumes that the impact of a specific feature in a class is independent of other features. Even if the features of the dataset are dependent, they will still be separately considered. This theory makes the computation of this method simpler, thus the name 'naïve'. [5]

Pros [5]

- Minimal error rate
- Simple, fast, and highly accurate method for prediction
- Works efficiently on large datasets
- Performs well in case of discrete variable compared to a continuous variable
- Can be used in the case of multiple class prediction problems

Cons [5]

- The model has an assumption of independent features (it usually is almost impossible that a model will have variables which are fully independent)
- It assigns zero probability to categorical values it has not seen

Hypothesis statement

- Similarly, to the work presented by Halgamuge (2017) [2], we expect both models to perform relatively well.
- KNN is expected to perform better than Naïve Bayes and therefore, to have a lower generalisation error.
- Also, we expected the NB model to have a shorter running time than K-NN.
- Moreover, we expected that the best 'Distance' hyperparameter for K-NN is going to be one of the following: the 'Manhattan', 'Minkowski', 'Chebychev', 'Euclidean' or 'Mahalanobis'. [4]

Choice of parameters & experimental results

K-Nearest Neighbors

Parameters

- We used Bayesian Optimisation to optimise two hyperparameters: the number of neighbors and the distance.
- Through the Bayesian Optimisation we found that the hyperparameters for the best accuracy (83.77%) are 'NumNeighbors' = 23 and 'Distance' = 'cityblock'.

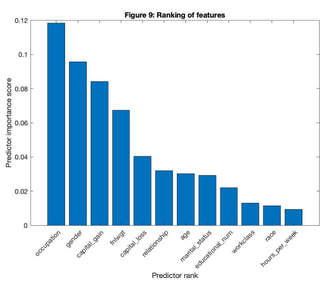
Naïve Bayes

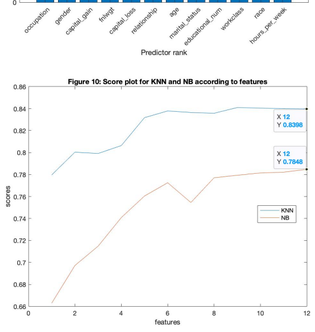
Parameters

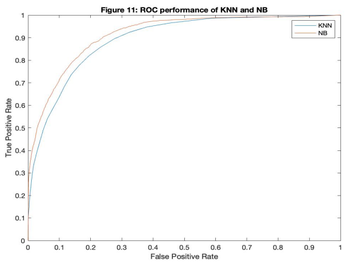
- We used Bayesian Optimisation to optimise three hyperparameters: the distribution names, width and kernel.
- Bayesian Optimisation showed that the best estimated feasible points use the following hyperparameters: 'DistributionNames' = 'kernel', 'Width' = 0.001554 and 'Kernel' = 'triangle'. This gave us an accuracy of 80.24%.

Experimental Results

- As stated in the methodology section, we have used MRMR for feature selection. Figure 9 shows the ranking of the predictors, where the top 5 features are 'occupation', 'gender', 'capital_gain', 'fnlwgt', and 'capital_loss'.
- In the Score plot (figure 10), we can see that KNN reaches its optimal accuracy (83.98%) at the 12th feature ('hours_per_week'). Its score plot spiked after adding the 4th feature ('educational_num').
- NB reaches its optimal accuracy at the 12th feature ('hours_per_week'). Its score has been rising at an almost constant rate between the 1st and 6th ('occupation') features, then drops around the 7th ('relationship') to then go back up and reach its highest accuracy of 78.48%.
- NB appears to perform better than KNN according to their ROC curves (figure 11).
- The AUC for NB performs better than K-NN's, for both its train and tests. This shows that the generalisation error for Naïve Bayes is lower than the one of the K-Nearest Neighbors.







KNN				NB		
Train	Validation	Test		Train	Validation	Test
0.9114	-	0.8931	AUC	0.928	-	0.915
0.8508	0.8373	0.8377	Accuracy	0.8072	0.8226	0.8024
0.9037	-	0.8951	F1-score	0.8586	-	0.8551
0.9312	-	0.9208	Precision	0.778	-	0.7755
0.7004	-	0.6926	Recall	0.5852	-	0.5833
0.2559	-	0.2911	Fallout	0.4292	-	0.4353
-	0.1627	-	Loss	-	0.1774	-

Description of the choice of training & evaluation method [5]

- Import the dataset and perform cleaning and pre-processing on it.
- Then, to perform the feature selection we started by ranking our predictors using the minimum redundancy maximum relevance (MRMR) algorithm. We perform a feature selection analysis on both models by adding a feature every iteration on the validation set.
- Partitioning the data into a 70% training and 30% test sets using 'cvpartition' .
- We normalised the dataset and used 10-kfold cross-validation was used to estimate the generalisation error.
- In order to get the optimum values for the two chosen methods, we used hyperparameter tuning (using Bayesian Optimisation and trial-and-error, as Grid Search would be more time consuming).
- The evaluation criteria used are ROC-AUC, accuracy, F1-score, precision, recall and fallout.

Analysis & critical evaluation of results

- As expected, K-NN performed better than Naïve Bayes with an accuracy of 83.77% compared to 80.24%. However, since we have an imbalanced data, accuracy might not be the best criteria to compare our models. In fact, when the data is imbalanced, accuracy cannot differentiate between the number of correctly classified classes.
- Better measures are F1-score, precision and recall. For K-NN, the F1-score is higher in the test set than the one of Naïve Bayes, with a value of 89.51% compared to 85.51%.
- Both models have recall scores above 50%, with 69.26% for K-NN and 58.33% for NB.
- Running the train/validation/test for NB took 335.251 seconds while K-NN took 27.223 seconds. This is the opposite of what we expected before doing this analysis. The Bayesian optimisation for K-NN took 613.050s and 3670.9346s for NB. The kernel function might be causing the model to run slower, making therefore K-NN more efficient to run.
- From the feature analysis, we have decided that we would not be removing any features during the analysis as there are not many of them and the models are improved almost every time, we added a feature.
- The results can be improved but we would require more studies in order to select the predictors and improve the hyperparameters.
- Kernel density estimation seems to perform better than the normal approximation when tuning the hyperparameters for NB. This can be because our predictors do not follow a Gaussian distribution. One downside is that kernel requires the tuning of more hyperparameters making the model more complex.
- Despite the Euclidean distance being the most popular and contrarily to our hypotheses about the best 'Distance' hyperparameter supported by Prasath et al.'s work [4], Bayesian optimisation found that the 'Distance' to use for an optimal prediction is 'cityblock'. The 'cityblock' distance is a special case of 'Minkowski' distance. [1]

Future Work & Recommendations

- Both the KNN and Naïve Bayes models require some hyperparameters tuning in order to improve their prediction accuracy. If done well, this can lead to drastic improvements in the predictions.
- K-NN requires a lot of memory and is time-costly. It would have been preferable to choose a smaller dataset or to apply it on a portion of the current dataset.
- In the future, we could try experimenting with different machine learning algorithms and look at their error/accuracy.
- For Naïve Bayes, we can also look at other hyperparameters and their impact on the model's generalised error.
- We could create more data to correct the imbalance in the dataset using the Synthetic Minority Over-Sampling Technique (SMOTE)
- When looking at the capital gain/loss features, we could consider using a Random Forest algorithm in the future as this algorithm can reduce the impact of outliers on the analysis.
- We could look at this problem as a regression problem using the same methods.

References

[1] Chujai, P. and Kerdprasop, N., 2015. *An Empirical Study Of Distance Metrics For K-Nearest Neighbor Algorithm*. [online] ResearchGate. Available at: <https://www.researchgate.net/publication/299847267_An_Empirical_Study_of_Distance_Metrics_for_k-Nearest_Neighbor_Algorithm> [Accessed 1 December 2020].

[2] Halgamuge, M., 2017. *Impact Of Different Data Types On Classifier Performance Of Random Forest, Naïve Bayes, And K-Nearest Neighbors Algorithms*. [online] ResearchGate. Available at: <https://www.researchgate.net/profile/Malka_Halgamuge/publication/322248829_Impact_of_Different_Data_Types_on_Classifier_Performance_of_Random_Forest_Naive_Bayes_and_K-Nearest_Neighbors_Algorithms/links/5bc97a00458515f7d9c969e8/Impact-of-Different-Data-Types-on-Classifier-Performance-of-Random-Forest-Naive-Bayes-and-K-Nearest-Neighbors-Algorithms.pdf?origin=publication_detail> [Accessed 20 November 2020].

[3] Jadhav, S. and Channe, H., 2016. *Comparative Study Of K-NN, Naive Bayes And Decision Tree Classification Techniques*. [online] Ijsr.net. Available at: <https://www.ijsr.net/archive/v5i1/NOV153131.pdf> [Accessed 1 December 2020].

[4] Prasatha, V., Abu Alfeilat, H., Hassanat, A., Lasassmeh, O., Tarawneh, A., Alhasanat, M. and Salman, H., 2019. *Effects Of Distance Measure Choice On KNN Classifier Performance - A Review*. [online] Arxiv.org. Available at: <https://arxiv.org/pdf/1708.04321.pdf> [Accessed 1 December 2020].

[5] Rahangdale, G., Ahirwar, M. and Motwani, D., 2016. *Application Of K-NN And Naïve Bayes Algorithm In Banking And Insurance Domain*. [online] Ijcsi.org. Available at: <https://ijcsi.org/papers/IJCSI-13-5-69-75.pdf> [Accessed 22 November 2020].

[6] Safri, Y., Arifudin, R. and Muslim, M., 2018. *K-Nearest Neighbor And Naive Bayes Classifier Algorithm In Determining The Classification Of Healthy Card Indonesia Giving To The Poor*. [online] Pdfs.semanticscholar.org. Available at: <https://pdfs.semanticscholar.org/207c/538045241b2d1d048696f26bcbfc0b90996d.pdf> [Accessed 1 December 2020].