A method performance comparison on annual wage prediction

MA429 - Mock Project

Jessica Kärrberg (201664554), Ba Dat Nguyen (201931420),
Shuai Zhang (201947952)

Department of Mathematics
London School of Economics
England
April 10, 2020

Executive Summary

This project provides an analysis of the performance of different statistical data prediction techniques applied to the census dataset from the UCI Machine Learning Repository. The choice of methods are Support Vector Machines (SVM), K-Nearest Neighbours (KNN) and Naive Bayes (NB). Based on past papers studying the same dataset, the hypothesis is that the SVM method will have the best performance. The performance metrics was chosen to be classification accuracy, precision and recall, calculated from the confusion matrix.

A preliminary analysis found that missing values in the dataset must be handled, and numeric variables should be scaled. Furthermore, it was hypothesized that no numeric variables would be removed in an attribute selection due to the low correlation between numeric variables.

In the data processing, the missing values in the dataset were found not to be missing at completely random, and therefore a data imputation was done. Moreover, the numeric variables were scaled using a min-max scaling. Finally, the attributes which were found to be less important using random forest method were dropped from the dataset.

Thus, all the models that are mentioned are constructed, trained and tested on the processed data. To conclude, the project shows that SVM has the best performance on the census data, compared to the KNN and NB, according to the classification metrics accuracy, precision and recall. However, one main drawback of the SVM is its runtime. Hence, it depends on the requirements and priorities of the end user which prediction method to use.

Contents

1 Introduction

Census data is in the U.S. gathered by the U.S. Census Bureau once a decade. One of the data's main purposes is to determine how to allocate the House of Representatives among the states (Bureau n.d.). In this report, however, the data will be used to try predict if an individual has an annual wage above or below $50.000 using only the attributes recorded for the same census data.

1.1 Background and Aims

The aim of this report is to compare the performance of different prediction methods when applied to the census dataset from the UCI Machine Learning Repository, further described in section 2. The prediction will be on whether an individual has an annual wage above or below $50k.

The methods that will be tested in this study are Support Vector Machines (SVM), K-Nearest Neighbours (KNN) and Naïve Bayes (NB). These methods are well-studied on this dataset, see for example Caruana and Niculescu-Mizil 2004 and Kohavi 1996. Based on these papers, the hypothesis for this project is that the SVM method will have the best performance.

Before the methods are applied a preliminary analysis and data processing phase is performed to understand the data further and appropriately prepare the data for the learning methods. Thereafter the methods are applied and evaluated, after which a conclusion is made.

1.2 Performance metrics

In this project, confusion matrix will be uesed to derive the performance measures used on each model, see Table 1.

		Actual	
		True	False
Predicted	True	TP	FP
	False	FN	TN

Table 1: *The table displays a confusion matrix and its abbreviated quantities*

More specifically, the models' performance will be measured based on the following three statistics:

- **Classification accuracy**, the number of correctly classified instances:

$$ACC = \frac{TP + TN}{TP + TN + FN + FP}$$

- **Precision**, the number of correct positive predictions among all positive predictions:

$$PPV = \frac{TP}{TP + FP}$$

- **Recall**, how often there is a positive prediction when it is actually positive:

$$TRP = \frac{TP}{TP + FN}$$

A confusion matrix is used because it can give us a more complete picture of how these models are performing in different aspects. Normally the choice of metric depends on the data and objective. For example, if the cost of a False Positive (FP) or a False Negative (FN) is high, the performance in precision or recall, respectively, should be the preferred measure. In this project, however, a more complete performance measure is requested, therefore several classification metrics are used.

2 Dataset introduction & Preliminary analysis

The dataset used in this project is the Adult dataset from the UCI Machine Learning Repository, which can be found by clicking here. The dataset contains census data from the 1994 U.S Census database, with 6 numerical and 8 nominal attributes. For details, see Table 2.

Attribute	**Type**	**Range**	**Description**
age	Numeric	17-90	Age of person
workclass	Nominal	8 categories	The type industry a person works in
fnlwgt	Numeric	13769 - 1484705	Measure used by the Population Division at the Cencus Bureau. People with similar demographic characteristics should have similar weights
education	Nominal	16 categories	Educational level ranging from 'Preschool' to 'Doctorate'
education-num	Numeric	1-16	Number of years in school
marital-status	Nominal	7 categories	Person's marital status
occupation	Nominal	14 categories	Type of work person does
relationship	Nominal	6 categories	A person's household relationship status
race	Nominal	5 categories	Person's race
sex	Nominal	2 categories	Gender of person
capital-gain	Numeric	0-99999	Profit earned from sold assets
capital-loss	Numeric	0-4356	Loss incurred form sold assets
hrspw	Numeric	1-99	Number of hours worked per week
native	Nominal	41 categories	A person's origin
wage-class	Nominal	2 categories	Describes if person makes more or less than 50k per year

Table 2: *The table displays and describes all attributes present in the original dataset.*

The dataset has a total of 32561 observations with 2399 rows missing one or more values. For further details on how incomplete cases are handled, see subsection 3.1

From Table 2 it is evident the range of the numeric variables vary greatly, see for example the variables *fnlwgt* and *age* or variables *capital-gain* and *hrspw*. As this can influence the methods applied in a later stage, especially the ones that are dependent on a Euclidean distance,

a data scaling is called for. For details on this, see section 3.2.

By plotting each attribute in a histogram, the variety within each variable was graphically examined, see Appendix 1 - Preliminary Analysis for plots. From these plots it can be concluded the variables *capital-gain*, *capital-loss*, *work-class*, *hrspw* and *native* all have one dominant category, i.e. most datapoints carry one specific nominal value in this attribute. In other variables, the distribution between nominal or numeric values is more even. The attribute *age* seems to have a truncated bell-curve, which we would expect as the range starts at 16. The interpretation of this is an indication of representativeness in the sample.

To further understand the dataset and its attributes, the correlation between the numerical variables was examined using Pearson's R-value. It showed no correlation worth mentioning, see plot in Appendix 1 - Preliminary Analysis. Due to this it is reasonable to believe the attribute selection, see subsection 3.3, will not remove any or few numeric attributes.

Class	<=50	>50
Percentage	75.1%	24.9%

Table 3: *The table displays the distribution of instances in the wage-class variable*

It can be seen from the Table 3 there is an imbalance in the response variable *wage_class*, where the group of instances having an annual wage lower than 50k is approximately three times as large as the one having an annual wage above 50k. Looking at Statista, the median annual income of American households in 1994 was approximately 52k (Statista n.d.). Since some of the households must have consisted of two or more earners, it is reasonable to believe the median of individual earners must have been below 52k per year. As the dataset in this report consists of individual earners, it is therefore assumed the distribution of wage classes in the dataset is representative and case-control sampling deemed to be unnecessary.

3 Data Processing

In this section what is done to prepare the dataset before applying the prediction methods is described. As previously mentioned in section 2, the missing values in the dataset must be handled and the numeric variables must be normalized. In addition to this, an attribute selection is done to remove any multicollinear variables or variables that have low contribution to the prediction.

Note that before moving on to these steps, the response variable, *wage_class*, is transformed from a factor to a binary variable. Also, after the step of handling the missing values and transforming the data, the dataset is divided into training and test set with proportions 70% and 30%. The attribute selection is not done on the test set as this would risk introducing bias.

3.1 Incomplete cases

First the missing attributes for all observation of the data set are investigated.

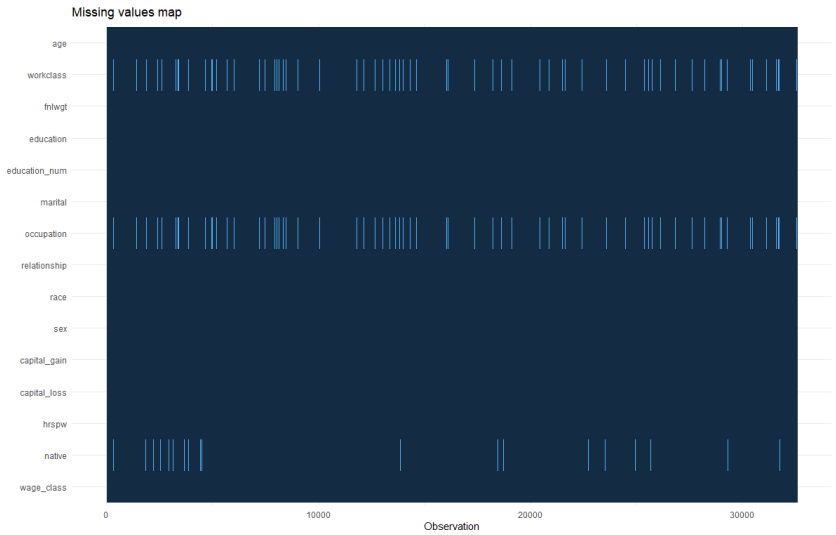

Figure 1: *The figure displays the missing attributes for all observation.*

From Figure 1 it can be seen that most missing data are in the attributes of *workclass*, *occupation* and *native*. To add, it is obvious from the figure that there is strong correlation between missing data for both *workclass* and *occupation*. Hence, one could argue that these two attributes are similar, so excluding one or the another should not affect the final outcome.

Next, the question raises whether it is appropriate to delete the incomplete cases or not. The first issue is that there are 2399 incomplete cases out of 32561 observations, which translates to 7.368% missing data of all observations. Assuming that the incomplete cases are MCAR, it can be argued that deleting these observations should not affect the final analysis.

Hence, the issue is whether the missing observations are MCAR or not. From the library called *BaylorEdPsych* the function *LittleMCAR* is used on the data set. The null hypothesis is that the data is MCAR. The results show significant p-value (< 0.001) meaning there is enough evidence to reject the null hypothesis, hence the missing values are not MCAR. Thus, the correlation between missing and observed values has to be examined.

Plotting all attributes against each other makes the graph more challenging to read due to the number of variables, see Appendix 2 - Missing Relation 1. The missing values in the numerical features seemed to have no correlation to the observed counterparts, they looked more completely random, while in the categorical attributes tend to have more missing values for specific values. Hence, another figure is constructed just for the categorical variables see Appendix 2 - Missing Relation 2.

Several observations can be made from the figure, see Appendix 2 - Missing Relation 2, for instance it can be seen that people with income less than $\leq 50K$ tend not to tell their *occupation* and *workclass*. Also, compared to other marital status, people who are widowed have the most missing *occupation* and *workclass*. Furthermore, in terms of sex, female participants have more missing values than male participants. For missing *native_country* values, the only significant relation we can observe is that people with Asian-Pac-Islander or Other tend to have missing values more frequently. Furthermore, missing values for for the category "Never-

worked" in the attribute *workclass* can be observed, which is expected for this group.

Since the removal of incomplete observations is not an option, missing entries are replaced with data imputation. The choice of data imputation method is going to be k-Nearest Neighbour. The reason this method is chosen because it can to predict the missing data values from the values of the same predictor in the points k nearest neighbours of the data point. In this situation the variable k is set to 5, by default. Also, since there are several factor attributes (not numeric or integer) using regression, mean/median or univariate sampling would be less useful. However, kNN is more versatile, since it can be used with any kind of data type so using here is appropriate. We are using the function *kNN* from the package *VIM* for the purpose.

3.2 Data Transformation

As seen in section 2, Dataset introduction & Preliminary analysis, the ranges in the numeric variables differ a lot. Therefore, a data transformation is required.

The chosen method to transform the data is min-max scaling, see equation 1, which brings down all values to the range [0,1]. The reason this method is chosen is because most numeric variables are measured in different units, and therefore this scaling makes them comparable. The method is applied to all numeric variables, except the variable *education_num*. The reason for this is the variable is actually categorical and not continuous.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

3.3 Attribute Selection

Firstly, the *varimp* function in the *randomForest* package is used to get a rank of variable importance. This function is based on the mean decrease in accuracy. The outcome shows that the variable *fnlwgt* is the attribute with the least importance, which receives a much smaller score compared to other variables' importance.

To try and confirm the result of the variable ranking method mentioned above, correlation was also used. In Figure 5, it can be seen that the attributes are neatly uncorrelated. However, when the correlation between *fnlwgt* and the *wage-class* is considered, in Figure 2, it is not hard to see that *fnlwgt* exhibits little variation with wage class, which means the two wage classes have similar *fnlwgt* values and therefore this attribute should be removed. This result is cohesive with the previous conclusion using the random forest method.

In addition, the correlation between the variables *education* and *education-num* is examined, see Figure 3. It can be seen from the picture there is a perfect correlation and hence only *education-num* is kept in the dataset. In the ranking output from the random forest method, the variables *education* and *education-num* had similar scores, which makes sense as they add the same information.

To conclude, the attributes *fnlwgt* and *education* are removed from the dataset.

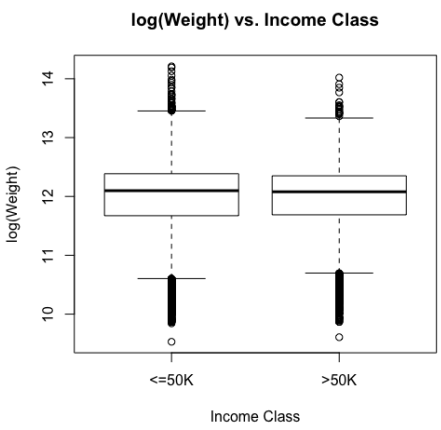

Figure 2: *Boxplot of the variable fnlwgt divided based on wage-class*

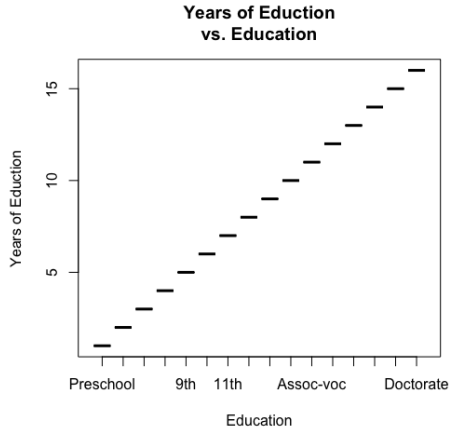

Figure 3: *Education and Education-num have perfect correlation.*

4 Experiments with data mining methods

In this section the data mining methods presented in subsection 1.1 are applied to the finalised dataset. The choice of parameters used in each model is also evaluated and discussed.

4.1 K-Nearest Neighbours

K-Nearest Neighbours (KNN) is one of the most common methods for classification problems. Consider a test instance $(x^0, y^0) \in \text{Te}$, the k nearest points to x^0 in Tr is denoted by $N_k(x^0)$. Based on the points in $N_k(x^0)$ a probability distribution of the response classes is returned. Let $(y_1 \dots y_k)$ be the possible class values, then the prediction for $\Pr(Y = y_j | X = x^0)$ is (James et al. 2013)

$$p_j(x^0) = \frac{1}{k} |\{(x, y) \in N_k(x^0) : y = y_j\}|$$

and the distance of two data points with p attributes, say $x = (x_1 \dots x_p)$ and $z = (z_1 \dots z_p)$, is measured by the standard Euclidean distance (Hastie, Tibshirani, and Friedman 2009)

$$d(x, y) = \sqrt{\sum_{i=1}^{p} (x_i - z_i)^2}$$

In order to use this method, all categorical variables must be converted to numeric variables and then normalised. By letting k=1,...,20 the graph over all performance metrics in Figure 4 is obtained. The runtime of creating this KNN model is approximately 2 minutes.

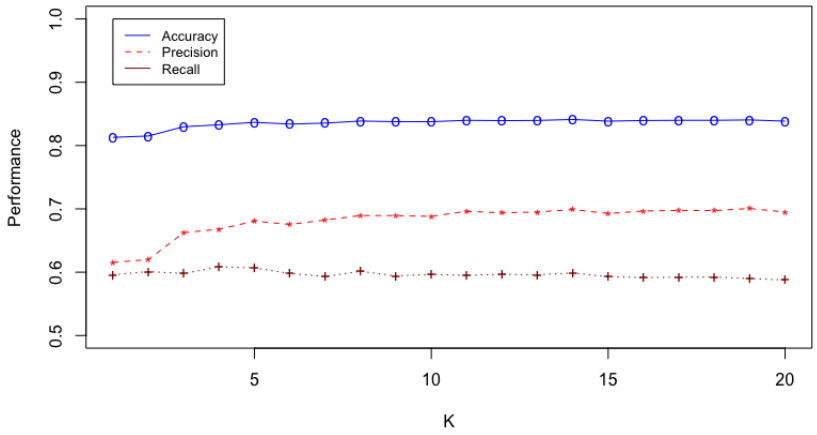

Figure 4: *The figure displays the performance of KNN in the selected performance metrics when k = 1,...,20.*

It can be seen that the values of the various metrics are becoming relatively stable as k increases. But if the graph in the figure is magnified, see for example accuracy in Appendix 3 - KNN, a range in which KNN has relatively better performance can be selected and studied. However, this maximum performance range does not coincide for the various metrics. Therefore, a new metric aggregating the metrics with equal weights is created to reflect a "total performance" for each k. Based on this new metric, the optimal k is selected, see Figure 6. As can be seen, when $k = 14$, this new metric gives its best performance, see Table 4.

Accuracy	Precision	Recall
0.8412	0.6989	0.5988

Table 4: *The table displays the performance of the KNN model*

4.2 Support Vector Machines

A Support Vector Machine (SVM) is a generalization of a support vector classifier. The support vector classifier seeks to divide observations with a hyperplane with "soft" or violatable margins at a minimal cost. The difference is SVMs enlarge the feature space by using kernels, allowing non-linear boundaries between classes. The kernel is a generalisation of an inner product, see equation 2 for notation. This type of kernel is called a linear kernel (James et al. 2013).

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j} \tag{2}$$

The kernels that will be considered in this analysis are linear, radial and polynomial. The form of the radial and polynomial kernels can be seen in equations 3 and 4, respectively. What kernel to use in the SVM depends on the distribution of the two classes in the data (James et al. 2013).

$$K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2) \tag{3}$$

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d \tag{4}$$

If the dataset only consisted of two attributes and one response variable it would be easy to plot the data to judge an appropriate kernel. As this is not the case in this study, a plotting method is not viable. To select a kernel, a 20 % random sample is instead taken and removed from the training data to create three SVM models, each with a linear, radial and polynomial kernel. These models are created using the function *tune.svm* in the *e1071* package. The function uses a grid-search algorithm to select the optimal parameters in terms of lowest error generated from a model with the inputted parameter values. For linear kernels the *cost* parameter is tuned on the values $(0.01, 0.1, 1, 10)$, for radial kernel the *cost* parameter is tuned on the same values and the *gamma* parameter on the values $(0.01, 0.5, 1, 2)$. Finally, the polynomial kernels is tuned in the *cost* parameter with the same values, the *degree* tuned in range $[3, 5]$ and the *coef0* parameter, which allows adjustment of the independent term in equation 4, is tuned on the values $(0, 0.1, 0.5, 1, 2, 3, 4)$.

After the tuning is executed, each best model is selected from the output. The models' performance are measured on the training set in terms of correct classifications. This performance is considered an indication of kernel fit on the data, and consequently the kernel used in the model with the lowest error is then used in creating the final model from the remaining training data.

Kernel:	Linear	Radial	Polynomial
Error:	15.05%	14.68%	14.85%

Table 5: *The table displays the error rates for the three models with different kernels created from the smaller sample of training data, tested on the training data*

As the results displayed in Table 5 indicate a radial model is just slightly better than a polynomial model, a radial SVM model is created using the same *tune.svm*-function. Like previously, the tuning is done on the parameters *cost* and *gamma*, but to save in on computational power the tuning is only done in the range $\{1, 10\}$ for the *cost* parameter and $\{0.01, 0.5\}$ for the *gamma* parameter. These ranges were chosen as they are the closest values to the optimal ones chosen in the initial round when comparing the kernels. This time, however, the model is created from the remaining training data and tested on the test data.

The runtime for tuning and creating the model is 58.35 minutes and the optimal parameters found by the function is *cost*= 10 and *gamma*= 0.01. The larger value of the *cost* parameter means a smaller decision margin is accepted for a better classification of the training points and the smaller value of the *gamma* parameter gives a less smooth shape of the decision boundary. The performance of this model is summarized in Table 6.

Accuracy	Precision	Recall
0.8524	0.9381	0.8762

Table 6: *The table displays the performance of the final SVM model when applied to the test data*

Worth noting is that the final model picks the same parameters as the initial model. Therefore,

it is reasonable to believe runtime can be saved by using the parameters selected for the initial model and skip the tuning for the final model.

4.3 Naïve Bayes

The *Naïve Bayes* method is a simple parametric classifier method. It is based on *Bayes's theorem* which says the following:

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}, \tag{5}$$

where E and H represents the "evidence" and "hypothesis", respectively. The conditional probability $P(H|E)$ is the probability of the hypothesis assuming we know that E is true. However, this equation holds for a single predictor variable, in general there are several predictors and several classes for the model. So for this study, let the features be represented by $X_1, X_2, X_3, ..., X_n$, where $n = 13$, and the classes be Y_0, Y_1. Y_1 represents the individuals with an annual income of at least $50k$ and Y_0 otherwise. By substituting it into Equation 5 the following result is obtained:

$$P(Y_i|X_1, X_2, X_3, ..., X_n) = \frac{P(X_1, X_2, X_3, ..., X_n|Y_i) * P(Y_i)}{P(X_1, X_2, X_3, ..., X_n)}, \tag{6}$$

for $i \in \{0, 1\}$ and $n = 13$. An important assumption is that the predictors are independent from each other, which was found to be the case in subsection 3.3. Hence by this assumption the following equation can be written (Narasimha et al. 2015):

$$P(X_1, X_2, X_3, ..., X_n|Y_i) = P(X_1|Y_i) * P(X_2|Y_i) * ... * P(X_n|Y_i) \tag{7}$$

Thus, now the values for the dependent variable in Equation 6 can be predicted from the dataset with estimates for these conditional probabilities.

After implementing the model using function *naive_bayes* from the package *naivebayes*, which has a runtime of approximately 1 minute, the confusion matrix in Table 7 is obtained for this method. Also, result for the performance metrics are shown in Table 8.

	False	True
False	6992	423
True	1313	1040

Table 7: *The confusion matrix for the Naive Bayes method.*

Accuracy	Precision	Recall
0.8222	0.7104	0.4412

Table 8: *The table displays the performance of the Naive Bayes method*

4.4 Summary of results

As can be seen in Table 9, the SVM method performs best relative to KNN and NB in all performance metrics. However, the SVM performs worst in terms of runtime, which is almost one hour for constructing the final model alone. The KNN takes approximately 2 minutes and the NB requires less than 1 minute.

Method	Accuracy	Precision	Recall
KNN	0.8412	0.6989	0.5988
SVM	**0.8524**	**0.9381**	**0.8762**
NB	0.8222	0.7104	0.4412

Table 9: *The table displays the performance of all applied methods*

5 Analysis

In this section the results found in Section 4 are analysed and compared. In addition, the methods used are analysed and possible improvements of the methodology are presented. Finally, this section discusses possible ethical implications from predicting income levels and its possible applications.

5.1 Analysis of results

As can be seen from Table 9, SVM model gives the best performance in all metrics, except for that its runtime is relatively longer. It is worth mentioning that the precision generated by SVM is improved by more than 30% than those of KNN model and Naïve Bayes. This means, when the SVM model predicts *True*, or that the individual has an annual wage above \$50k, it is correct to a greater extent compared to the other models. Similarly, recall from SVM is increased by 46% and 98% compared with those of KNN model and Naïve Bayes, respectively. The differences in terms of accuracy are not as great, however. The KNN model is almost as accurate as the SVM model. As hypothesised in section 1.1, the SVM model had the best performance among the applied methods.

However, the SVM model's main drawback is the long runtime, which is reasonable for such a complicated model. If the KNN and Naïve Bayes models are considered instead, it is evident their runtimes are much shorter but at the cost of less accuracy, precision and recall. Generally, what should be valued the most all depends on the aim and specific situation in which the prediction is done.

5.2 Further developments

For the KNN method, the main disadvantage is that it learns nothing from the training data and simply uses the training data for classification, which makes the model vulnerable to noisy data. Another potential flaw is the way that categorical variables are converted to numerical variables. For example, when categories a, b, c are assigned the value 1, 2, 3, it is assumed that a is closer to b than c, which could be wrong in many cases that have no intrinsic order, such as race and marital-status. One possible solution to this is that these categories can be assigned coordinates (1,0,...,0), (0,1,...,0),..., (0,...,0,1) in N dimensional space, then each pair would have the same distance.

For the SVM methodology possible improvements are foremost in the parameter selection. In this project, the tuning was only done on two different values for each parameter with the radial kernel to save runtime. Preferably, more effort should be put into exploring what parameter values create the best response. This could be done by, for example, fixing the value for one parameter and tuning on the other, noting the error rate and the repeat by

picking another value for the fixed parameter. Important to note here is the risk of overfitting. The goal is not to optimise the parameters and get the overall lowest error rate, but rather to expand the exploration of parameter values, which in this project was rather limited.

For the Naïve Bayes method one potential flaw is the assumption that all attributes are independent. In real life features are not completely independent from each other, so one should take into consideration how accurate the results are for this method. One further possible issue is zero frequency, meaning the test data set may have some feature classification that the training data set does not have. This can be solved easily by resampling the training and test data set.

Finally, another natural way of developing this project is to apply more methods to the dataset. Similarly to Caruana and Niculescu-Mizil 2004, the scope could be expanded to also include methods like neural networks and decision trees.

5.3 Ethical implications

When it comes to ethical implications, this project is rather harmless. This is primarily due to its limited applicability, the prediction is made on only two types of classes which is a bit too rough to give any useful insight in a real-world setting. Nevertheless, issues with the data arises in this project. It is uncertain if the individuals behind the data gave their consent to the data being used for this purpose. Even though no apparent harm is caused by the data usage in this project, it is important to remember the data should be obtained with an informed consent. This cannot be guaranteed in this project.

Another thing worth noting is the harmlessness in this project also stems from the age of the data and the absence of a geographical attributes. If a geographical attribute was available in the dataset, it is reasonable to believe a triangulation could be done to narrow down the identity of the actual individual behind the data. This would in turn be a threat of the anonymity of the individual. What alleviates this threat is the age of the dataset, which is from 1994, possibly making the data invalid for such triangulation.

6 Conclusions

In conclusion, three models have been tested: KNN, SVM and NB. The SVM gave the best performance according to the classification metrics accuracy, precision and recall, as hypothesised in subsection 1.1. However, its runtime was the longest, up to nearly one hour while the other two methods took only a few minutes. If in the particular situation in which the methods are to be used precision and recall are crucial measures, then SVM is the best choice, in spite of the long runtime. However, when it comes to accuracy the difference between the methods are not as great, so KNN and NB can also be considered acceptable choices.

References

Bureau, U.S. Census (n.d.). *Decennial Census of Population and Housing*. URL: https://www.census.gov/programs-surveys/decennial-census.html. (accessed: 10.03.2020).

Caruana, Rich and Alexandru Niculescu-Mizil (2004). "An Empirical Evaluation of Supervised Learning for ROC Area". In: *ROCAI*, p. 1.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. Springer. ISBN: 9781282827264. URL: https://web.stanford.edu/~hastie/Papers/ESLII.pdf.

James, Gareth et al. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer. ISBN: 978-1-4614-7137-0. URL: https://faculty.marshall.usc.edu/gareth-james/ISL/.

Kohavi, Ron (1996). "Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. DOI: http://robotics.stanford.edu/~ronnyk/nbtree.pdf.

Narasimha, Murty M et al. (2015). *Introduction to pattern recognition and machine learning*. Vol. 5. World Scientific.

Statista (n.d.). *Average (median) household income in the United States from 1990 to 2018*. URL: https://www.statista.com/statistics/200838/median-household-income-in-the-united-states/. (accessed: 09.03.2020).

Appendix

Appendix 1 - Preliminary Analysis

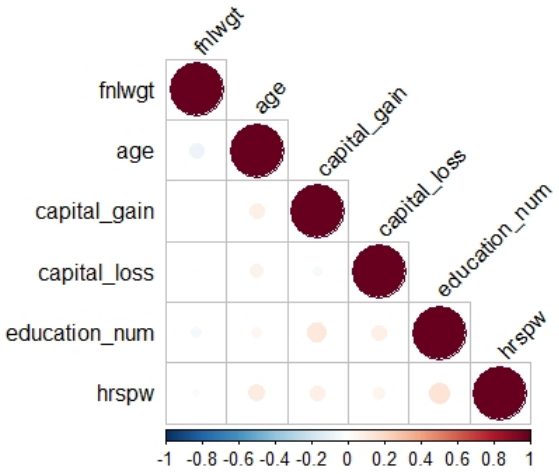

Figure 5: *The figure displays a correlation plot between all numeric variables in the dataset.*

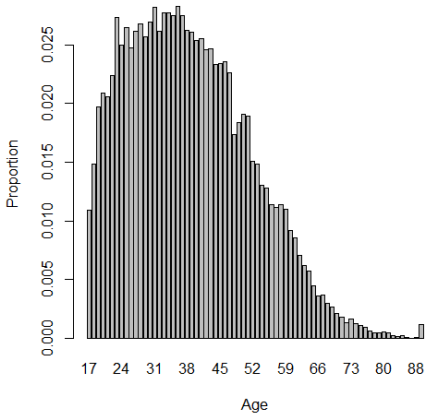

Figure 6: *Histogram over all ages in dataset*

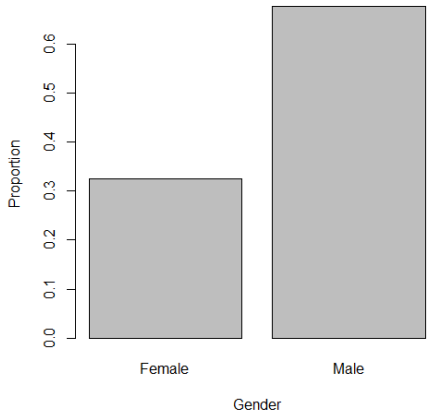

Figure 7: *Histogram over distribution of gender in dataset*

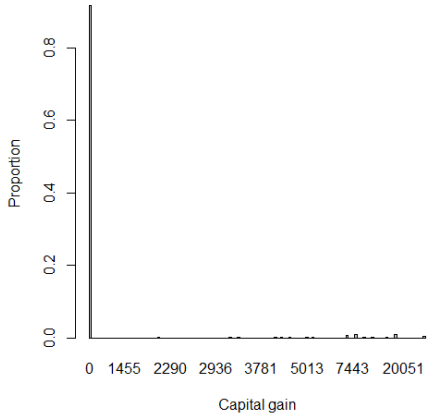

Figure 8: *Histogram over all capital gains in dataset*

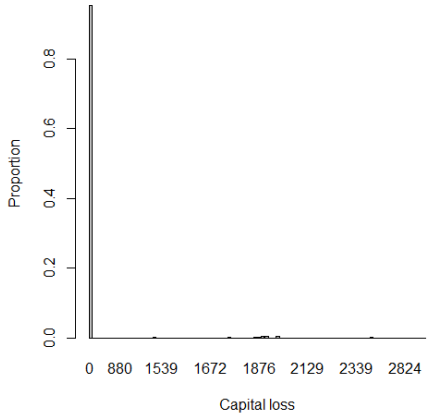

Figure 9: *Histogram over all capital loss in dataset*

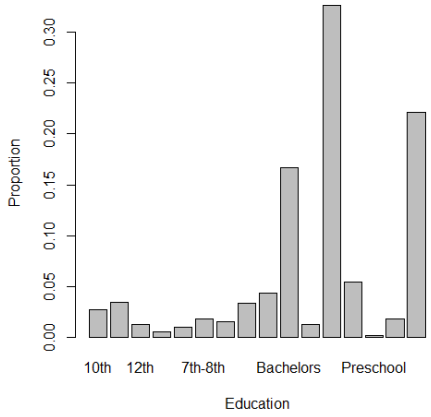

Figure 10: *Histogram over the distribution of educational levels in dataset*

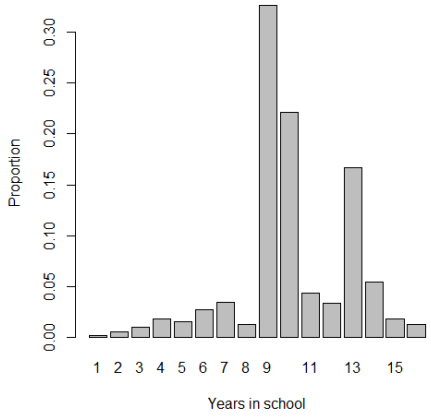

Figure 11: *Histogram over years in school in dataset*

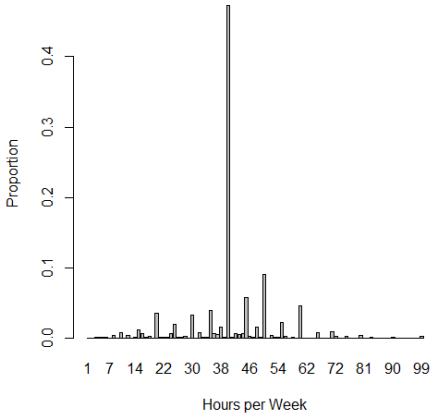

Figure 12: *Histogram over the distribution of hours worked per week in dataset*

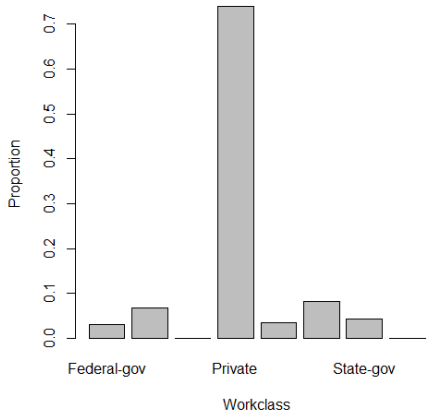

Figure 13: *Histogram over the work classes in dataset*

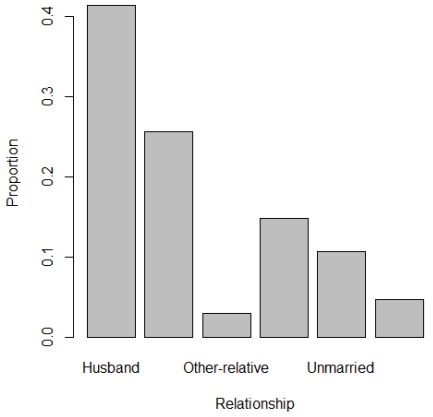

Figure 14: *Histogram over the relationship statuses in dataset*

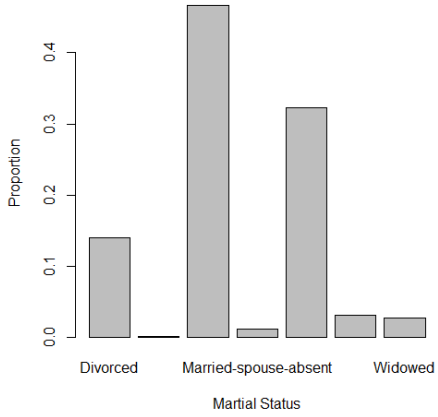

Figure 15: *Histogram over the marital statuses in dataset*

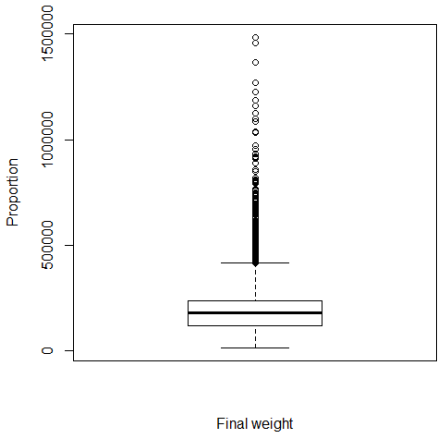

Figure 16: *Boxplot of the final weight attribute in dataset*

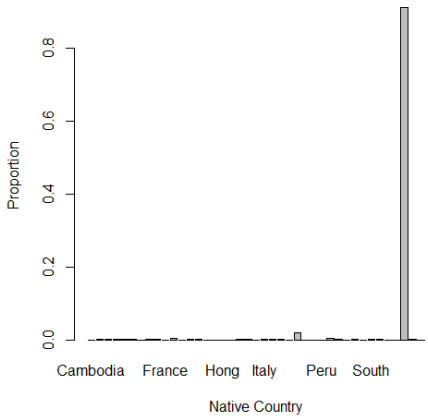

Figure 17: *Histogram over the native origins in dataset*

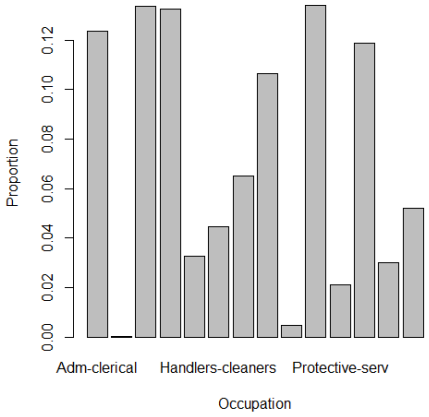

Figure 18: *Histogram of the distribution of occupations in dataset*

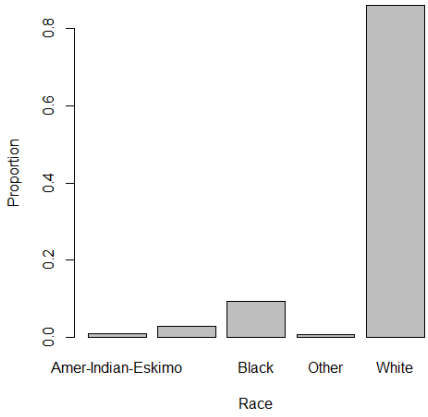

Figure 19: *Histogram over the race attribute in dataset*

Appendix 2 - Missing Relation 1

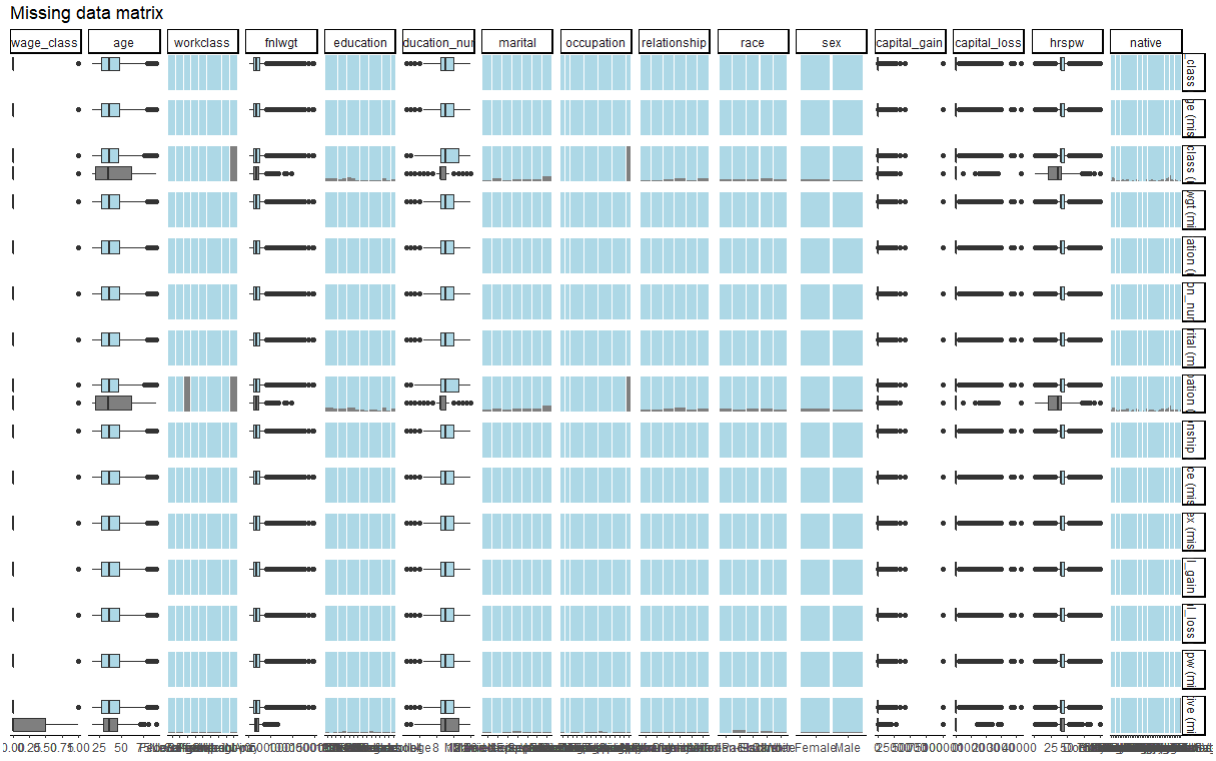

Figure 20: *The figure displays relationships between missing values and observed values in all variables. For each attribute's value the grey bar shows the amount of missing values. In every attribute column we are looking relatively high grey bars compared to others.*

Appendix 2 - Missing Relation 2

Figure 21: *The figure displays relationships between missing values and observed values for certain variables.*

Appendix 3 - KNN

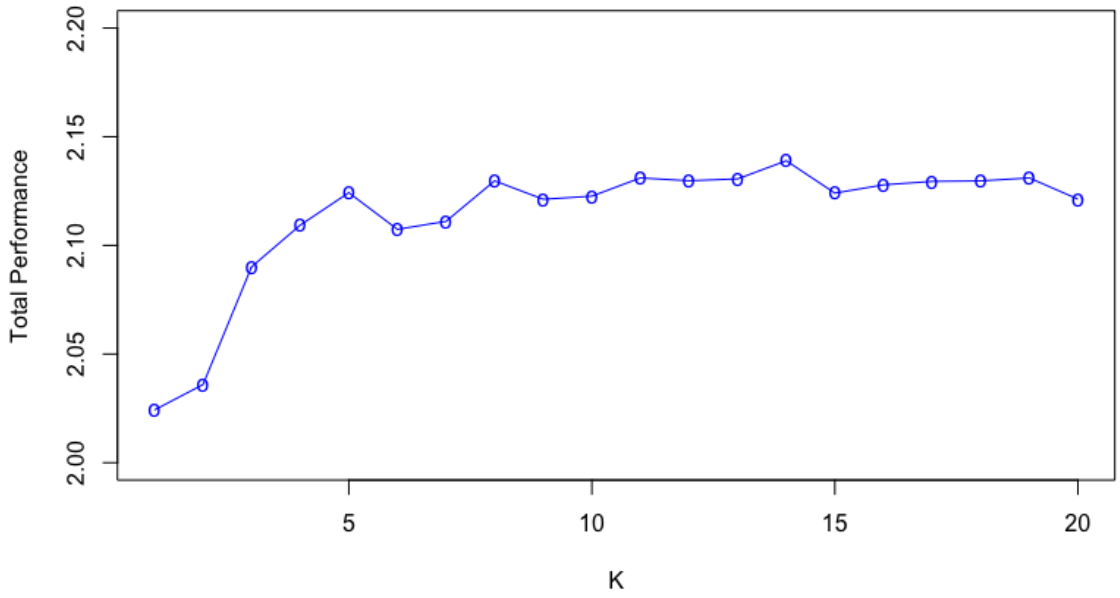

Figure 22: *The figure displays the sum of Accuracy, Precision and Recall of KNN when k = 1,...,20.*