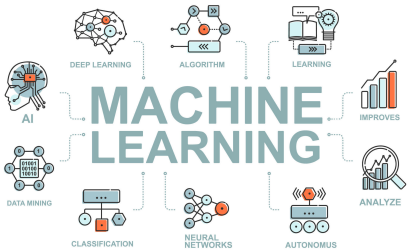
EXPLORING MACHINE LEARNING: PREDICTING INCOME AND RACE

Cassandra Pengelly | 20346212



Data Science 871: Machine Learning Project

Table of Contents

1. Introduction

Labour economists have long been interested in the identifying the factors that influence a person's income.

This paper investigates how well machine learning techniques can predict income and race. The focus is on machine learning and the code, rather than the economic interpretation of the model results.

This paper[1] is structured as follows. First, the data set - NIDS - is discussed in section 2; then the methodology is explained in section 3. Section 4 applies machine learning techniques to the NIDS data set, and comprises two subsections. The first subsection (4.1) compares the effectiveness of linear regression and regularized regression on predicting people's incomes. The second subsection (4.2) evaluated 5 classification algrithms - Linear Discriminant Analysis, Classification and Regression Trees, k-Nearest Neighbors, Support Vector Machines with a linear kernel and Random Forest - on their accuracy in predicting a person's race.

2. Data

The data used for this assignment was sourced from Wave 5 of the National Income Dynamics Survey (NIDS) (*National income dynamics study 2017, wave 5 dataset* (2018)). The survey is a nationally representative household panel study, which started in 2008 with a group of over 28,000 individuals from 7,300 households. The same households are surveyed every 2 years for NIDS. The latest survey - wave 5 - was conducted in 2017. For wave 5, a total of 39,434 individuals were interviewed; 20,113 of which were part of the original study - wave 1 - and 2,016 were from a top-up sample. NIDS is funded by the Department of Planning, Monitoring and Evaluation and the survey is implemented by the Southern Africa Labour and Development Research Unit (SALDRU) at the University of Cape Town. The data set is comprehensive and covers topics relating to poverty, health, household composition, mortality, expenditure, income and employment. The NIDS data set is partitioned into different units of observations (e.g. adults, children, household etc.); for this assignment, data on adults was used.

One weakness of the NIDS data set is that it suffers from the common problem that households at the higher end of the income distribution tend to be underrepresented. This could be explained by the fact that the rich refuse to fill out forms or they underreport their incomes. Because race and income are highly correlated in South Africa, this could also imply that white people are undersampled. To correct for the sample data not representing the population income/race distribution, some balancing techniques are applied in section 4.2.

[1] This assignment was written using the package by Katzke (2017)

3. Methodology

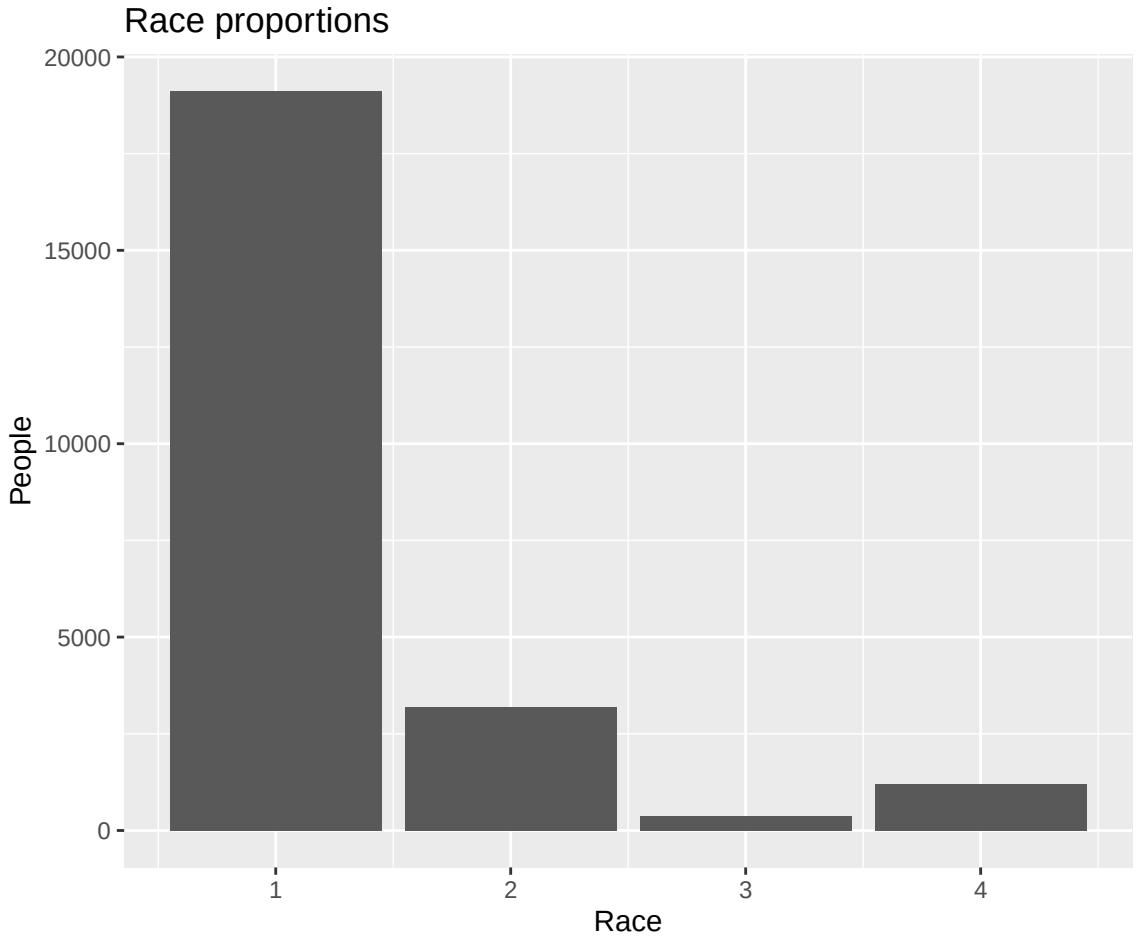
I first made use of SQLite[2] to investigate and visualise the data. I first ran the code in the R script called SQL. I opened a new connection and called it nids, and wrote the NIDS data, which included waves 2-5, into tables in the NIDS database. I used a few lines of code to check what tables were in nids and what their source was. Then I started to explore the NIDS wave 5 data, for example looking at the column names. I queried the data and selected 7 variables for analysis: date of birth, income, gender, marital status, race, years of schooling and tertiary qualification. I used date of birth to calculate the variable 'age'.

I cleaned the data by renaming the variables to be reader-friendly and removed values that were nonsensical (e.g. negative years of schooling) by applying filters to the data. I wanted to see the proportion of the races so I first manipulated the data in the SQL script and then used the function 'show_query' to get the code for SQL. I copied this into the r chunk in the markdown file and then graphed the results using ggplot2. The code below shows some of this process and output:

```
library(DBI)
nids <- DBI::dbConnect(RSQLite::SQLite(), "data/nids-db~output.sqlite")
```

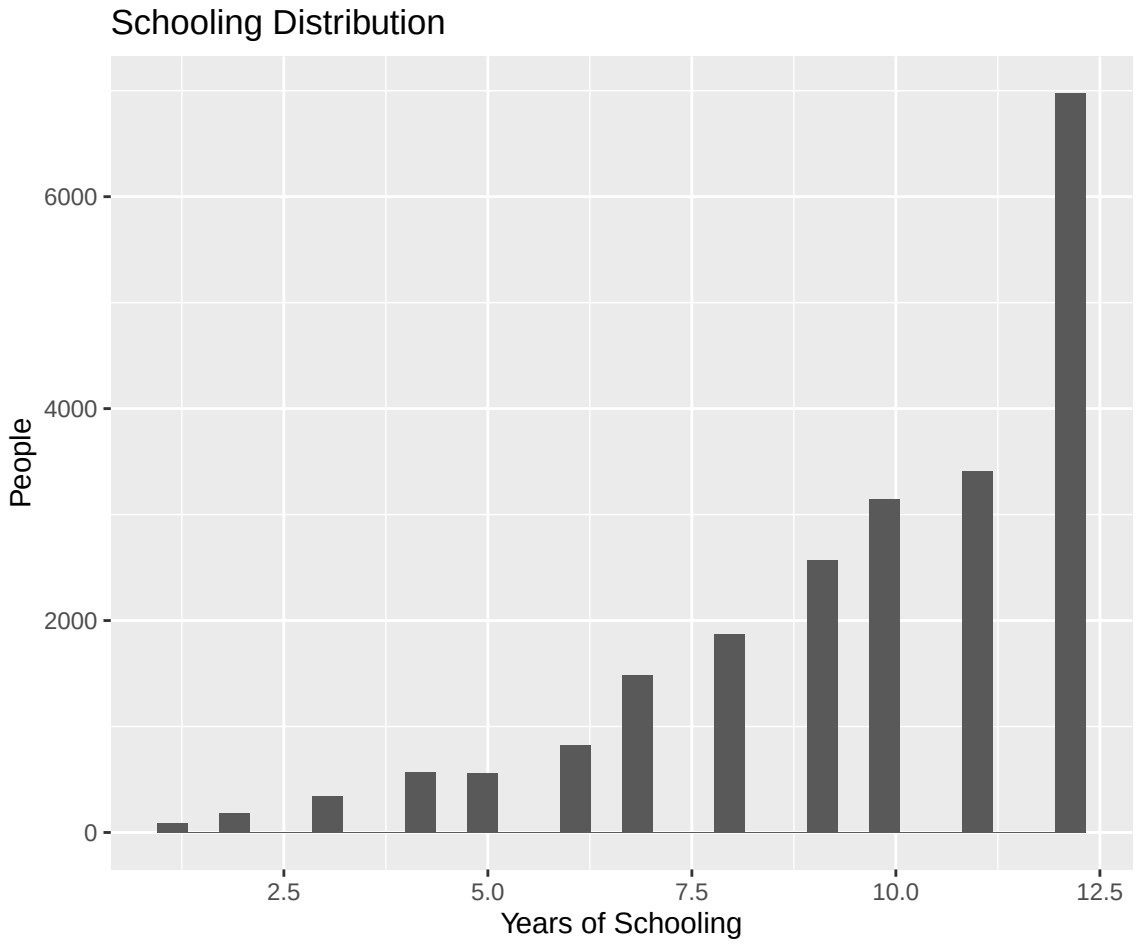
```
SELECT `race`, COUNT(*) AS `n`
FROM (SELECT `w5_a_dob_y`, `w5_a_gen` AS `gender`, `w5_a_popgrp` AS `race`,
`w5_a_em1pay` AS `income`, `w5_a_mar` AS `married`, `w5_a_edschgrd` AS `school`,
`w5_a_edter` AS `tertiary` FROM `wave5`) WHERE (`race` > 0.0 AND `race` < 5.0)
GROUP BY `race`
```

[2] I found I struggle quite a bit using SQL but it has got easier with practice. I'm still not 100% comfortable with it so I used dplyr to manipulate the data for the machine learning section.

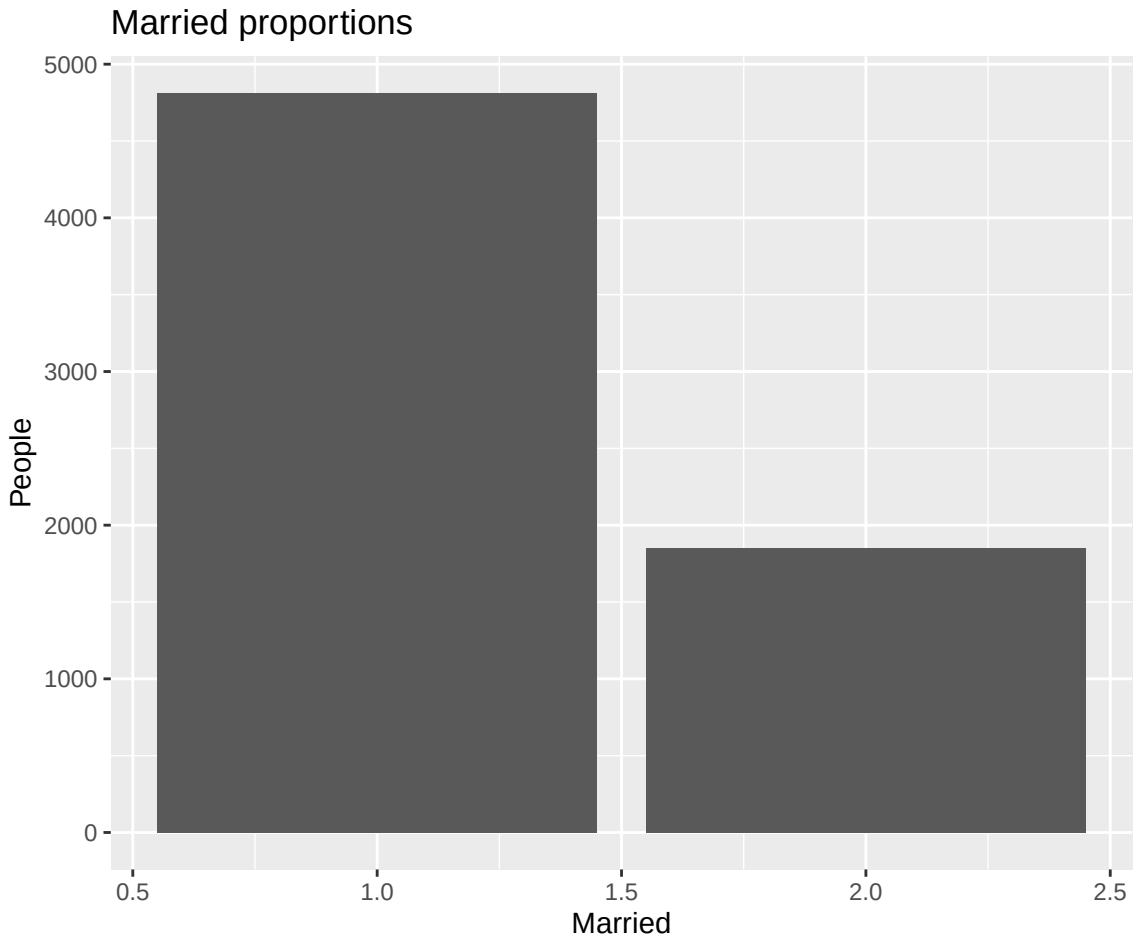


Race
1 = African
2 = Coloured
3 = Asian/Indian
4 = White

The bar graph above shows that the majority of people sampled are African and the smallest proportion are Asian/Indian. In general the proportions appear to match the race distribution of South Africans. The histogram below reports the number of years of schooling. We can see that the majority of the sample has had 12 years of schooling. The data for this graph was also manipulated using the SQL script and then the SQL code was copied into the r chunk.



Looking at the proportion of people who are married versus those that are not, we can clearly see that there are more people married than unmarried in this sample, as the bar graph below demonstrates.



Married
1 = Married
2 = Not married

If I had wanted to use more than one of the NIDS data sets, say wave 4 and 5, I could merge the two data sets using the SQL code below. I lightly cleaned both data sets beforehand and then used the union_all function to merge them.

```
SELECT 2021.0 - `w5_a_dob_y` AS `age`, `w5_a_gen` AS `gender`,
`w5_a_popgrp` AS `race`, `w5_a_em1pay` AS `income`,
`w5_a_mar` AS `married`, `w5_a_edschgrd` AS `school`,
`w5_a_edter` AS `tertiary` FROM `wave5`
UNION ALL
SELECT 2021.0 - `w4_a_dob_y` AS `age`, `w4_a_gen` AS `gender`,
`w4_a_popgrp` AS `race`, `w4_a_em1pay` AS `income`,
`w4_a_mar` AS `married`, `w4_a_edschgrd` AS `school`,
```

```
`w4_a_edter` AS `tertiary` FROM `wave4`
```

The table below provides a glimpse into the first 5 rows of the newly joined data set. I output a variable in the previous chunk as a dataframe and then displayed the dataframe in a table in the next r chunk. However, I could have used the function 'head' in the SQL script on the joined dataframe and the used the 'collect()' function to save the output to a global object and then displayed that object as a dataframe.

age	gender	race	income	married	school	tertiary
41	1	2	23752	1	12	1
28	1	1	NA	NA	10	2
42	2	1	NA	1	11	1
49	1	1	NA	1	11	1
54	1	1	NA	NA	25	NA

After the initial data exploration I decided to focus on predicting income and race using machine learning techniques, which are presented in the following section.

4. Machine Learning

4.1. Predicting Income

Econometrics often makes use of regression analysis to model economic phenomena, test economic hypotheses and to forecast economic activity Studenmund (2014: 2). A popular method in econometric regression modeling is that of Ordinary Least Squares; however, advances in machine learning have presented alternative/augmenting methods that may be (more) useful. One such augmenting method is K-fold cross validation, which evaluates the skill of machine learning models. As Rodriguez, Perez & Lozano (2009: 569) explain, in K-fold cross validation, a data set is randomly split into k number of groups, of similar sizes. The first group is considered a validation set and the method is fitted to the other $k - 1$ groups.

Below, 4.1 displays 3 different linear regressions of log of income using K-fold cross validation. For these regressions I built a function "linreg", which takes in a data frame, cleans and splits the data into a training (70% of the full data set) and a test set (30% of the full data set) and runs 3 different linear regressions, applying K-fold cross validation. The sample size for regressions amounts to 4258 observations, with the training and testing sets amounting to 2982 and 1276 respectively. The results of the regressions are then collected and stored in a list, which is returned by the function. I use k =

10, because empirically k=10 has been shown have test error rate estimates that have relatively low bias and variance (Kassambara (2018a)). I have also set seed in the function for reproducibility.

Based on the Mincerian wage equation, Regression 1 (see 4.1) regresses log of income on age, years of schooling and a dummy variable for if a person has a tertiary qualification or not. Regression 2 includes a variable for age-squared and the categorical variable race. Regression 3 includes a variable each for gender and marriage. The signs of the coefficients of the three regressions look fairly standard[3] and most of the coefficients are statistically significant at 1% and lower. For the variables whose coefficients are not statistically significant, labour market literature and economic theory suggest that they are important controls and should be included, which justifies their presence.

[3]This at least is a good indication that the data is fairly well cleaned and is usable for testing the machine learning techniques

Table 4.1: Log-Income Regression Output

	Reg 1	Reg 2	Reg 3
(Intercept)	6.097 ***	5.791 ***	5.526 ***
	(0.145)	(0.367)	(0.354)
age	0.019 ***	0.041 **	0.040 **
	(0.002)	(0.015)	(0.015)
school	0.127 ***	0.109 ***	0.104 ***
	(0.009)	(0.009)	(0.008)
tertiary	0.691 ***	0.624 ***	0.594 ***
	(0.048)	(0.046)	(0.044)
age2		-0.000	-0.000 *
		(0.000)	(0.000)
‘raceAsian/Indian‘		0.575 ***	0.554 ***
		(0.123)	(0.117)
raceColoured		0.023	0.008
		(0.048)	(0.046)
raceWhite		0.767 ***	0.777 ***
		(0.070)	(0.066)
married			0.317 ***
			(0.045)
male			0.477 ***
			(0.039)

*** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$.

Typically, economists are interested in evaluating the performance of a model, which can be done by assessing how well the model predicts the outcome variable. A useful statistical metric for measuring the performance of a regression model is the Root Mean Squared Error (RMSE) (Kassambara (2018b)).The RMSE measures the average error performed by the model in predicting the outcome

for an observation. The mathematical formula for the RMSE is given by

$$RMSE = mean(\sqrt{(observeds - predicteds)^2})$$

This implies lower the RMSE, the better the model performs.

Table 4.2 reports the RMSES for both the training data and the test data. We can see that the RMSEs decreased from regression 1 to regression 3 for both the training and test data. This indicates that regression 3 is a better model than both regressions 1 and 2 (and that regression 2 has better predictive power than regression 1). The RMSEs for the regression based on the training data are lower than for the test data. However, the RMSEs are close enough between the two data sets for all 3 models that the out-of-sample performance is fair; it does not seem that any of the models have been overfitted to the training data.

Table 4.2: Regression RMSEs and Observations

Regression	RMSE Train	RMSE Test
Reg 1	0.86	0.91
Reg 2	0.83	0.88
Reg 3	0.79	0.82

The table above shows that as more explanatory variables are added to the regression, it appears that the model improves in predictive power. However, it could actually be the case that the model is over fitting the data. One method of addressing this issue is to use regularized regression, which constrains the estimated coefficients. It does this by introducing a penalty parameter in the objective function such that the sum of the sum of squared errors and the penalty parameter is minimised. Two common penalty parameters include the ridge and lasso methods.

To apply the ridge and lasso methods I created two functions: 'plot_ridge' and 'plot_lasso'. These functions use the 'glmnet' package to run the regressions and return a plot of the results, which are reported below. The tuning parameter here is given by λ. Initially, as λ increases there is a decrease in the mean-squared error (MSE) for the lasso method, where the first dotted line indicates the lowest MSE. For the smaller values of lambda, this means that the lasso models has improved upon the OLS model - it is providing a better fit. In the ridge model, as λ increases, the coefficient sizes are being reduced but the number of coefficients remains the same. In the lasso model, after the log of λ increases to more than -3, the number of parameters starts to decrease.

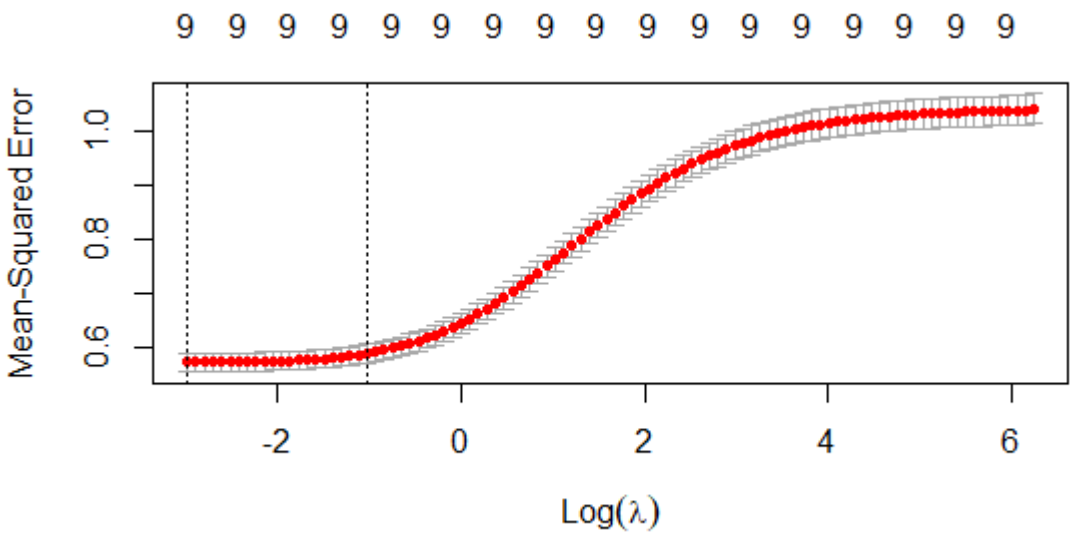


Figure 4.1: Ridge

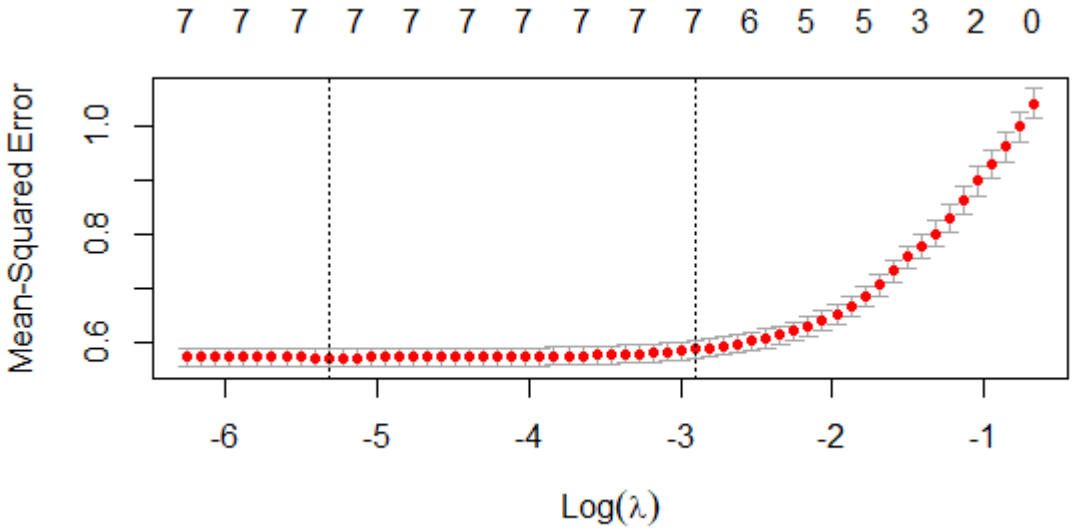


Figure 4.2: Lasso

I then created a function 'comp_RMSE', which finds the λ that minimises the RMSEs for the ridge and lasso models and records the minimised RMSEs. These models are then used to predict the log of income for the test data. The bar chart below displays the RMSEs for the ridge and lasso models and compares them to a base linear regression and the three regressions from 4.1. Here, I adjusted the code from Hartmann & Waske (2018) tutorial.

The graphs again shows that the RMSEs are lower for the training data than for the test data for all the models.

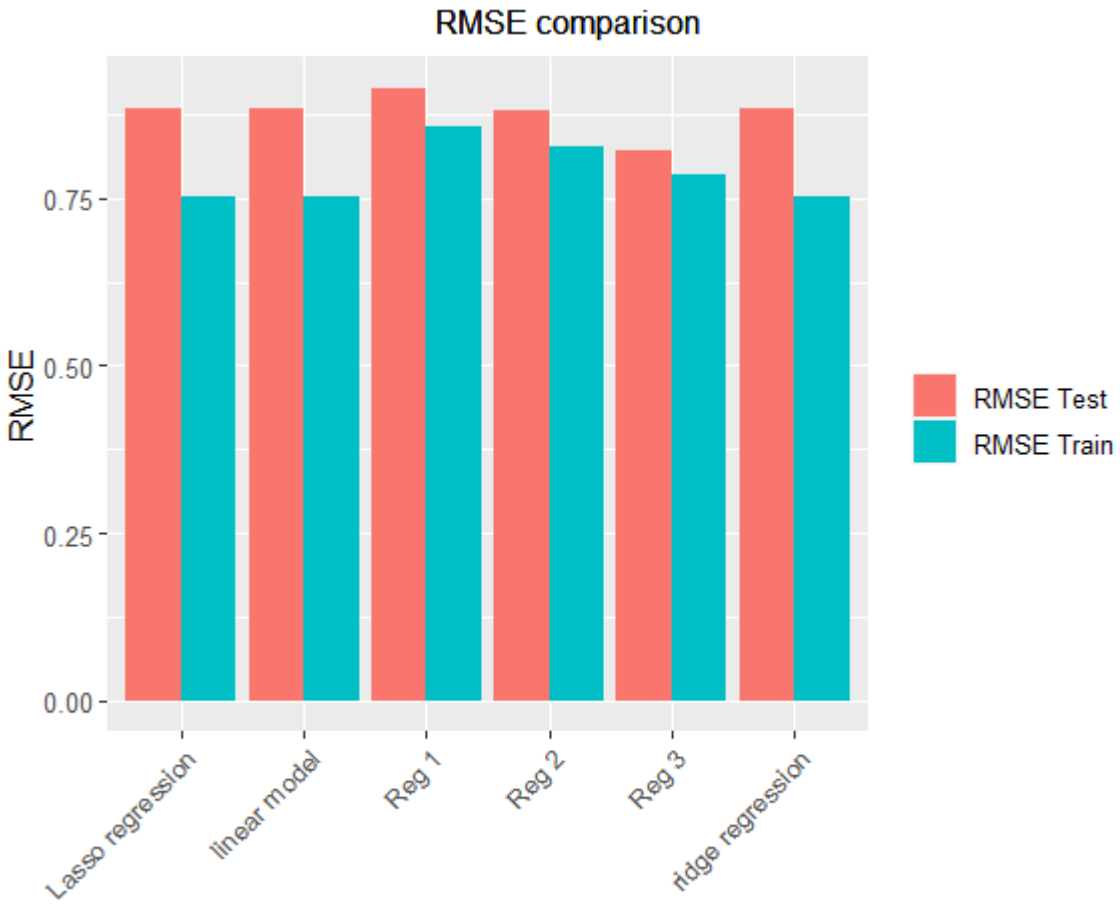


Figure 4.3: RMSE comparison

While economists are interested in predicting wage based on variables such as race and education, it might be interesting to see if race can be predicted using income and other variables. The next section explores this idea by applying classification algorithms on the NIDS data set.

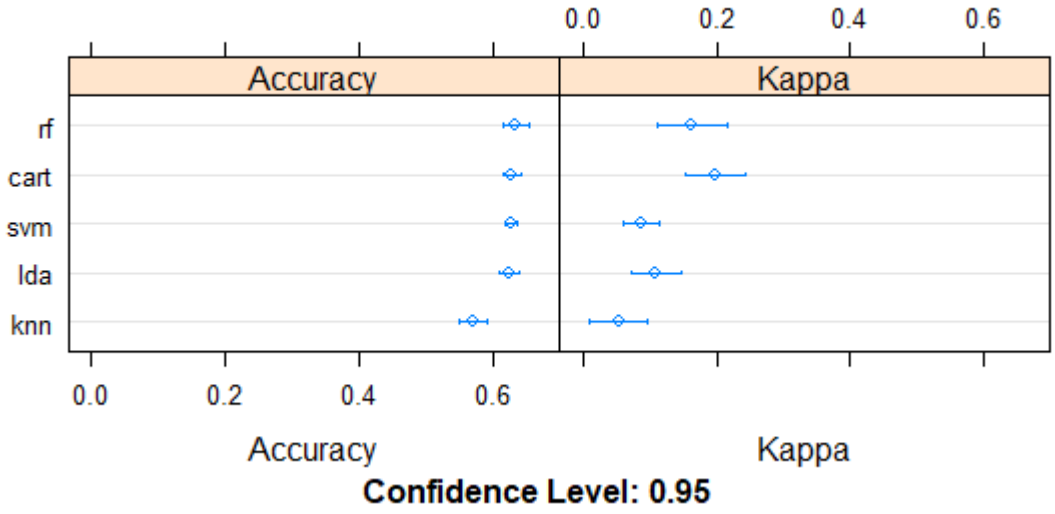


Figure 4.4: Machine Learning applied to unbalanced data

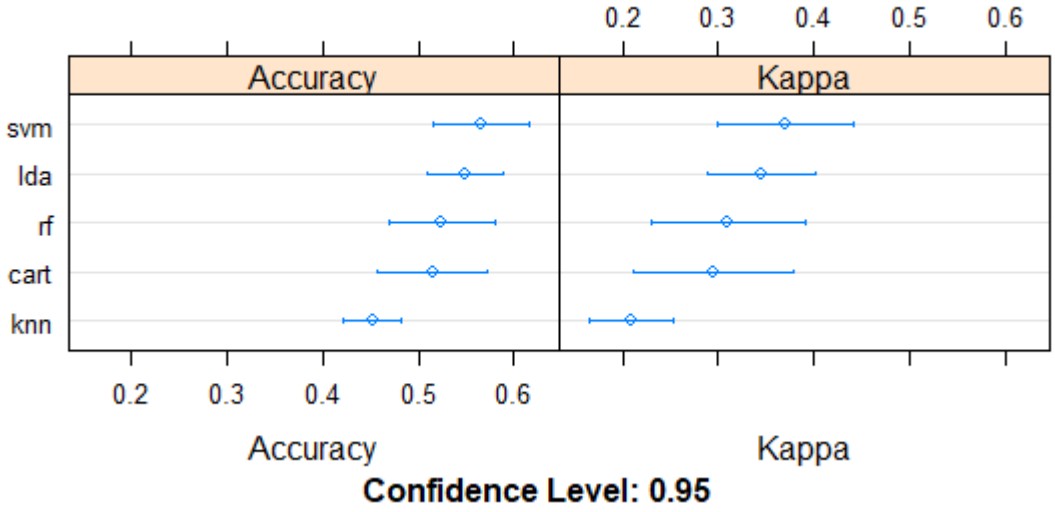


Figure 4.5: Machine Learning applied to balanced (undersampled) data

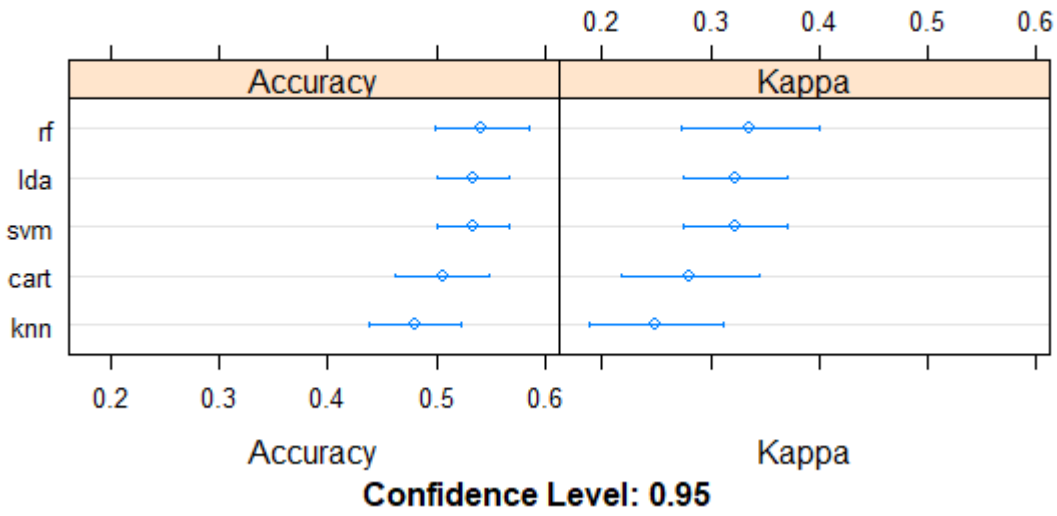
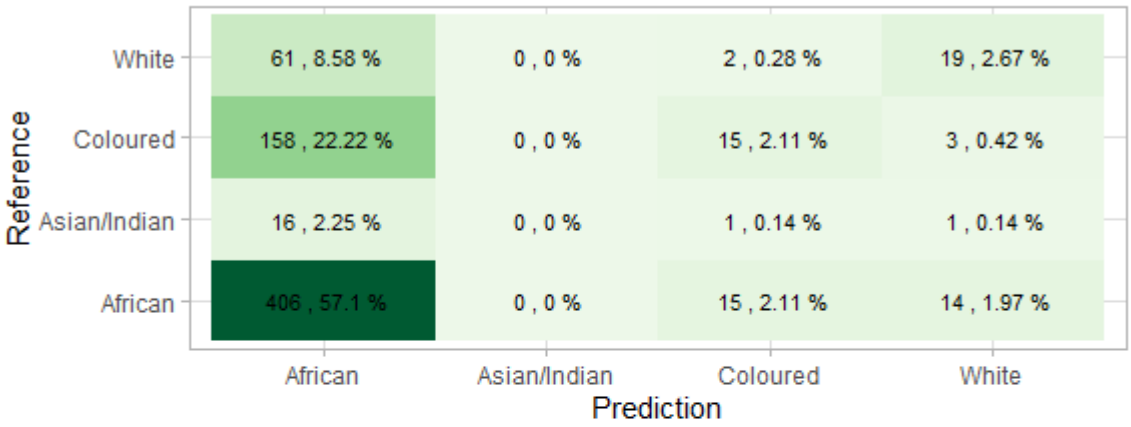


Figure 4.6: Machine Learning applied to balanced data

The graph below displays the confusion matrix and the tables report the relevant statistics.



	Statistics
Accuracy	0.62
Kappa	0.12

Figure 4.7: Random Forest Confusion Matrix

	Sen	Spec	Pos	Neg	Prec
Class: African	0.93	0.15	0.63	0.59	0.63
Class: Asian/Indian	0	1	NaN	0.97	NA
Class: Coloured	0.09	0.97	0.45	0.76	0.45
Class: White	0.23	0.97	0.51	0.91	0.51

	Rec	F1	Prev	DetRat	DetPrev	BalAcc
Class: African	0.93	0.75	0.61	0.57	0.9	0.54
Class: Asian/Indian	0	NA	0.03	0	0	0.5
Class: Coloured	0.09	0.14	0.25	0.02	0.05	0.53
Class: White	0.23	0.32	0.12	0.03	0.05	0.6

Figure 4.8: Random Forest Statistics

5. Conclusion

References

10 Hartmann, K., K & Waske, B. 2018. E-learning project SOGA: Statistics and geospatial data analysis.

Kassambara, A. 2018a. Cross-validation essentials in r.

Kassambara, A. 2018b. Linear regression essentials in r. [Online], Available: http://www.sthda.com/english/articles/40-regression-analysis/165-linear-regression-essentials-in-r/.

Katzke, N.F. 2017. *Texevier: Package to create elsevier templates for rmarkdown*. Stellenbosch, South Africa: Bureau for Economic Research.

National income dynamics study 2017, wave 5 dataset. 2018. Cape Town, South Africa: Department of Planning, Monitoring,; Evaluation [funding agency] & DataFirst [distributor]. [Online], Available: https://doi.org/10.25828/fw3h-v708.

Rodriguez, J.D., Perez, A. & Lozano, J.A. 2009. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*. 32(3):569–575.

Studenmund, A.H. 2014. *Using econometrics a practical guide*. Pearson.