MECON6102 – PSet 2

1. Simple Logistic Model

The results from the baseline logit model as specified in section 2.1 are displayed below:

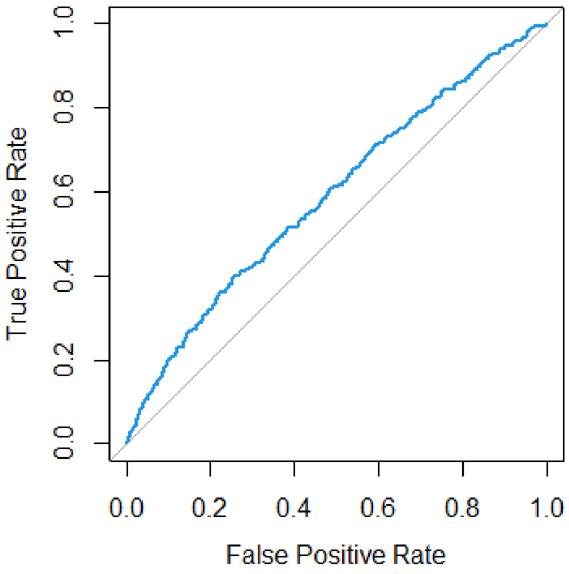
Outcome Variable	default_label
Regressors	(1) income (2) past_bad_credit

Logit Output

	Estimate	***Std. Error***	***z value***	***Pr(>|z|)***
(Intercept)	-5.386e+00	5.820e-01	-9.255	< 2e-16
income	4.366e-05	7.936e-06	5.501	3.78e-08
past_bad_credit	1.254e+00	5.822e-01	2.154	0.0313

Area Under Curve	**0.5889**

Receiver Operating Characteristic Curve



2. Extended Multivariate Logistic Models

From the baseline specification in section 2.1, the remaining variables in the data set were systematically added to the regression one by one, with the incremental effect on AUC observed and recorded at each stage. Various transformations were also performed in an attempt to better model the given data. For example, the income data was *logged*, in keeping with most economic methods which prefer to work with income data in the form ln(x); *squared*, to capture potential nonlinearities in the relationship with default risk, and *standardised*, so as to scale the data into a more manageable range reflecting the binary nature of most of the other variables in the data set. Numerous permutations of the variables and their transformed states were tested, with the AUC gradually increasing. Interestingly, outcomes using ln(income) performed worse than those with the raw data. Quadratic terms were also seen to have only marginal impact. It is not useful to show the results of all regression specifications tested, so the table below shows the outcome of the regression with all of the variables included, with income and age standardised, and income raised to the second power.

Outcome Variable	default_label
Regressors	(1) standardise.income (2) stand.inc.quad (3) past_bad_credit (4) standardise.Age (5) gender (6) edu (7) housing (8) job_occupation (9) married

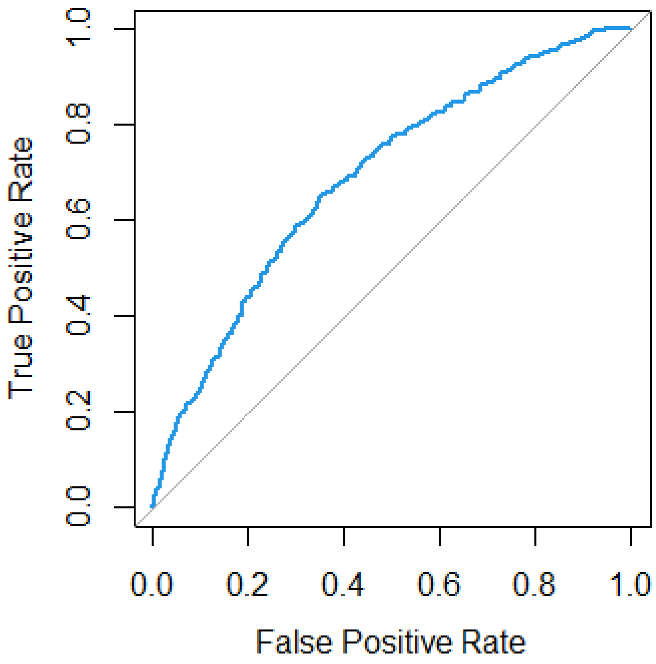
Logit Output

	Estimate	*Std. Error*	*z value*	*Pr(>|z|)*
(Intercept)	-4.005171	0.610187	-6.564	5.24e-11
standardise.income	0.473502	0.094580	5.006	5.55e-07
stand.inc.quad	-0.007677	0.040716	-0.189	0.850447
past_bad_credit	1.261049	0.583178	2.162	0.030590
standardise.Age	0.075034	0.069019	1.087	0.276973
gender	0.104984	0.118038	0.889	0.373780
edu1	-0.240701	0.191105	-1.260	0.207843
edu2	-0.481819	0.200216	-2.406	0.016107

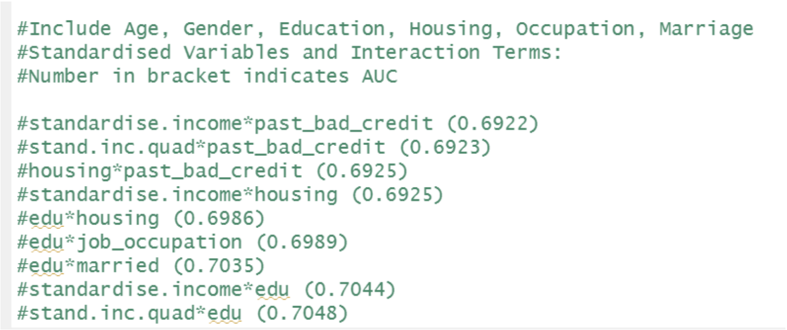
edu3	-0.915440	0.235032	-3.895	9.82e-05
edu4	-1.224578	0.378736	-3.233	0.001224
housing	-1.085243	0.140927	-7.701	1.35e-14
job_occupation	-0.452784	0.128244	-3.531	0.000415
married	0.004453	0.119683	0.037	0.970319

Area Under Curve	**0.6907**

Receiver Operating Characteristic Curve



Having exhausted the variables and their transformations, the AUC had only reached 0.6907. The next logical improvement required interacting the variables so that their combined effect could add explanatory power to the model. The p-values in the previous logit summary indicate which variables are more significant in their own right – standardised income and housing, for example. We can also use economic intuition to guide our testing of interaction terms – for example we might assume that the effect of income on default risk varies depending on age – younger individuals with high income may exhibit lower risk aversion than older individuals with the same level of income. However, whilst the p-values from the uninteracted variables, combined with economic intuition can provide a starting point for the addition of the interaction terms, the next stage was characterised predominantly by trial and error. After each interaction term was added, the incremental change to AUC was recorded. Terms that caused a significant increase were kept, whilst those that made little or no difference to the AUC were discarded. The image below shows how this process evolved:



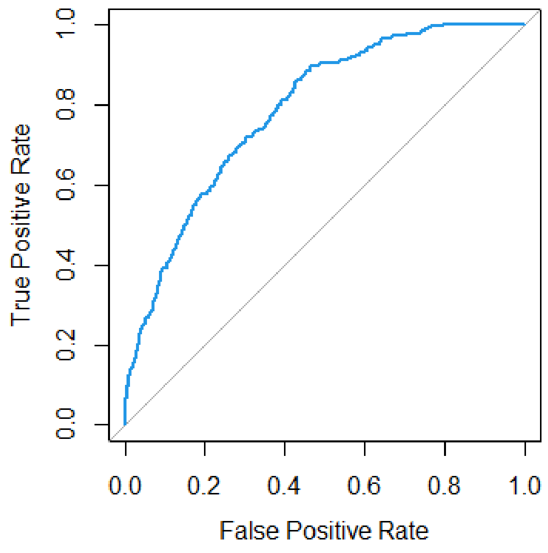
```
#Include Age, Gender, Education, Housing, Occupation, Marriage
#Standardised Variables and Interaction Terms:
#Number in bracket indicates AUC

#standardise.income*past_bad_credit (0.6922)
#stand.inc.quad*past_bad_credit (0.6923)
#housing*past_bad_credit (0.6925)
#standardise.income*housing (0.6925)
#edu*housing (0.6986)
#edu*job_occupation (0.6989)
#edu*married (0.7035)
#standardise.income*edu (0.7044)
#stand.inc.quad*edu (0.7048)
```

Eventually, having tested numerous terms, the desired AUC threshold of 0.78 was exceeded. The details of the most parsimonious model to exceed this threshold are presented below. The logistic summary is not included as the complex nature of the model and the interaction terms with up to 7 components mean that coefficient interpretation becomes an incredibly long and complex process:

Outcome Variable	default_label
Regressors	(1) edu*past_bad_credit (2) gender*past_bad_credit (3) standardise.Age*married*gender*edu (4) standardise.Age*married*gender*stand.inc.quad (5) (4) + standardise.income (6) (5) + job_occupation (7) (6) + housing (8) housing*married*edu*standardise.income (9) housing*married*edu*standardise.Age (10)housing*married*edu*gender (11)housing*married*edu*job_occupation (12)housing*married*edu*stand.inc.quad
Area Under Curve	**0.7869**

Receiver Operating Characteristic Curve



3. SVM/Random Forest

Next, we progress to using different approaches. Firstly, the ROSE package was used to address the class imbalance problem: initially the default_label was distributed as follows:

0	1
13674	308

After using the package's oversampling command *ovun.sample(method = "over")* the distribution was balanced, by duplicating observations where default_label evaluated to 1:

0	1
13674	13674

Some of the models were computationally intensive and so a smaller balanced data set using the undersampling command *ovun.sample(method = "under")* was also constructed.

0	1
298	308

We subsequently scaled the income and age variables in the new balanced data set, as before.

Support Vector Machine

The results of the baseline SVM using past_bad_credit and income are presented below. Both linear and radial kernels were used, and the linear kernel offered better performance – a higher AUC and fewer support vectors, so future specifications typically used a linear kernel too.

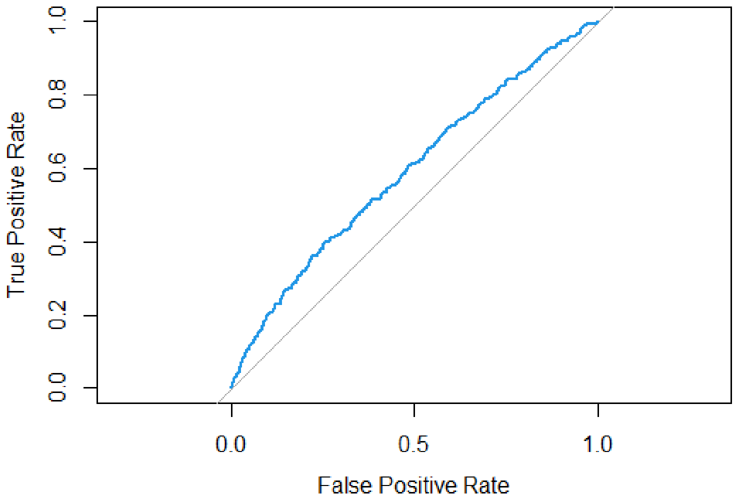
Outcome Variable	default_label
Regressors	(1) income (2) past_bad_credit

SVM Summary Output

Number of Support Vectors	**619**
Number of Classes	**2**

Area Under Curve	**0.5889**

Receiver Operating Characteristic Curve



To improve model performance with the SVM approach, it became imperative that a balanced data set were used. Due to computation time, use of the overbalanced data set became undesirable and so instead the underbalanced data set was used. Using the same set of interaction terms as the final logit model with this data set resulted in optimal SVM performance, as displayed below.

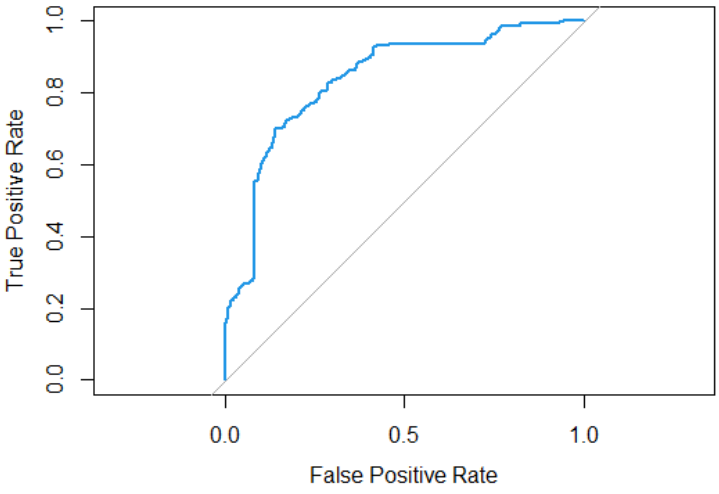
Outcome Variable	default_label
Regressors	(1) edu*past_bad_credit (2) gender*past_bad_credit (3) standardise.Age*married*gender*edu (4) standardise.Age*married*gender*stand.inc.quad (5) (4) + standardise.income (6) (5) + job_occupation (7) (6) + housing (8) housing*married*edu*standardise.income (9) housing*married*edu*standardise.Age (10)housing*married*edu*gender (11)housing*married*edu*job_occupation (12)housing*married*edu*stand.inc.quad

SVM Summary Output

Number of Support Vectors	**436**
Number of Classes	**2**

Area Under Curve	**0.8363**

Receiver Operating Characteristic Curve



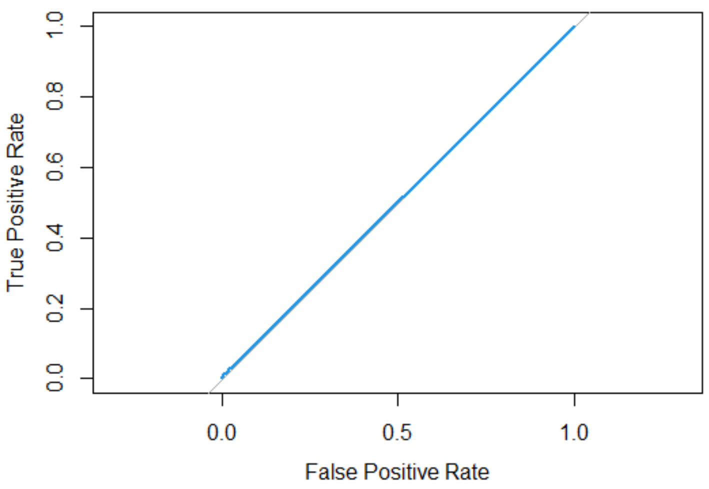
Random Forest

Next, we used the random forest approach to see how performance compared when using the same data and variables as the SVM models. Overall it was found that the SVM models attained superior accuracy as measured by AUC, so we will only briefly present some random forest output below.

The baseline specification using income and past_bad_credit is virtually useless, with the AUC only marginally outperforming random classification. The results are shown below.

Outcome Variable	default_label
Regressors	(1) income
	(2) past_bad_credit
Area Under Curve	**0.5014**

Receiver Operating Characteristic Curve

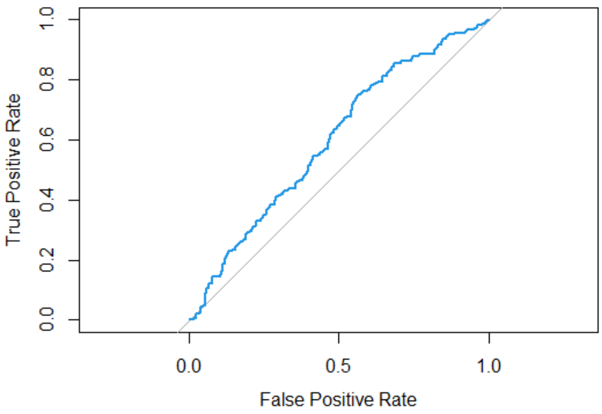


Replicating the same data inputs and variables as the successful SVM with the AUC of 0.8363 also results in undesirable performance for the random forest approach, with the AUC only rising to 0.5966 with the underbalanced data set and the same set of interaction terms.

Outcome Variable	default_label
Regressors	(1) edu*past_bad_credit
	(2) gender*past_bad_credit
	(3) standardise.Age*married*gender*edu
	(4) standardise.Age*married*gender*stand.inc.quad

	(5) (4) + standardise.income (6) (5) + job_occupation (7) (6) + housing (8) housing*married*edu*standardise.income (9) housing*married*edu*standardise.Age (10)housing*married*edu*gender (11)housing*married*edu*job_occupation (12)housing*married*edu*stand.inc.quad
Area Under Curve	**0.5966**

Receiver Operating Characteristic Curve



As a result, the favoured approach in section 2.3 is a support vector machine, using an underbalanced data set and the 12 interaction terms from the initial logit model. For such a model, the AUC is 0.8363.

4. Out-of-Sample Testing

The final task involves splitting the data into a training and a testing set so as to assess how the logit model performs out of sample.

We use the logit model from 2.2 but modify it slightly so as not to use standardised terms which did not work with the testing set. The results of the training model are displayed below:

Outcome Variable	default_label
Regressors	(1) edu*past_bad_credit (2) gender*past_bad_credit (3) Age*married*gender*edu

(4) Age*married*gender*income

(5) (4) + income

(6) (5) + job_occupation

(7) (6) + housing

(8) housing*married*edu*income

(9) housing*married*edu*Age

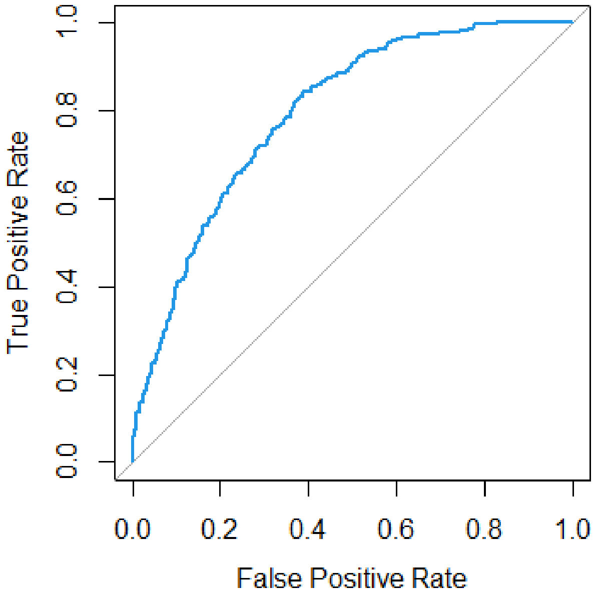
(10)housing*married*edu*gender

(11)housing*married*edu*job_occupation

TRAINING DATA

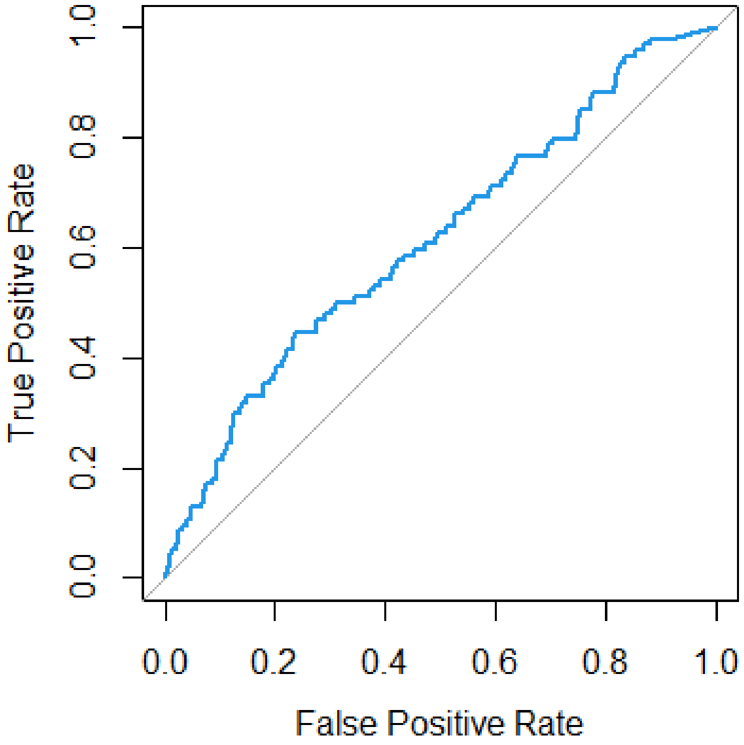
Area Under Curve	**0.7933**

Receiver Operating Characteristic Curve



TESTING DATA

Area Under Curve	**0.614**



Unfortunately, the AUC achieved by the model when used out of sample is only 0.614, indicating that the logit specification employed in the training stage is overfitting the data, leading to an unrealistic level of accuracy in-sample, and poor performance out of sample.

This overfitting is the cause of the discrepancy between the results in section 2.2 and 2.4. In section 2.2, all available data is used in the model. As more interaction terms are added, the model better fits the given data and higher accuracy in describing the data is achieved. However, in section 2.4, we constrain the data available for the initial modelling. The model that we construct is then tested on completely new data in the testing set so that its performance can be tested outside of the sample from which it was derived. However, because we have trained the model too well on the given training data, and fitted it to the training set's unique idiosyncrasies, we find that it performs poorly when tasked with classifying new observations. This is why the AUC in section 2.2 is much higher than that in section 2.4.