MSDS 6372 Project 2

3/16/2020

Ellen Lull
Fred Poon
Sanjay Pillay
Jordan Salsman

Project Professor: Dr Jacob Turner

Citations

Document all citations here

INTRODUCTION

Problem Statement: Direct marketing campaigns often produce mixed results. In this project, we compare several models to analyze and predict the outcome of a customer subscription for a term deposit based on previous recorded data. The models used are Logistic regression, LDA and a non parametric Random Forest. The main goal is to have a higher sensitivity that is identifying a potential customer who would subscribe for a term deposit.

DATA DESCRIPTION

We are using the Bank Marketing dataset. The data contains the phone call based marketing campaigns of a Portuguese banking institution. More than one phone call may have been made to a client.

Variable	Summary	Description
y	no : 36548 yes: 4640	yes=subscribed to deposit no=did not subscribed
Age	**Group.1 “yes” “no”** x.Min. "17.00000" "17.00000" x.1st Qu "31.00000" "32.00000" x.Median "37.00000" "38.00000" x.Mean "40.91315" "39.91119” x.3rd Qu. "50.00000" "47.00000" x.Max. "98.00000" "95.00000"	numeric
Job	Admin: 10422 Blue-collar: 9254 technician : 6743 services : 3969 management 2924 retired : 1720 (Other) : 6156	Type of Job (categorical) Values: 'admin.' ,'blue-collar' ,'entrepreneur ','housemaid', 'management', 'retired', ‘self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown'
marital	divorced: 4612 married : 24928 single : 11568 unknown : 80	Marital Status (categorical) Values: 'divorced', 'married', 'single', 'unknown' note: 'divorced' means divorced or widowed
education	university.degr :12168 high.school : 9515 basic.9y : 6045	Education (categorical) Values: 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'illiterate', 'professional.course', 'university.degree'

	Professi.course: 5243 basic.4y : 4176 basic.6y : 2292 (Other) : 1749	,'unknown'
default	no : 32588 unknown: 8597 yes : 3	Has Credit in Default (categorical) Values 'no', 'yes', 'unknown'
housing	no : 18622 unknown: 990 yes : 2 1576	Has housing loan? (categorical) Values: : 'no', 'yes', 'unknown'
loan	No: 33950 unknown: 990 yes : 6248	Has personal loan? (categorical) Values: : 'no', 'yes', 'unknown'
contact	cellular : 26144 Telephone: 15044	Contact communication type (categorical: 'cellular','telephone')
month		Last contact month of year (categorical) Values: 'jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug' , 'sep', 'oct',, 'nov', 'dec'
Day_of_ week		Last contact day of the week (categorical) Values: : 'mon', 'tue', 'wed', 'thu', 'fri'
duration	Group.1 "yes" "no" x.Min. " 37.0000" " 0.0000" x.1st Qu. " 253.0000" " 95.0000" x.Median " 449.0000" " 163.5000" x.Mean " 553.1912" " 220.8448" x.3rd Qu. " 741.2500" " 279.0000" x.Max. "4199.0000" "4918.0000"	Last contact duration, in seconds (numeric).
campaign	Group.1 "yes" "no" x.Min. " 1.000000" " 1.000000" x.1st Qu. " 1.000000" " 1.000000" x.Median " 2.000000" " 2.000000" x.Mean " 2.051724" " 2.633085" x.3rd Qu. " 2.000000" " 3.000000" x.Max. "23.000000" "56.000000"	Number of contacts performed during this campaign and for this client (numeric, includes last contact)
pdays	Min. : 0.0 1st Qu.: 999.0 Median : 999.0 Mean : 962.5 3rd Qu.: 999.0 Max. : 999.0	Number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
previous	Min. : 0.000 1st Qu.: 0.000 Median :0.000	Number of contacts performed before this campaign and for this client (numeric)

	Mean : 0.173 3rd Qu.: 0.000 Max. : 7.000	
poutcome	failure : 4252 Nonexistent: 35563 success : 1373	Outcome of the previous marketing campaign (categorical) Values: 'failure', 'nonexistent', 'success'
emp.var.rate	Group.1 "yes" "no" x.Min. "-3.4000" "-3.4000" x.1st Qu. "-1.8000" "-1.8000" x.Median "-1.8000" " 1.1000" x.Mean "-1.2334" " 0.2488" x.3rd Qu. "-0.1000" " 1.4000" x.Max. " 1.40000" " 1.4000"	Employment variation rate - quarterly indicator (numeric)
cons.price.idx	Group.1 "yes" "no" x.Min. "92.20100" "92.20100" x.1st Qu. "92.89300" "93.07500" x.Median "93.20000" "93.91800" x.Mean "93.35439" "93.60376" x.3rd Qu. "93.91800" "93.99400" x.Max. "94.76700" "94.76700"	Consumer price index - monthly indicator (numeric)
cons.conf.idx	Group.1 "yes" "no" x.Min. "-50.80000" "-50.80000" x.1st Qu. "-46.20000" "-42.70000" x.Median "-40.40000" "-41.80000" x.Mean "-39.78978" "-40.59310" x.3rd Qu . "-36.10000" "-36.40000" x.Max. "-26.90000" "-26.90000"	Consumer confidence index - monthly indicator (numeric)
euribor3m	Group.1 "yes" "no" x.Min. "0.634000" "0.634000" x.1st Qu. "0.849000" "1.405000" x.Median "1.266000" "4.857000" x.Mean "2.123135" "3.811491" x.3rd Qu. "4.406000" "4.962000" x.Max. "5.045000" "5.045000"	Euribor 3 month rate - daily indicator (numeric)
nr.employed	x.Min. "4963.600" "4963.600" x.1st Qu. "5017.500" "5099.100" x.Median "5099.100" "5195.800" x.Mean "5095.116" "5176.167" x.3rd Qu. "5191.000" "5228.100" x.Max. "5228.100" "5228.100"	Number of employees - quarterly indicator (numeric)

Objective 1 EDA/Feature selection

Tools such as Odds ratio, histograms, heat maps etc to analyze the data and lasso/forward selection techniques to select appropriate features were used to build models as follows.

EDA

Response y is unbalanced with 11% responding yes and 89% responding no. The additional market related variables helped boost the prediction so were included in further analysis, especially Emp.var.rate odds ratio listed below. Correlation plots and PCA Analysis show some correlation between continuous variables although in PCA there is significant overlap. The Proportions, BoxPlot and Odds ratio Histogram (normalization helped significantly on LDA) highlight the following

<table>
<tr>
<td>Education: The odds of 9y and 6y with respect to 4y is1. 27 and 1.34 times higher respectively.
basic.4y 1.0000000 NA NA
basic.6y 1.2780036 1.0678958 1.5294499
basic.9y 1.3452226 1.1728135 1.5429767</td>
<td>Previous Outcome: The odds of a previously successful customer are less by 8% than unsuccessful and for unknow the odds are 1.71 times higher.
failure 1.00000000 NA NA
unknown 1.71234581 1.55948111 1.8801947
Success 0.08888278 0.07723805 0.1022831</td>
</tr>
<tr>
<td>Job: The odds are double for Blue collar workers and about 1.6 higher for entrepreneurs and 1.68 more for service workers compared to Admin.
admin 1.0000000 NA NA
Bluecollar 2.0130493 1.8239432 2.2217619
entrp 1.6012235 1.3205930 1.9414889
services 1.6826103 1.4814508 1.9110844</td>
<td>Emp.var.rate: Odds are significantly higher when emp.var.rate increases.
-3.4 1.0000000 NA NA
-0.2 6.6223663 0.8360372 52.4566816
-0.1 10.9452998 9.1433360 13.1023937
1.1 23.0648433 19.3300854 27.5211924
1.4 13.0578041 11.3611343 15.0078543</td>
</tr>
<tr>
<td colspan="2">Duration: Box plot and number summary indicates as duration increases the yes response is higher, and average campaign is less for folks responding yes. T.test confirms this with a p-value of p-value of < 2.2e-16, also we log transformed it.</td>
</tr>
<tr>
<td colspan="2">Convert pdays value of 999 to -1 and log transform duration (lduration) to have it normally distributed.</td>
</tr>
</table>

Assumptions: During the EDA we noticed an unbalanced bias of positive outcomes (11% to 89%), so positive results were boosted. Models took into account market indicators which improved the predictability and should be used cautiously as these indicators tend to vary with time. For the LDA model we used logtrans formed duration/pdays to normalize the spread. For Logistic regression/Random Forest we opted not to use log transformed data as there is no assumption required for normalized spread and it becomes easier to interpret the model.

All Models used a common splitting mechanism of **80% train and 20% test**. The split was randomly selected using the same proportions of yes/no from the full dataset

Feature Selection

Forward: Our first feature selection technique we attempted was forward selection. Specifically we were looking for a model that produced the best AIC. We prioritized AIC at the start because that metric penalizes unhelpful parameters in the model. By trimming the features down at the start we were able to build from that both a highly predictive logistic model and a highly interpretable model. The forward model is:

$$ln(p(X)/1-p(X)) = \beta_0 + \beta_1 \text{ job} + \beta_2 \text{ contact} + \beta_3 \text{ default} + \beta_4 \text{ month} + \beta_5 \text{ day\_of\_week} + \beta_6 \text{ pdays} + \beta_7 \text{ poutcome} + \beta_8 \text{ emp.var.rate} + \beta_9 \text{ cons.conf.indx} + \beta_{10} \text{ nr.employed} + \beta_{11} \text{ cons.price.idx} + \beta_{12} \text{ euribor3m}$$

This model is still a little complex but it gives us a nice jumping off point. Next we performed a lasso selection to further penalize variables. AIC:

LASSO: The LASSO feature selection analysis resulted in selecting the 10 features listed here. After further analysis below, we decided that taking the log of the duration field smoothed the data out because it contained outliers. In order to account for values of 0 in duration, we added .1 to all values before taking the log.

Additional Variable Selection: We removed day_of_week from the final model. It is the day the customer was last contacted and doesn't have a valid business reason to impact the customer's default probability. We ran a Variable Importance Factor Analysis (VIF) and determined that it did not indicate any additional multicollinearity issues.

Unlogged duration	**Logged duration (Selected Model)**
Residual deviance: 13737 on 32913 degrees of freedom AIC: **13811**	Residual deviance: 12946 on 32913 degrees of freedom AIC: **13020**

Final Logistic Regression model (I) with no Interactions

$$ln(p(X)/1-p(X)) = \beta_0 + \beta_1 \text{ job} + \beta_2 \text{ education} + \beta_3 \text{ default} + \beta_4 \text{ month} + \beta_5 \, ln(\text{duration}) + \beta_6 \text{ pdays} + \beta_7 \text{ poutcome} + \beta_8 \text{ emp.var.rate} + \beta_9 \text{ cons.conf.indx} + \beta_{10} \text{ nr.employed}$$

Model Coefficients: The complete list of the coefficients are here. Below are couple.

	coef	Std err	z-score	p-value
(Intercept)	-38.1271765579882	3.29669619872448	-11.5652684565657	6.17977150134818E-31
jobblue-collar	0.301525948613872	0.0891539697586221	3.38208101591249	0.000719389250157612

nr.employed	0.010361985242364	0.000639020301099465	16.2154241806335	3.92359335404881E-59

Odds ratio for above Complete list is [here]

	Odds ratio	2.5%	97.5%
(Intercept)	2.76425185887642E-17	4.3194128748611E-20	1.76901087269817E-14
jobblue-collar	1.35192019513809	1.13517562972499	1.6100488472123
nr.employed	1.0104158565218	1.00915114649882	1.01168215153177

Interpretation: (Categorical) The odds of a positive sale is 35% more when a person has a blue collar job keeping other conditions same with a 95% confidence interval of (13.3 ~ 61 %).

(Continuous) The odds of a positive sale increases by 10.4 % for every 1000 point increase in the nr.employed index with a 95 % interval of (9 ~ 11.7)

Model Summary

Model	**Accuracy**	**Sensitivity**	**Specificity**	**CM**
Logistic I (Simple) 80% cutoff	89%	78%	90%	Truth p yes no yes 728 690 no 200 6620

Lack Of Fit Test

The lack of fit test for the model showed a very low p-value indicating a bad fit but we ignore this due to the large number of observations and use this model.

Objective 2

Logistic Regression Model II

The complex model is built for predictability and not for interpretability. The logistic model we came up with is:

$\mathit{ln}\,(p(X)/\ 1 - p(X)\ =\ \beta_0 + \beta_1$ month*day_of_week + β_2 age*duration + β_3 campaign + β_4 month + β_5 day_of_week + β_6 age + β_7 duration + β_8 emp.var.rate + β_9 pdays + β_{10} cons.price.idx + β_{11} cons.conf.idx

This model is more convoluted and harder to interpret than the other. The interaction terms and extra terms are to increase the amount of information our model is using and increase prediction power. The biggest challenge is to maximize our data without overfitting it. This model led to an increase in prediction accuracy as compared to the

LDA and simpler logistic model.

Model Summary

Model	Accuracy	Sensitivity	Specificity	CM
Logistic II (complex) 85% cutoff	88.3%	82.97%	88.9%	Truth pred yes no yes 770 809 no 153 6501

LDA Model

PCA Analysis on using continuous variables (previous, age, campaign, lduration, cons.price.idx) shows a potential separation which we can use for LDA.

LDA is performed (lduration, campaign, emp.var.rate, cons.price.idx, euribor3m, cons.conf.idx, nr.employed), LDA plot here. The ROC Curve indicated that to have a Sensitivity of 70% we need to balance on yes / no to 90%/10%. **Increasing Sensitivity is critical in this case as we do not want to lose a potential sale opportunity.**

Model Summary

Model	Accuracy	Sensitivity	Specificity	CM
LDA 90% cutoff	85.4%	76%	86.6%	Truth pred yes no yes 707 982 no 221 6328

Random Forest

As we can see, the random forest model provided some solid results. First, it requires us to adjust the hyperparameter, mtry. This parameter is the number of variables randomly sampled at each node or split. Mtry was tuned by cross validation on the training set by tuning values. Based on out-of-bag (OOB) error, mtry was set to 11 for the random forest model. As indicated in, the AUC for the Random Forest training set is 0.946. The ROC curve indicated that to have a sensitivity of 88.5%

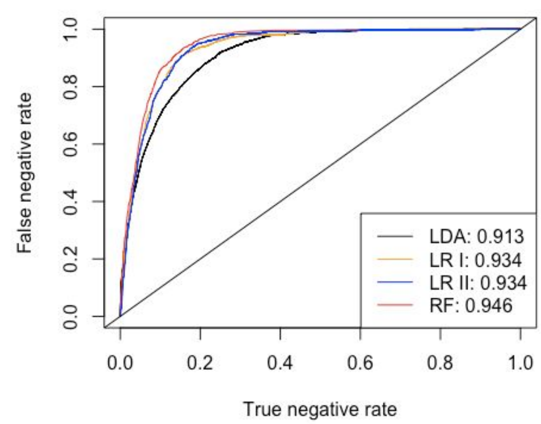
Model Summary

Model	Accuracy	Sensitivity	Specificity	CM
RM 80% cutoff	88%	88.5%	87%	Truth p yes no yes 821 885 no 107 6425

Conclusion And Summary

Model	AUC	Accuracy	Sensitivity	Specificity	Pr Cutoff	CM
LDA	91.3	85.4%	76%	86.6%	90	Truth pred yes no yes 707 982 no 221 6328
LR Simple	93.4	89%	78%	90%	80	Truth p yes no yes 728 690 no 200 6620
LR Complex	93.4	88.3%	82.97%	88.9%	85	Truth pred yes no yes 770 809 no 153 6501
RM	94.6	88%	88.5%	88%	80	Truth p yes no yes 821 885 no 107 6425

Of the four models Random Forest performed the best for prediction of new deposits with an overall accuracy of 94%, as the random forest algorithm consists of many decision trees and uses bagging as well as continuous/categorical feature at random (mtry) when building each individual tree to create an uncorrelated forest of trees for prediction.

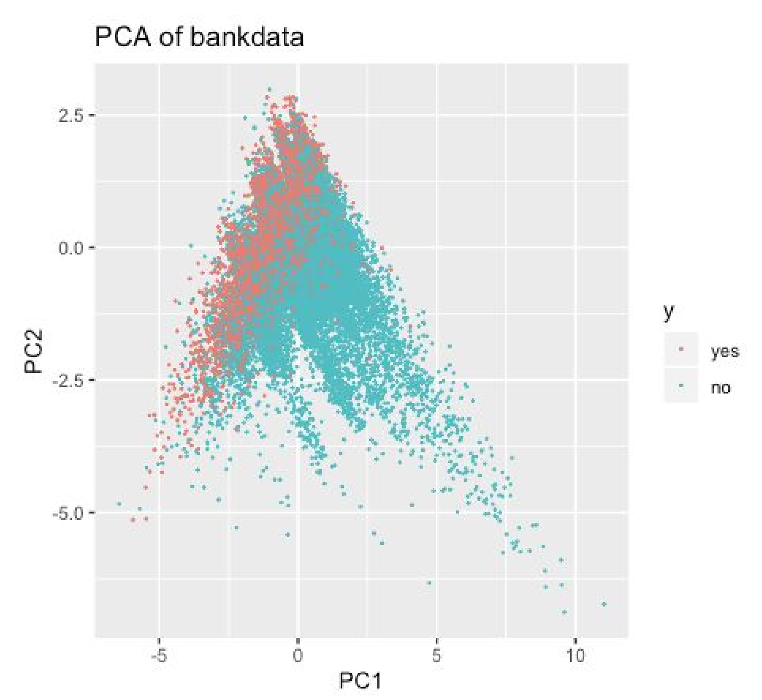


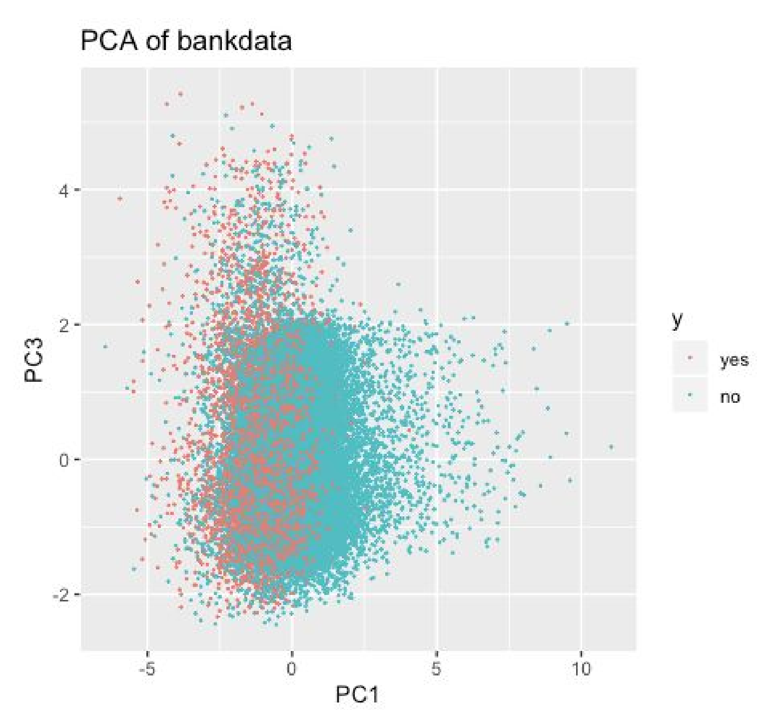
The simpler Logistic Regression (I) model allows better interpretation of factors that influence the sales of new deposits based on the customer profile and market conditions. The interpretation is easier to valida with the EDA done. The complex Logistic Regression model or the random forest would be harder to do the same interpretation.

LDA model was not upto the mark as many of the critical predictors with this data set were categorical, which do not perform very well in an LDA model. Although [PCA analysis] showed separation a closer look indicates overlap of negatives with positives.

APPENDIX

PCA Analysis





Odds Ratio

Education

	Response		
Treatment	Yes	No	Total
basic.4y	428	3748	4176
basic.6y	188	2104	2292
basic.9y	473	5572	6045
HS	1031	8484	9515
illitrate	4	14	18
professional	595	4648	5243
degree	1670	10498	12168
unknown	251	1480	1731
Total	4640	36548	41188

	odds ratio with 95% C.I.		
Treatment	estimate	lower	upper
basic.4y	1.0000000	NA	NA
basic.6y	1.2780036	1.0678958	1.5294499
basic.9y	1.3452226	1.1728135	1.5429767
HS	0.9396934	0.8342039	1.0585226
illitrate	0.3996798	0.1309713	1.2196871
professional	0.8920585	0.7821659	1.0173908
degree	0.7178510	0.6414405	0.8033637
unknown	0.6733365	0.5697575	0.7957457

	two-sided		
Treatment	midp.exact	fisher.exact	chi.square
Basic.4y	NA	NA	NA
basic.6y	6.862098e-03	7.842591e-03	7.317806e-03
basic.9y	2.414000e-05	2.378896e-05	2.140607e-05
HS	3.059994e-01	3.208936e-01	3.058580e-01
illitrate	1.376313e-01	1.063542e-01	9.539739e-02
professional	8.824505e-02	8.926469e-02	8.845913e-02
degree	3.414443e-09	4.114282e-09	6.896878e-09
unknown	4.787954e-06	4.680671e-06	3.122762e-06

Loan

	Response		
Treatment	Yes	No	Total
no	3850	30100	33950
unknown	107	883	990
yes	683	5565	6248
Total	4640	36548	41188

	odds ratio with 95% C.I.		
Treatment	estimate	lower	upper
no	1.000000	NA	NA
unknown	1.055531	0.8612505	1.293638

```
 yes     1.042170 0.9560448 1.136055

        two-sided
Treatment midp.exact fisher.exact chi.square
 no              NA             NA            NA
 Unknown   0.6093112   0.6470181   0.6025085
 yes         0.3482678   0.3607284   0.3479236
```

Housing

```
        Response
Treatment  Yes      No       Total
 no          2026   16596    18622
 unknown  107       883        990
 yes         2507  19069      21576
 Total       4640  36548      41188

        odds ratio with 95% C.I.
Treatment    estimate          lower        upper
 no           1.0000000           NA            NA
 Unknown  1.0074255     0.8199887    1.2377075
 yes          0.9285592     0.8726153    0.9880897

        two-sided
Treatment midp.exact fisher.exact chi.square
 no                          NA             NA              NA
 unknown     0.95364204   1.00000000    0.94384730
 yes           0.01931128   0.02010454    0.01937352
```

Previous Outcome

```
        Response
Treatment   Yes       No     Total
 failure     605     3647     4252
 Unknown  3141   32422    35563
 success     894      479      1373
 Total       4640    36548    41188

        odds ratio with 95% C.I.
Treatment           estimate       lower            upper
 failure         1.00000000          NA               NA
 unknown       1.71234581     1.55948111     1.8801947
 success        0.08888278     0.07723805     0.1022831

        two-sided
Treatment midp.exact  fisher.exact    chi.square
 failure        NA           NA           NA
 unknown      0  5.185674e-27  4.623383e-30
 success       0 2.512956e-277 6.454682e-301
```

Default

Treatment	Yes	No	Total
NoDef	4197	28391	32588
unknown	443	8154	8597
Default	0	3	3
Total	4640	36548	41188

odds ratio with 95% C.I.

Treatment	estimate	lower	upper
NoDef	1.000000	NA	NA
unknown	2.720979	2.459679	3.010039
Default	Inf	NaN	Inf

two-sided

Treatment	midp.exact	fisher.exact	chi.square
NoDef	NA	NA	NA
unknown	0.000000	4.445019e-105	2.498371e-90
Default	0.661273	1.000000e+00	5.054478e-01

Job

Treatment	Yes	No	Total
admin	1352	9070	10422
bluecollar	638	8616	9254
entrp	124	1332	1456
household	106	954	1060
mgmt	328	2596	2924
retired	434	1286	1720
selfempl	149	1272	1421
services	323	3646	3969
student	275	600	875
tech	730	6013	6743
unempyl	144	870	1014
unknown	37	293	330
Total	4640	36548	41188

odds ratio with 95% C.I.

Treatment	estimate	lower	upper
admin	1.0000000	NA	NA
Bluecollar	2.0130493	1.8239432	2.2217619
entrp	1.6012235	1.3205930	1.9414889
household	1.3415656	1.0889291	1.6528149
mgmt	1.1797779	1.0377384	1.3412588
retired	0.4416931	0.3906140	0.4994516
selfempl	1.2725365	1.0638996	1.5220883
services	1.6826103	1.4814508	1.9110844
student	0.3252280	0.2788812	0.3792772

tech	1.2278286	1.1157291	1.3511909
unempyl	0.9005880	0.7482156	1.0839907
unknown	1.1804166	0.8345775	1.6695672

two-sided

Treatment	midp.exact	fisher.exact	chi.square
admin	NA	NA	NA
bluecollar	0.000000e+00	2.729347e-46	3.120888e-45
entrp	4.805827e-07	5.306567e-07	1.378229e-06
household	4.530089e-03	4.958458e-03	5.617877e-03
mgmt	1.070385e-02	1.159229e-02	1.146458e-02
retired	0.000000e+00	1.044831e-35	2.327082e-40
selfempl	7.069561e-03	8.353710e-03	8.204699e-03
services	0.000000e+00	9.060952e-17	6.361062e-16
student	0.000000e+00	4.359868e-41	1.964089e-50
tech	2.336320e-05	2.499718e-05	2.587670e-05
unempyl	2.694425e-01	2.622230e-01	2.680245e-01
unknown	3.505887e-01	4.041298e-01	3.478713e-01

emp.var.rate

Response

Treatment	Yes	No	Total
-3.4	454	617	1071
-3	88	84	172
-2.9	594	1069	1663
-1.8	1461	7723	9184
-1.7	403	370	773
-1.1	301	334	635
-0.2	1	9	10
-0.1	232	3451	3683
1.1	240	7523	7763
1.4	866	15368	16234
Total	4640	36548	41188

odds ratio with 95% C.I.

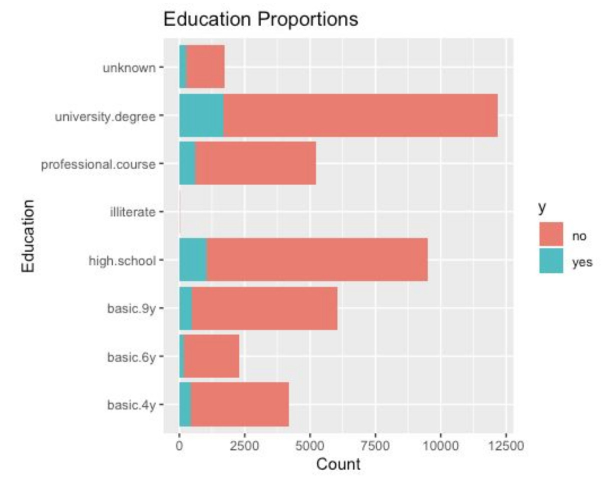
Treatment	estimate	lower	upper
-3.4	1.0000000	NA	NA
-3	0.7023722	0.5087016	0.9697762
-2.9	1.3242255	1.1314645	1.5498261
-1.8	3.8896140	3.4036226	4.4449984
-1.7	0.6755654	0.5608952	0.8136788
-1.1	0.8164896	0.6702521	0.9946336
-0.2	6.6223663	0.8360372	52.4566816
-0.1	10.9452998	9.1433360	13.1023937
1.1	23.0648433	19.3300854	27.5211924
1.4	13.0578041	11.3611343	15.0078543

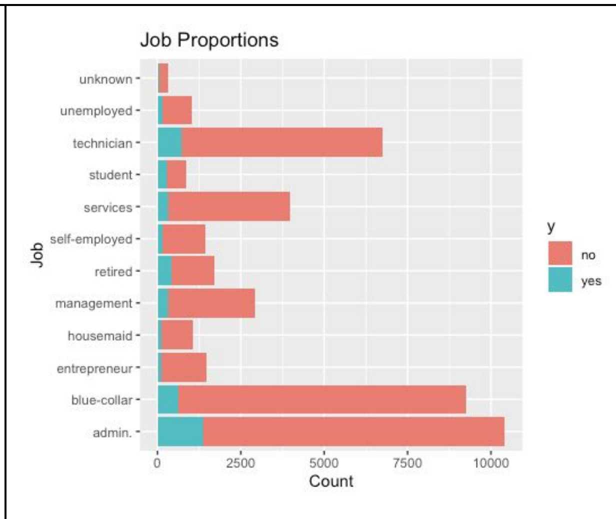
```
      two-sided
Treatment   midp.exact  fisher.exact    chi.square
    -3.4          NA           NA           NA
    -3   3.235779e-02  3.815583e-02  3.127379e-02
    -2.9 4.764073e-04  5.270172e-04  4.610587e-04
    -1.8 0.000000e+00  1.046073e-81  2.499875e-98
    -1.7 3.541249e-05  3.819646e-05  3.478245e-05
    -1.1 4.434708e-02  4.921799e-02  4.396230e-02
    -0.2 3.856410e-02  5.147350e-02  3.891873e-02
    -0.1 0.000000e+00 4.620380e-161 2.326047e-192
    1.1  0.000000e+00 4.137114e-276  0.000000e+00
    1.4  0.000000e+00 4.729083e-243  0.000000e+00
```

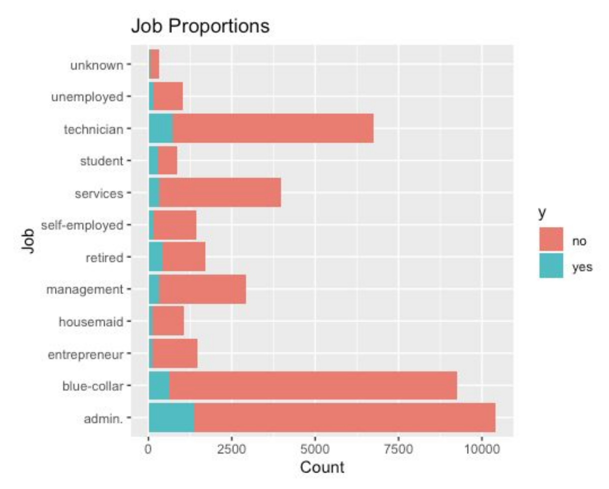
Marital

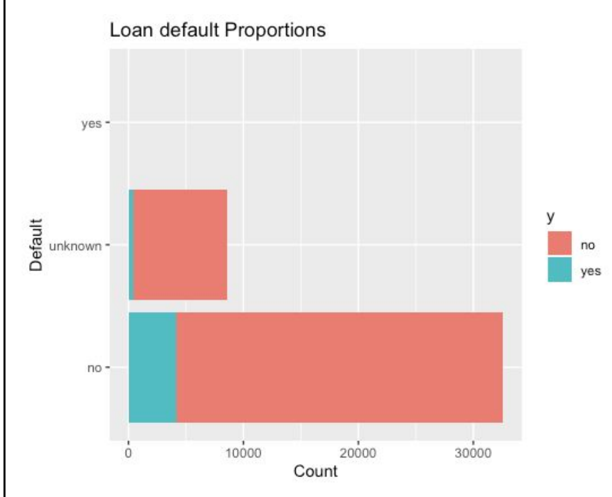
```
$data
       Response
Treatment  Yes No Total
  divorced     4    2     6
  married      3    1     4
  single        2    4     6
  unknown   1    3     4
  Total        10   10    20

$measure
       odds ratio with 95% C.I.
Treatment   estimate     lower    upper
  divorced   1.0000000        NA       NA
  married    0.6666667 0.03938267  11.28528
  single       4.0000000 0.36270644  44.11281
  unknown  6.0000000 0.35444402 101.56752

$p.value
       two-sided
Treatment  midp.exact fisher.exact chi.square
  divorced        NA          NA        NA
  married    0.8333333    1.0000000  0.7781597
  single       0.3235931    0.5670996  0.2482131
  unknown   0.2857143    0.5238095  0.1967056
```

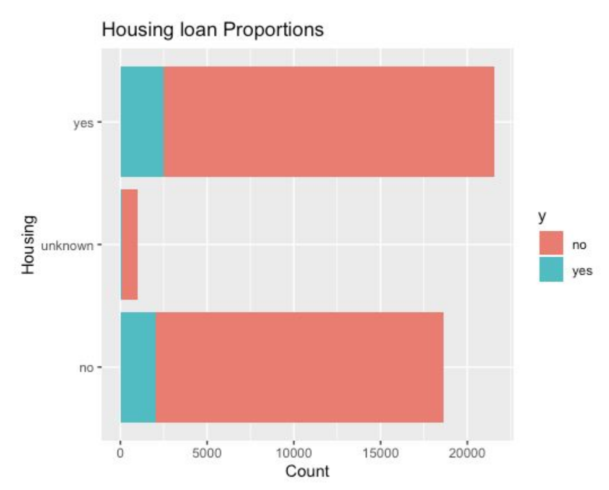
Proportion Plots

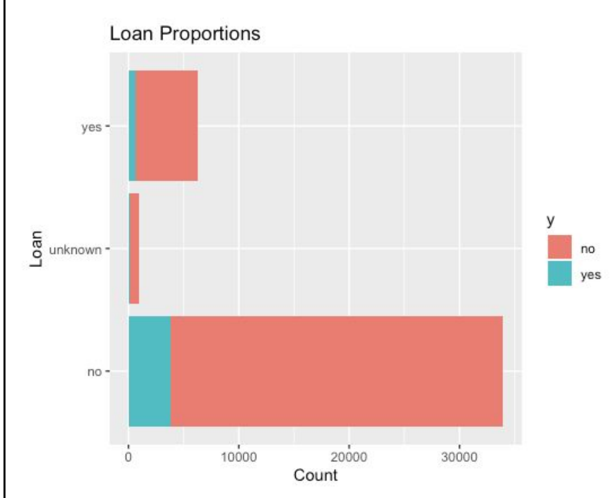


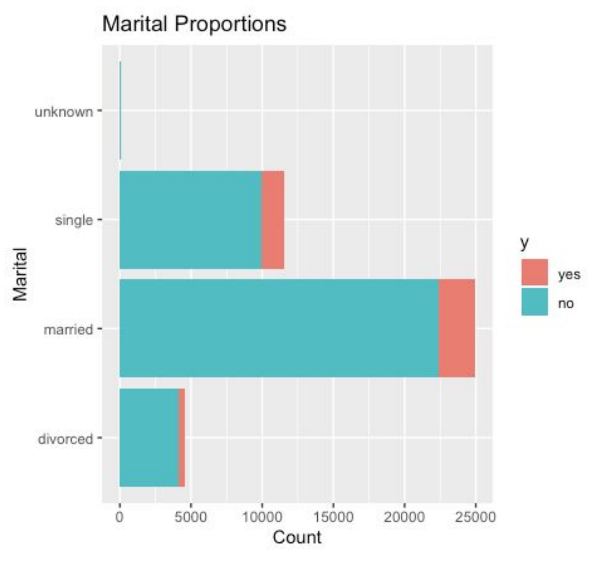








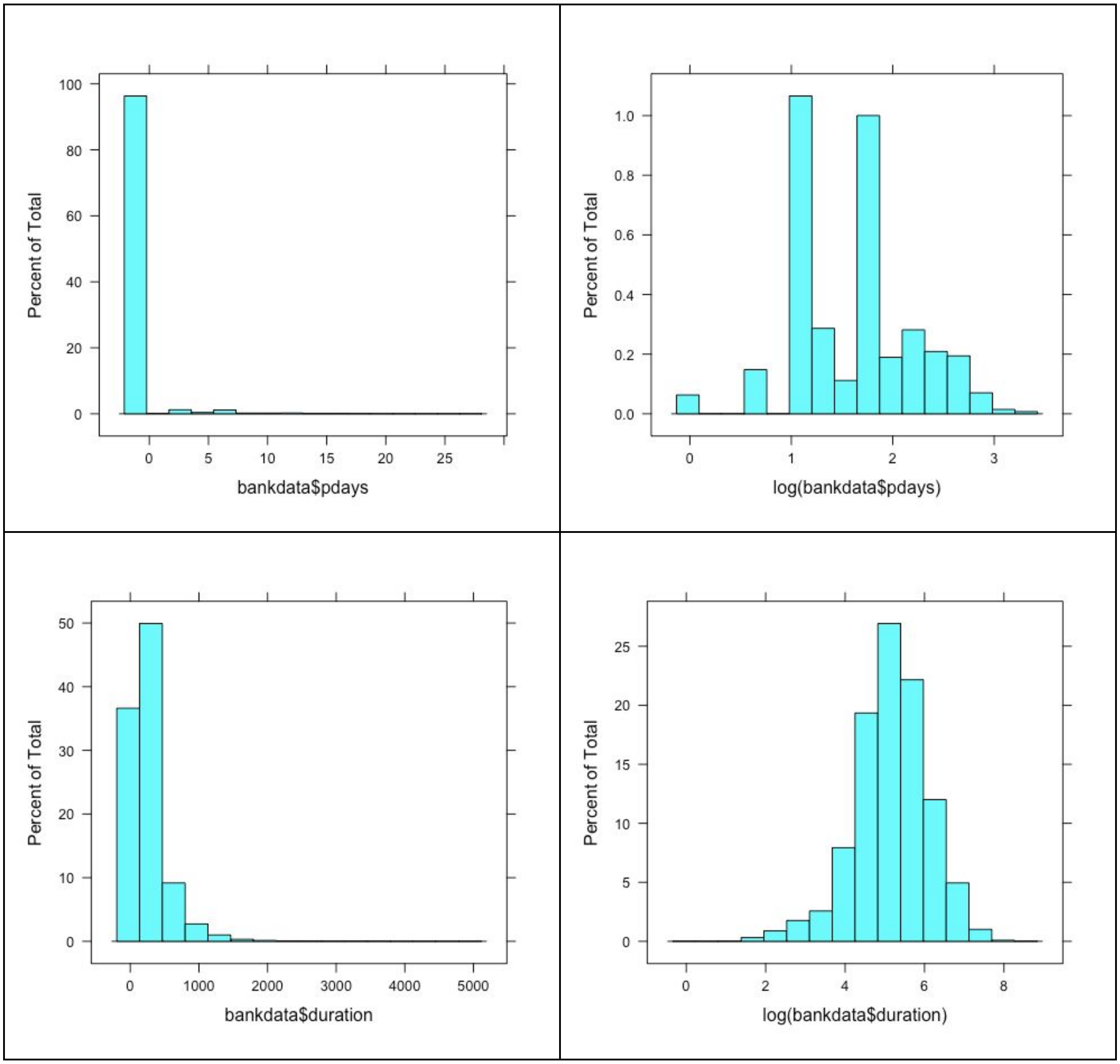




Code for Portion Plots

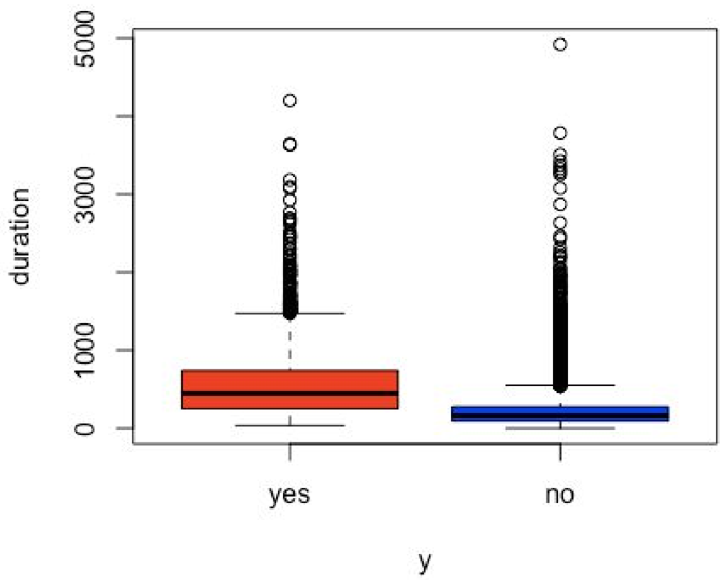
```
#EDA plots for portion
bankdata %>% ggplot(aes(x=education, fill=y)) +
  geom_histogram(stat='count') + labs(title="Education Proportions",
                              y = "Count", x="Education" ) + coord_flip()
bankdata %>% ggplot(aes(x=job, fill=y)) +
  geom_histogram(stat='count') + labs(title="Job Proportions",
                              y = "Count", x="Job" ) + coord_flip()
bankdata %>% ggplot(aes(x=default, fill=y)) +
  geom_histogram(stat='count') + labs(title="Loan default Proportions",
                              y = "Count", x="Default" ) + coord_flip()
bankdata %>% ggplot(aes(x=housing, fill=y)) +
  geom_histogram(stat='count') + labs(title="Housing loan Proportions",
                              y = "Count", x="Housing" ) + coord_flip()
bankdata %>% ggplot(aes(x=loan, fill=y)) +
  geom_histogram(stat='count') + labs(title="Loan Proportions",
                              y = "Count", x="Loan" ) + coord_flip()
```

Histogram (Before / after Xformation)

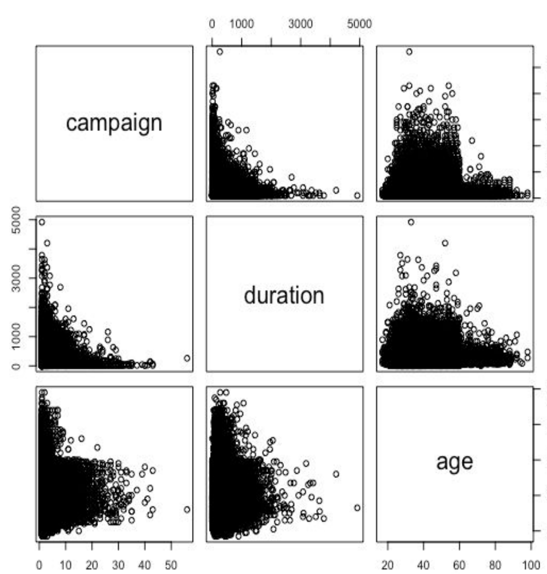


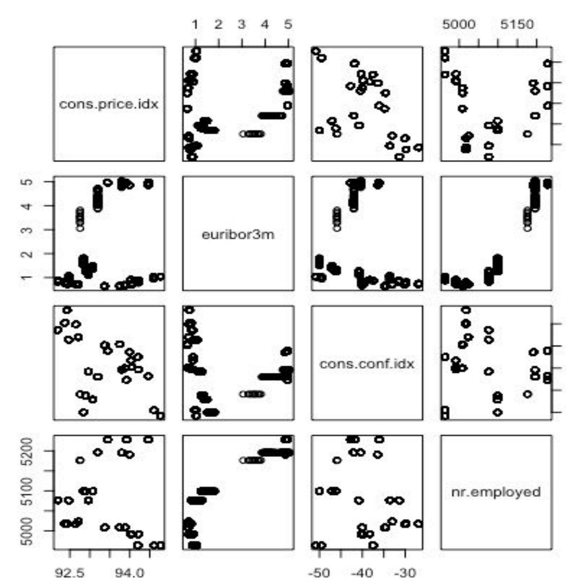
Percent of Total
bankdata$pdays
log(bankdata$pdays)
bankdata$duration
log(bankdata$duration)

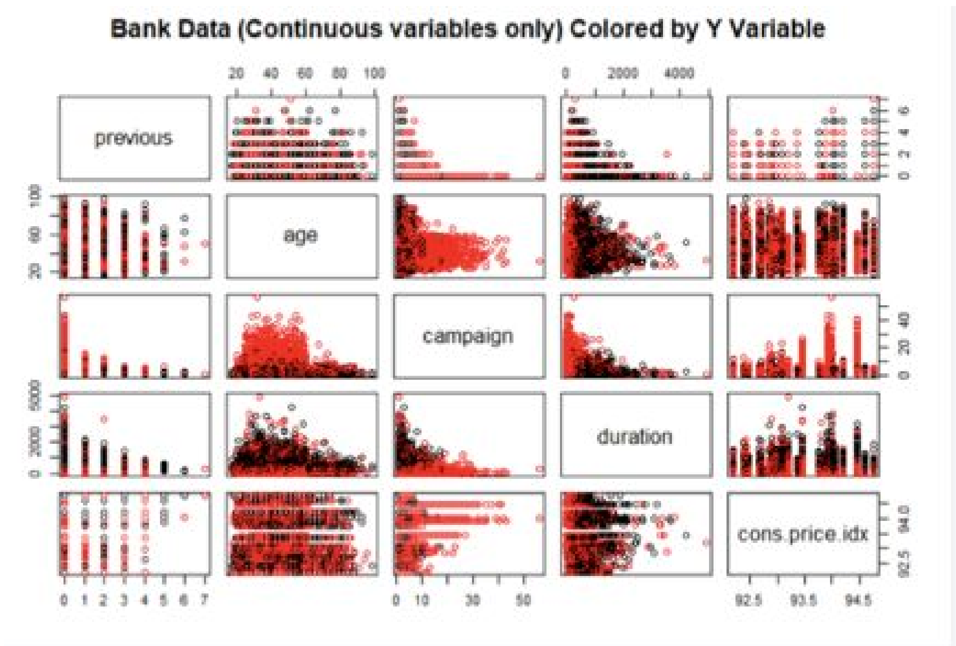
Box Plot

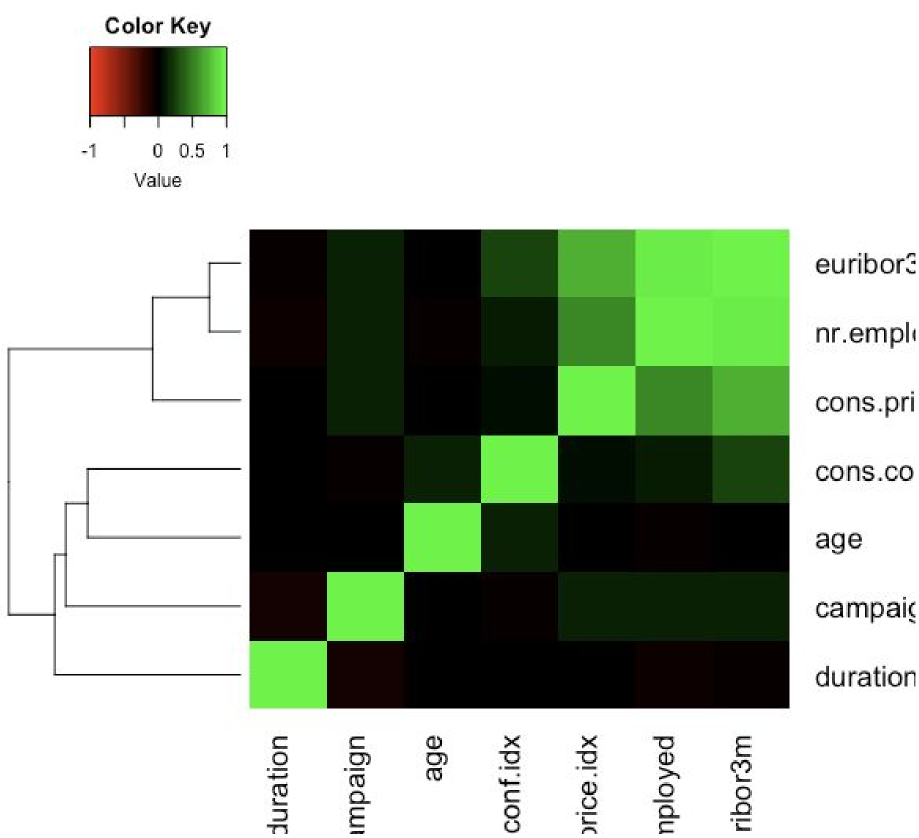


Correlation Plot Of Continuous Variables



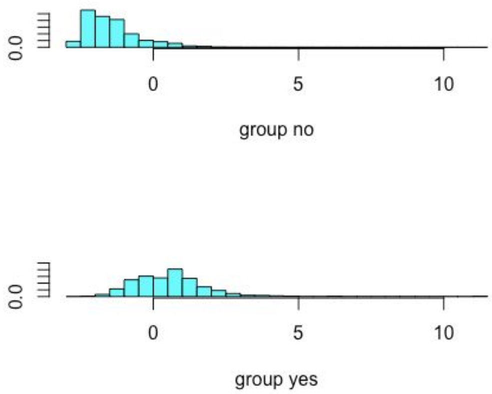



Bank Data (Continuous variables only) Colored by Y Variable
previous
age
campaign
duration
cons.price.idx

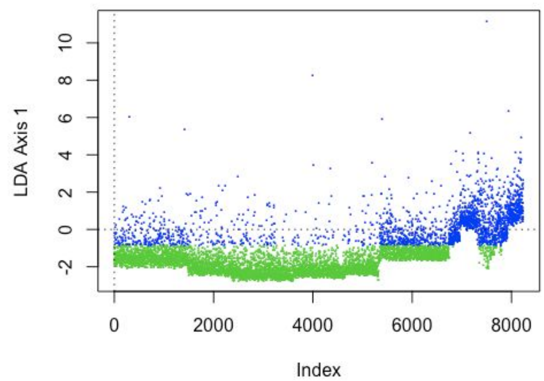

Color Key
-1
0
0.5
1
Value
euribor3
nr.empl
cons.pri
cons.co
age
campaig
duration
duration
mpaign
age
conf.idx
price.idx
nployed
ribor3m

LDA Separation Plot

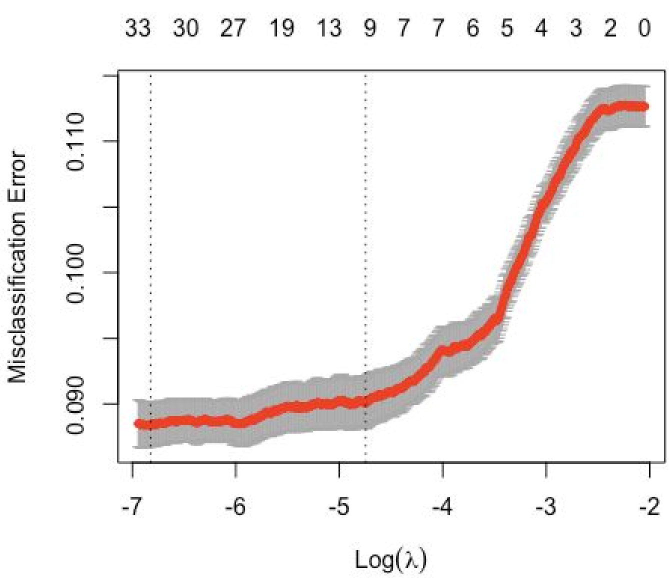
Histogram of test set 80%



Plot of LDA1 component vs observation



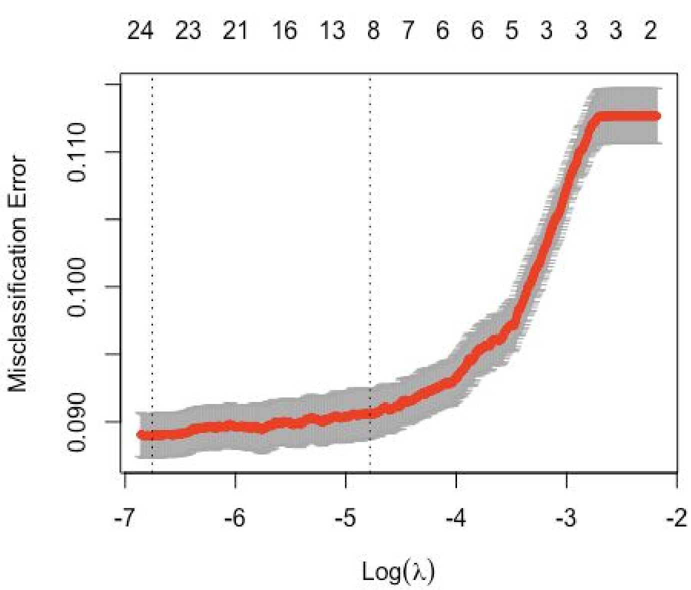
LASSO Feature Selection with all variables



Plot with all variables
Min Lambda: 0.001602756
Coefficient
(Intercept) -40.927238256

jobblue-collar	0.051904030
monthmar	-0.961838597
monthmay	0.704319465
duration	-0.001060742
poutcomesuccess	-1.348045851
emp.var.rate	0.121429717
cons.price.idx	.
cons.conf.idx	-0.004573949
euribor3m	.
nr.employed	0.009802255

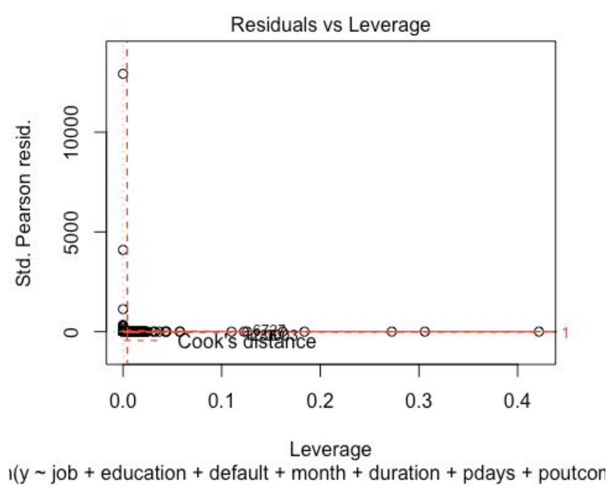
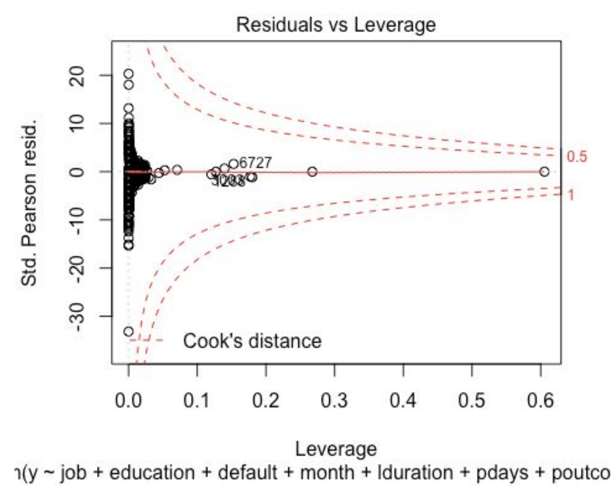
LASSO Plot with significant variables

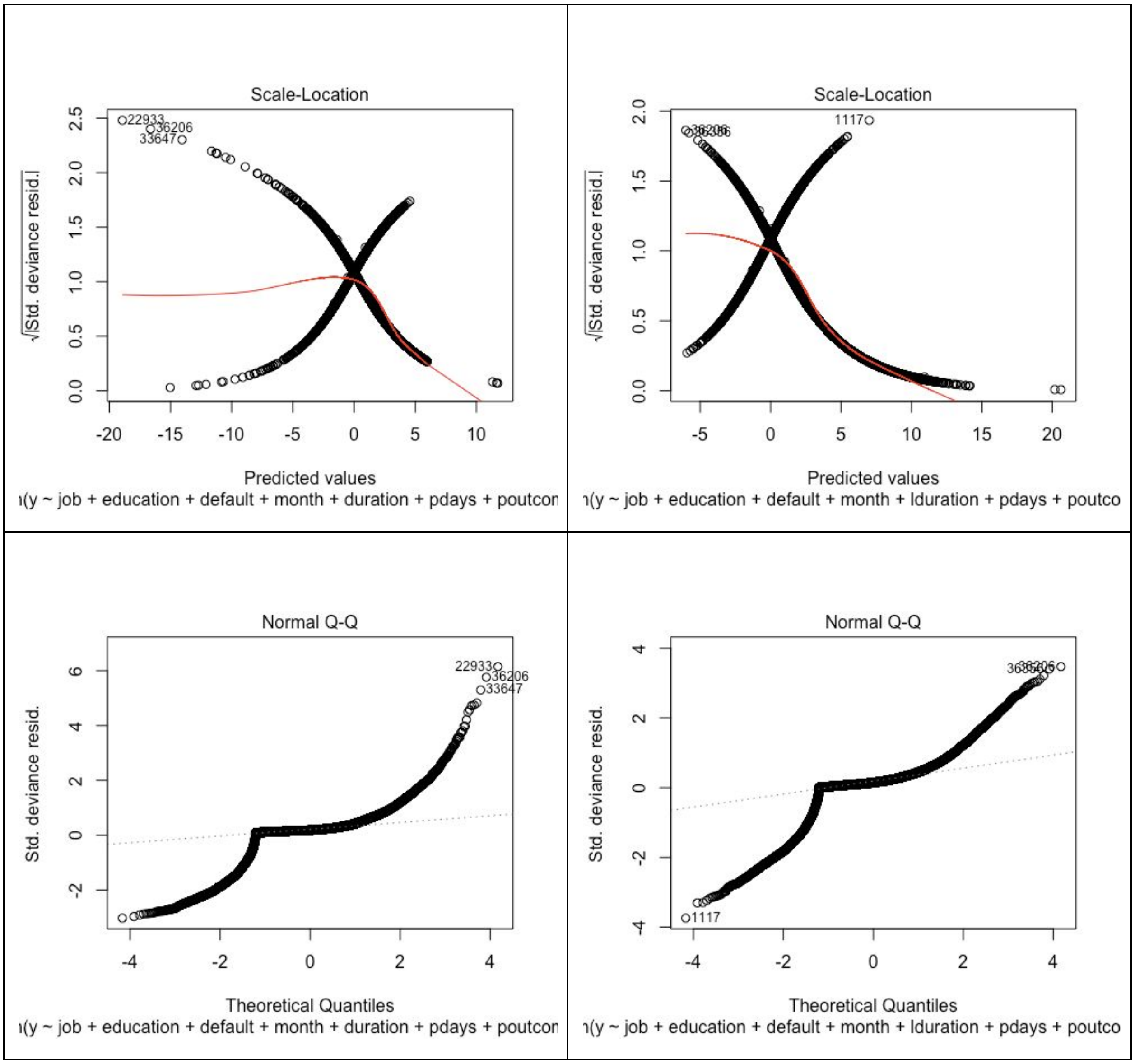


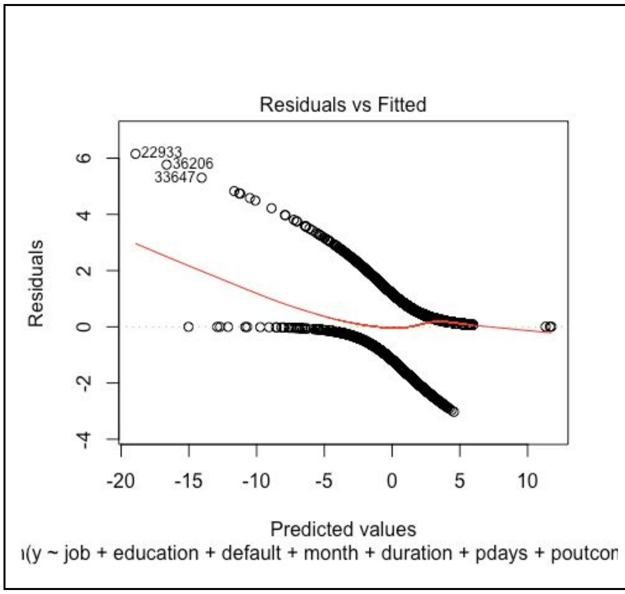
Plot with significant variables (Selected model)
Min Lambda: 0.001632584
Coefficients:

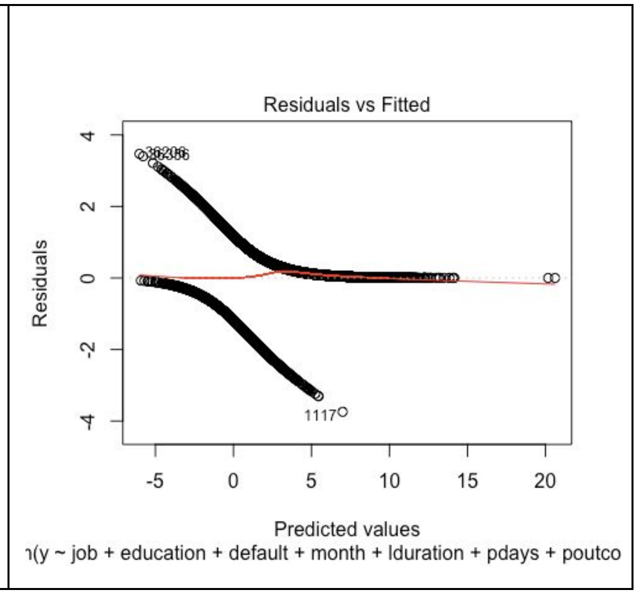
(Intercept)	-47.829667030
jobblue-collar	0.045243803
jobretired	-0.010516931
monthmar	-0.886586432
monthmay	0.672462611
duration	-0.004
019688	
poutcomesuccess	-1.357381492
emp.var.rate	0.127125642
cons.conf.idx	-0.001022818
nr.employed	0.009959558

LASSO plots to compare models with lduration

Unloged Duration (AIC 13811)	Loged duration (AIC: 13020)
	


Scale-Location
√|Std. deviance resid.|
Predicted values
ı(y ~ job + education + default + month + duration + pdays + poutcon
22933
36206
33647
Scale-Location
1117
Predicted values
ı(y ~ job + education + default + month + lduration + pdays + poutco
Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
ı(y ~ job + education + default + month + duration + pdays + poutcon
Normal Q-Q
Theoretical Quantiles
ı(y ~ job + education + default + month + lduration + pdays + poutco





LASSO Simple Model Coefficients

(Intercept)	-38.1271765579882	3.29669619872448	-11.5652684565657	6.17977150134818E-31
jobblue-collar	0.301525948613872	0.0891539697586221	3.38208101591249	0.000719389250157612
jobentrepreneur	0.192328437708624	0.138873046749135	1.38492272050496	0.166076098811776
jobhousemaid	-0.0258987236642298	0.161187759856626	-0.160674257693427	0.872349965360261
jobmanagement	0.0205254647669419	0.0951706341985803	0.215670148042873	0.829244872010658
jobretired	-0.289698103398291	0.106277499315421	-2.72586488451799	0.00641332458223137
jobself-employed	0.254150633651443	0.136525901938481	1.86155615925514	0.0626656777383371
jobservices	0.17993426506937	0.0957573747622527	1.87906430722555	0.0602357130173233
jobstudent	-0.238058636819166	0.12487233505079	-1.90641615472586	0.0565962344214376
jobtechnician	0.0542956820091201	0.0806458972955719	0.67326031242635	0.500781696908322
jobunemployed	-0.030236197811495	0.14691280214012	-0.205810503720818	0.836938936064014
jobunknown	0.0930628528320411	0.280723477632275	0.331510757906561	0.74025871634227
educationbasic.6y	-0.156835636875934	0.132880709231791	-1.18027392977228	0.237891283460038
educationbasic.9y	-0.0721359627338958	0.105032553769389	-0.686796237405393	0.492211141176713
educationhigh.school	-0.0643270551002235	0.101946748070557	-0.63098682711981	0.528049137607345
educationilliterate	-1.96231959732695	0.842559523197666	-2.32899818149297	0.0198591620399206
educationprofessional.course	-0.127663409945473	0.114077317421132	-1.11909547692278	0.263099409263502
educationuniversity.degree	-0.204958184827845	0.102241269620645	-2.00465218779383	0.0450002432705523
educationunknown	-0.15581868641533	0.135566470994815	-1.14938955976283	0.250395383099233
defaultunknown	0.247204511890438	0.0721072436129707	3.42828957957798	0.000607397190797454
defaultyes	6.24775983494382	111.09751769365	0.0562367185572224	0.95515323025491
monthaug	-0.544830799177488	0.112890902321565	-4.82617100203134	1.39183014580649E-06
monthdec	-0.194828488699295	0.234794677864553	-0.829782388899319	0.406661829681739
monthjul	-0.508300527311468	0.101085634460851	-5.02841506632007	4.94550475605961E-07

monthjun	-0.498701639030732	0.0996545298405914	-5.00430476997344	5.6063998090923E-07
monthmar	-1.73450314150615	0.143587122850758	-12.0797959250773	1.35054503872453E-33
monthmay	0.69365806722682	0.0844146683480911	8.2172693537865	2.08189118566575E-16
monthnov	0.0758546985461704	0.106917145829632	0.709471787313154	0.478031753146433
monthoct	-0.358317538650363	0.147446150828014	-2.43015864868738	0.0150922149839444
monthsep	-0.0698155151813263	0.161366051534118	-0.432653055073142	0.665266839439887
lduration	-2.23048346499739	0.0388937422472897	-57.3481320160912	0
pdays	-0.0181389436981929	0.0153293110614431	-1.18328499079236	0.236696219190689
poutcomenonexistent	-0.494458077647385	0.0734534245503573	-6.73158645324154	1.67823067413197E-11
poutcomesuccess	-1.82611269983879	0.131102297056199	-13.9289146021294	4.22730812076736E-44
emp.var.rate	0.259652129184094	0.0285565969614239	9.09254451904226	9.6747827966631E-20
cons.conf.idx	-0.0074533443807301	0.00534005254557788	-1.39574364055691	0.162791707553395
nr.employed	0.010361985242364	0.000639020301099465	16.2154241806335	3.92359335404881E-59

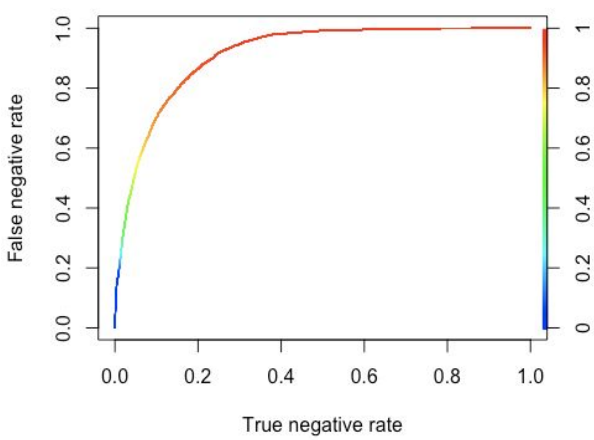
LASSO Final Simple model Odds ratio

	Odds ratio	**2.5%**	**97.5%**
(Intercept)	2.76425185887642E-17	4.3194128748611E-20	1.76901087269817E-14
jobblue-collar	1.35192019513809	1.13517562972499	1.6100488472123
jobentrepreneur	1.21206854061251	0.92324768532877	1.59124162506769
jobhousemaid	0.974433771693433	0.710475419763334	1.33645886825048
jobmanagement	1.02073756075634	0.847041532546868	1.23005204338212
jobretired	0.748489499909185	0.607746044388376	0.921826701542934
jobself-employed	1.28936601155073	0.986654632591004	1.68495100192913
jobservices	1.19713866670808	0.99228318794407	1.44428627305169
jobstudent	0.78815647475011	0.617050805188338	1.00670904805143
jobtechnician	1.05579673610676	0.901434971874563	1.23659141563548
jobunemployed	0.970216343505468	0.72747183251828	1.29396041348648
jobunknown	1.097530716014	0.633088502052431	1.90269396567628
educationbasic.6y	0.854844552192593	0.658838369278988	1.10916310052354
educationbasic.9y	0.930404386887081	0.757299861820174	1.14307735519471
educationhigh.school	0.937698270373	0.767866791336154	1.14509190419668
educationilliterate	0.14053206475817	0.026951780483653	0.732762766347587
educationprofessional.course	0.880149578887653	0.703807186222352	1.10067543551818
educationuniversity.degree	0.814681381741415	0.66674514934609	0.995441443266637
educationunknown	0.855714328938662	0.656046185968662	1.11615161921835
defaultunknown	1.28044095140237	1.11168503231329	1.47481434252697
defaultyes	516.853689251938	1.40277853698365E-92	1.9043471870318E+97
monthaug	0.579939901347104	0.464825644436628	0.723562250060715

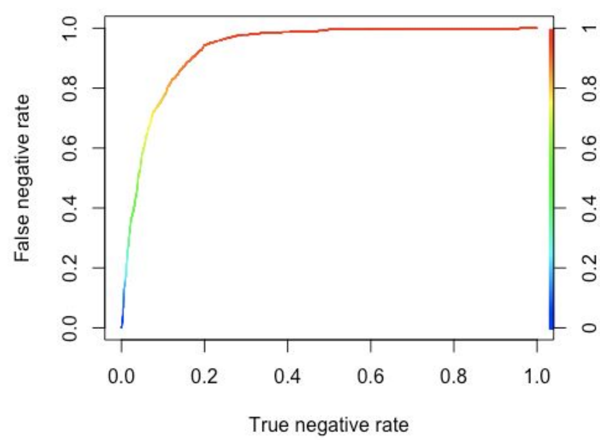
monthdec	0.822975795601483	0.519432919928546	1.30390110861488
monthjul	0.601516972317392	0.493405104090408	0.733317643020538
monthjun	0.607318666896056	0.499563316875944	0.738316747248267
monthmar	0.176487868001589	0.133196642062133	0.233849495524195
monthmay	2.00102203438339	1.6958911544299	2.36105316760961
monthnov	1.07880581071588	0.874853306310915	1.33030528528487
monthoct	0.6988511275261	0.523453706188012	0.933020232106413
monthsep	0.93256584827857	0.679711276393965	1.27948305637857
lduration	0.107476456468336	0.0995879805357026	0.115989787450797
pdays	0.982024576752287	0.952958577841785	1.0119771118799
poutcomenonexistent	0.60990133692199	0.528124015873507	0.704341460715405
poutcomesuccess	0.161038357047995	0.124547453839732	0.208220655189698
emp.var.rate	1.29647900100267	1.22590853168057	1.3711119195301
cons.conf.idx	0.992574362910437	0.982239947211808	1.00301750982911
nr.employed	1.0104158565218	1.00915114649882	1.01168215153177

ROC Curves With Cutoff

LDA

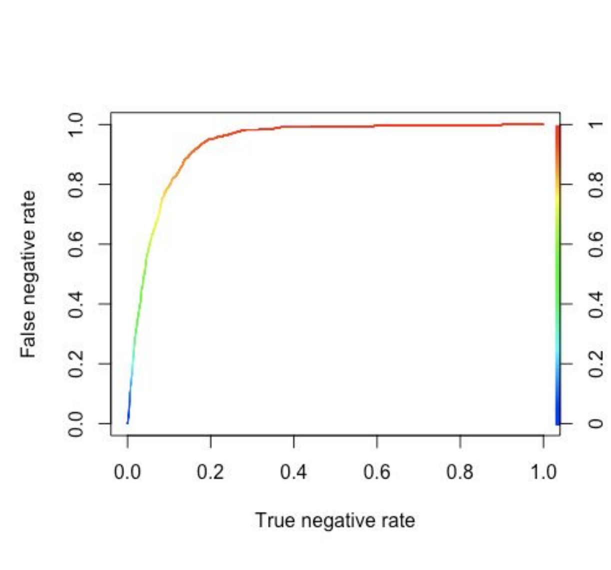


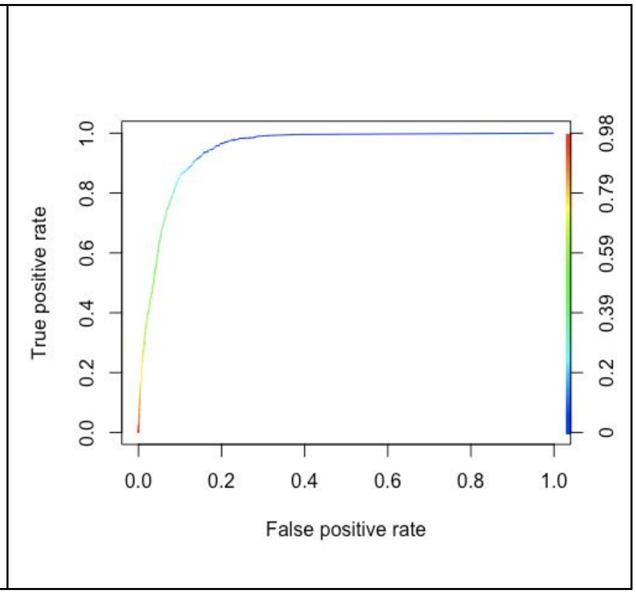
LR Simple



Complex LR

Random Forest





Lack of Fit Result

```
        Hosmer and Lemeshow goodness of fit (GOF) test

data:  lasso.finalmodel$y, fitted(lasso.finalmodel)
X-squared = 405.57, df = 8, p-value < 2.2e-16
```

Code Train/Test split

```
# Read the data
setwd("/Users/sanjaypillay/MS_DS/stat-applied/Project2")
bankdata<- read.csv("data/bank-additional-full.csv", sep=";")
# adjust durtion for 0 so when log transformed it does no tend up in Inf value
bankdata = bankdata %>% mutate(duration=duration + .1)

# log transform duration, adjust for pdays of 999
bankdata = bankdata %>% mutate(lduration=log(duration), pdays = ifelse(pdays == 999, -1, pdays))

#flip factor for y to make it easy for analysis
bankdata$y  <-relevel(bankdata$y, ref = "yes")
#split datasets yes/no
bankY = bankdata %>%  filter(y == "yes")
bankN = bankdata %>%  filter(y == "no")

#Train and Test Split 80%/20%, with a seed of 10 so all members of the group can use to compare results on the same
basis
#The split wa done using the bankdataN and bankdataY by taking the sampe propertion (keeping it unbalanced)
# of yes and no as the original data set.
set.seed(10)
trainInd = sample(seq(1,dim(bankY)[1],1),round(.8*dim(bankY)[1]))
trainY = bankY[trainInd,]
testY = bankY[-trainInd,]
trainInd = sample(seq(1,dim(bankN)[1],1),round(.8*dim(bankN)[1]))
train = bankN[trainInd,]
test = bankN[-trainInd,]
train = rbind(train,trainY)
test = rbind(test,testY)
table(test$y)
table(train$y)
```

Odds Ratio code

```
#Common method for calculating odds ration
getOddsRation <- function(dmxl,colName){
  d1 = dmxl %>% filter(y=='no')
  d2 = dmxl %>% filter(y=='yes')
  x = as.character(eval(substitute(colName), d2))
  d3=as.data.frame(cbind(x,d2$n,d1$n))
  v1=1
  for (row in 1:nrow(d3))
  {
    v1<-cbind(v1,as.integer(d3[row,2]),as.integer(d3[row,3])  )
  }
  mymatl = matrix(v1[,-1],nrow(d3),2,byrow=T)
  dimnames(mymatl)<-list("Treatment"=as.vector(d3$x),
                        "Response"=c("Yes","No"))
  #Odds Ratio Intervals
  o =oddsratio.wald(mymatl)
  print(o)
  #prop.table(mymatl,margin=1)
  #prop.test(mymatl,correct=TRUE)
}

#Prop / Odds ration for Education
bd = bankdata[,c('education','y')]
dmx = bd %>% group_by(education,y) %>% summarize(n = n())
getOddsRation(dmx,education)

#Prop / Odds ration for loan
bd = bankdata[,c('y','loan')]
dmx = bd %>% group_by(loan, y) %>% summarize(n = n())
getOddsRation(dmx,loan)

#Prop / Odds ration for housing
bd = bankdata[,c('y','housing')]
dmx = bd %>% group_by(housing,y) %>% summarize(n = n())
getOddsRation(dmx,housing)

#Prop / Odds ration for poutcome
bd = bankdata[,c('poutcome','y')]
dmx = bd %>% group_by(poutcome,y) %>% summarize(n = n())
getOddsRation(dmx,poutcome)

#Odds ration for Job
bd = bankdata[,c('y','job')]
dmx = bd %>% group_by(job,y) %>% summarize(n = n())
getOddsRation(dmx,job)

#Odds ration for emp.var.rate
distinct(bankdata, bankdata$emp.var.rate)
dmx = bankdata %>% group_by(emp.var.rate) %>% count(y)
getOddsRation(dmx,emp.var.rate)
```

LDA Code

```
##We will analyze two LDA using over/under sample
#lda has better accuracy vs qda so we have removed the qda model
# variables in th elda model are a set of continous variables using the EDA analysis
#PCA variables were not included in this analysis
mylda<-lda(y ~ lduration+poutcome+pdays+campaign+emp.var.rate+cons.price.idx+euribor3m+
         cons.conf.idx+nr.employed, data=train)
pred.values<-data.frame(predict(mylda,newdata=test)$posterior[,1])
pred = as.factor(ifelse(pred.values >0.1, "yes", "no"))
```

```
pred  <-relevel(pred, ref = "yes")
Truth<-test$y
#table(pred)
x<-as.matrix(table(pred,Truth)) # Creating a confusion matrix
CM = confusionMatrix(x)
CM

## ROC curve for LDA train set
pred.lda <- predict(mylda, newdata = train)
lda.preds <- pred.lda$posterior
lda.preds <- as.data.frame(lda.preds)
lda.pred <- prediction(lda.preds[,2],train$y)
lda.roc.perf = performance(lda.pred, measure = "fnr", x.measure = "tnr")
lda.auc.train <- performance(lda.pred, measure = "auc")
lda.auc.train <- lda.auc.train@y.values
plot(lda.roc.perf)
abline(a=0, b= 1)
text(x = .40, y = .6,paste("AUC = ", 1 -round(lda.auc.train[[1]],3), sep = ""))
plot( lda.roc.perf, colorize = TRUE)

# ROC curve for LDA using test set
pred.ldat <- predict(mylda, newdata = test)
predst <- pred.ldat$posterior
predst <- as.data.frame(predst)
predt <- prediction(predst[,2],test$y)
roc.perft = performance(predt, measure = "fnr", x.measure = "tnr")
auc.traint <- performance(predt, measure = "auc")
auc.traint <- auc.traint@y.values
#plot(roc.perft)
#abline(a=0, b= 1)
plot(roc.perft,col="orange")
abline(a=0, b= 1)
text(x = .40, y = .6,paste("AUC = ", 1 -round(auc.traint[[1]],3), sep = ""))
#text(x = .40, y = .6,paste("AUC = ", round(auc.traint[[1]],3), sep = ""))
plot( roc.perft, colorize = TRUE)
#######End LDA#################
```

Random Forest Code

```
#######Random Forest
train.rf<-randomForest(y~.,data=train,mtry=11,ntree=500,importance=T)
#rf.fit.pred<-data.frame(predict(train.rf,newdata=train,type="prob"))
#fit.pred = fit.pred %>% mutate(pred = ifelse(yes>0.3, "yes", "no"))
########ROC On test set
rf.fit.pred<-data.frame(predict(train.rf,newdata=test,type="prob"))
#fit.pred = fit.pred %>% mutate(pred = ifelse(yes>0.3, "yes", "no"))
rf.pred <- prediction(rf.fit.pred$yes, test$y)
rf.roc.perf = performance(rf.pred, measure = "tpr", x.measure = "fpr")
rf.auc.train <- performance(rf.pred, measure = "auc")
rf.auc.train <- rf.auc.train@y.values
plot(rf.roc.perf)
abline(a=0, b= 1)
text(x = .40, y = .6,paste("AUC = ", round(rf.auc.train[[1]],3), sep = ""))
plot( rf.roc.perf, colorize = TRUE)
###########Predict RM
#prediction on test
rf.fit.pred<-data.frame(predict(train.rf,newdata=test,type="prob"))
rf.fit.pred = rf.fit.pred %>% mutate(pred = ifelse(yes >0.2, "yes", "no"))
table(rf.fit.pred$pred)
p = as.factor(rf.fit.pred$pred)
p  <-relevel(p, ref = "yes")
Truth<-test$y
x = as.matrix(table(p,Truth))
CM = confusionMatrix(x)
```

CM

Code LR I & II Feature Selection

```
####Simple LR using Lasso feature selection and simple model I
lasso1.dat.train.y<-train[,c("y")]
lasso1.dat.train.x <- model.matrix(y~.,train)
lasso1.cvfit <- cv.glmnet(lasso1.dat.train.x, lasso1.dat.train.y, family = "binomial", type.measure = "class", nlambda =
1000)
plot(lasso1.cvfit)
#Optimal penalty
lasso1.cvfit$lambda.min

coef(lasso1.cvfit)
#based on coefficients and EDA we keep only required varibles(remove dayswk)
lasso1.dat.train.x <- model.matrix(y~job+education+default+month+duration+pdays+poutcome+emp.var.rate
                      +cons.conf.idx+nr.employed,train)
lasso.final.cvfit <- cv.glmnet(lasso1.dat.train.x, lasso1.dat.train.y, family = "binomial", type.measure = "class", nlambda =
1000)
plot(lasso.final.cvfit)
#Optimal penalty
lasso.final.cvfit$lambda.min
coef(lasso.final.cvfit)

#Prediciton and ROC curve
#lasso.finalmodel<-glmnet(lasso1.dat.train.x, lasso1.dat.train.y, family = "binomial",lambda=cvfit$lambda.min)

#In addition to LASSO, if we are concerned that the biased estiamtes are affecting our model, we can go back and refit
using regular
#regression removing the variables that have no importance.
lassomodel2<-glm(y~job+education+default+month+duration+pdays+poutcome+emp.var.rate
              +cons.conf.idx+nr.employed,data=train,family=binomial)
(vif(lassomodel2)[,3])^2
summary(lassomodel2)
plot(lassomodel2)

lasso.finalmodel<-glm(y~job+education+default+month+lduration+pdays+poutcome+emp.var.rate
                  +cons.conf.idx+nr.employed,data=train,family=binomial)

e = exp(cbind("Odds ratio" = coef(lasso.finalmodel), confint.default(lasso.finalmodel, level = 0.95)))
write.csv(e, "output.csv", row.names = TRUE)

####Lack of fit test
hoslem.test(lasso.finalmodel$y, fitted(lasso.finalmodel), g=10)

e = summary(lasso.finalmodel)
write.csv(e$coefficients, "output.csv", row.names = TRUE)
coef(lasso.finalmodel)
plot(lasso.finalmodel)
# with lduration

#Get training set predictions...We know they are biased but lets create ROC's.
#These are predicted probabilities from logistic model  exp(b)/(1+exp(b))
lasso.fit.pred <- data.frame(predict(lasso.finalmodel, newdx = lasso1.dat.train.x, type = "response"))

#Create ROC curves (Remember if you have a test data set, you can use that to compare models)
lasso.pred <- prediction(lasso.fit.pred[,1], train$y)
lasso.roc.perf = performance(lasso.pred, measure = "fnr", x.measure = "tnr")
lasso.auc.train <- performance(lasso.pred, measure = "auc")
lasso.auc.train <- lasso.auc.train@y.values
#Plot ROC for train (we will remove it)
plot(lasso.roc.perf,main="LASSO")
abline(a=0, b= 1) #Ref line indicating poor performance
text(x = .40, y = .6,paste("AUC = ", 1 -round(lasso.auc.train[[1]],3), sep = ""))
```

```
## Do same for test
lasso.fit.pred <- data.frame(predict(lasso.finalmodel, newdata  = test, type = "response"))

lasso.pred <- prediction(lasso.fit.pred[,1], test$y)
lasso.roc.perf = performance(lasso.pred, measure = "fnr", x.measure = "tnr")
lasso.auc.train <- performance(lasso.pred, measure = "auc")
lasso.auc.train <- lasso.auc.train@y.values

plot(lasso.roc.perf,main="LASSO")
abline(a=0, b= 1) #Ref line indicating poor performance
text(x = .40, y = .6,paste("AUC = ", 1 -round(lasso.auc.train[[1]],3), sep = ""))
plot( lasso.roc.perf, colorize = TRUE)

#prediction on test
#lasso.fit.pred
lasso.fit.pred = lasso.fit.pred %>% mutate(pred = ifelse(lasso.fit.pred <0.8, "yes", "no"))
table(lasso.fit.pred$pred)
p = as.factor(lasso.fit.pred$pred)
p  <-relevel(p, ref = "yes")
Truth<-test$y
x = as.matrix(table(p,Truth))
CM = confusionMatrix(x)
CM

#########Complex LR model
lr.complex.finalmodel<- glm(y~ contact + month*day_of_week + age*duration + campaign + pdays +
                  emp.var.rate*cons.price.idx +  cons.conf.idx +  euribor3m + log(nr.employed), data=train, family=binomial)
lr.complex.fit.pred <- data.frame(predict(lr.complex.finalmodel, newdata  = test, type = "response"))
lr.complex.pred <- prediction(lr.complex.fit.pred[,1], test$y)
lr.complex.roc.perf = performance(lr.complex.pred, measure = "fnr", x.measure = "tnr")
lr.complex.auc.train <- performance(lr.complex.pred, measure = "auc")
lr.complex.auc.train <- lr.complex.auc.train@y.values
plot(lr.complex.roc.perf,main="LASSO")
abline(a=0, b= 1) #Ref line indicating poor performance
text(x = .40, y = .6,paste("AUC = ", 1 -round(lr.complex.auc.train[[1]],3), sep = ""))
plot( lr.complex.roc.perf, colorize = TRUE)
#prediction on test
lr.complex.fit.pred = lr.complex.fit.pred %>% mutate(pred = ifelse(lr.complex.fit.pred <0.7, "yes", "no"))
table(lr.complex.fit.pred$pred)
p = as.factor(lr.complex.fit.pred$pred)
p  <-relevel(p, ref = "yes")
Truth<-test$y
x = as.matrix(table(p,Truth))
CM = confusionMatrix(x)
CM

###Comparative ROC curve
plot( lda.roc.perf)
plot(lasso.roc.perf,col="orange", add = TRUE)
plot(rf.roc.perf,col="red", add = TRUE)
plot(lr.complex.roc.perf,col="blue", add = TRUE)
rfauc = round(rf.auc.train[[1]],3)
rflabel = paste("RF:" , rfauc)
ldaauc = 1 - round(lda.auc.train[[1]],3)
ldalabel = paste("LDA:" , ldaauc)
lassoauc = 1 -round(lasso.auc.train[[1]],3)
lassolabel = paste("LR I:" , lassoauc)
lassoauc2 = 1 -round(lr.complex.auc.train[[1]],3)
lassolabel2 = paste("LR II:" , lassoauc2)

legend("bottomright",legend=c(ldalabel,lassolabel, lassolabel2, rflabel),col=c("black","orange","blue","red"),lty=1,lwd=1)
abline(a=0, b= 1)
```