Credit Card Default

December 2005

Grace and Mahima

Introduction

We are trying to predict whether a customer won't be able to pay their credit card bill the following month.

Our data consists of the following information on 30,000 customers from April to September 2005 :

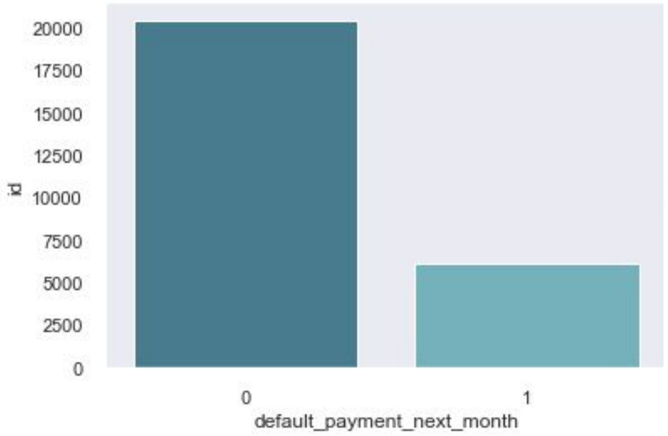
1. Demographics - age, sex, marital status, education
2. Amount of credit given or the limit balance
3. Repayment status from April to September 2005
4. Bill statement as of April to September 2005
5. Amount of previous payment as in each month
6. Default next month (1 = yes, 0 = no)

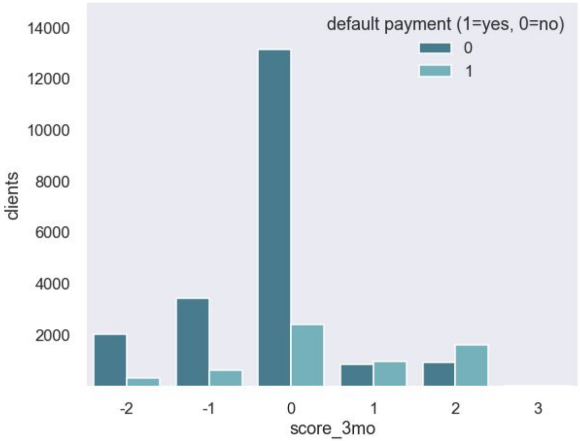
Source: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

citibank

EDA

Non-defaulters: 78% of the dataset

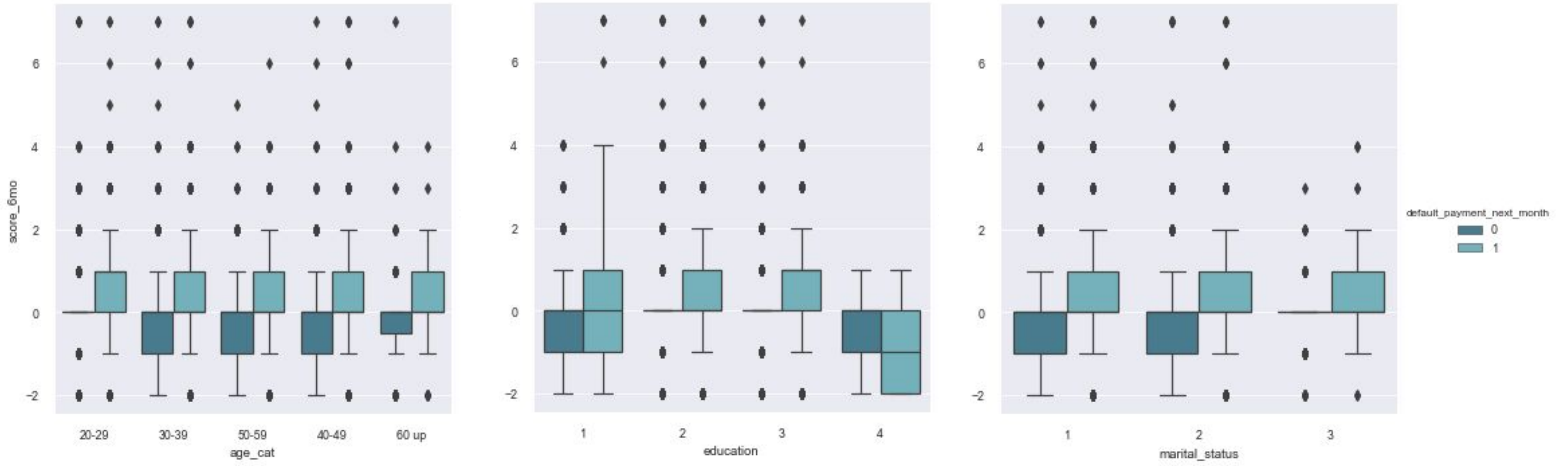




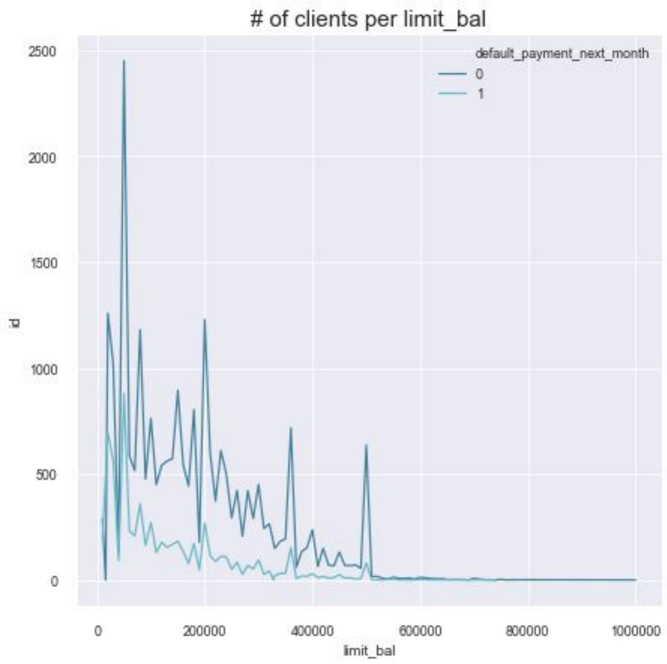
citibank

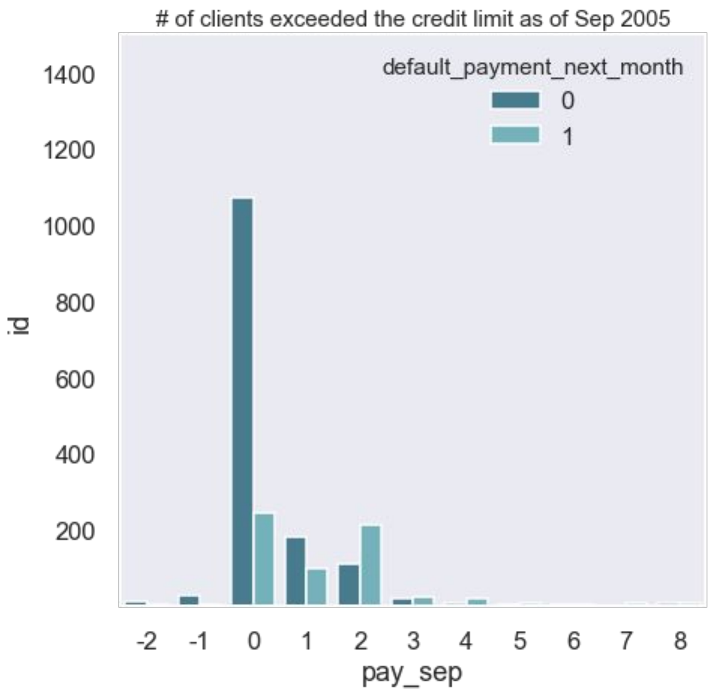
EDA

Defaulters by demographics (past 6 months)



EDA





citibank

EDA

"Best" predictors based on correlation matrix:

- pay_sep
- score_3mo: weighted average of last July - September pay columns
- score_6mo: weighted average of April - September pay columns
- pay_aug
- pay_jul
- pay_jun
- pay_may
- pay_apr
- limit_bal (negative correlation)

citibank

Logistic Regression

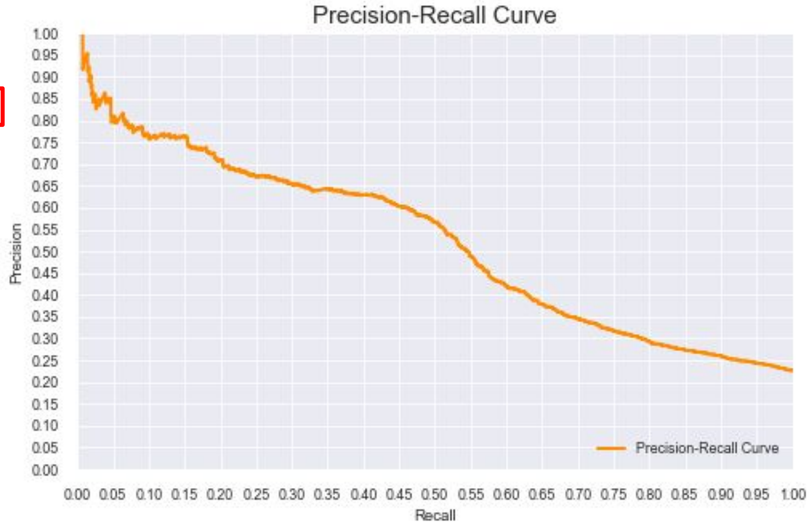
- Feature engineering:
  - weighted average of ‘pay’ columns for the past 3 and 6 months
  - bill amount as of September as percentage of the total credit limit
- Feature selection:
  - SelectFromModel: 18 features selected out of 58
- Class imbalance:
  - SMOTE
- Highly-correlated features
  - PCA

Logistic Regression

Classification Report:

	precision	recall	f1-score	support
0	0.86	0.89	0.87	5067
1	0.56	0.51	0.53	1474
accuracy			0.80	6541
macro avg	0.71	0.70	0.70	6541
weighted avg	0.79	0.80	0.80	6541

Precision-Recall AUC: 0.52

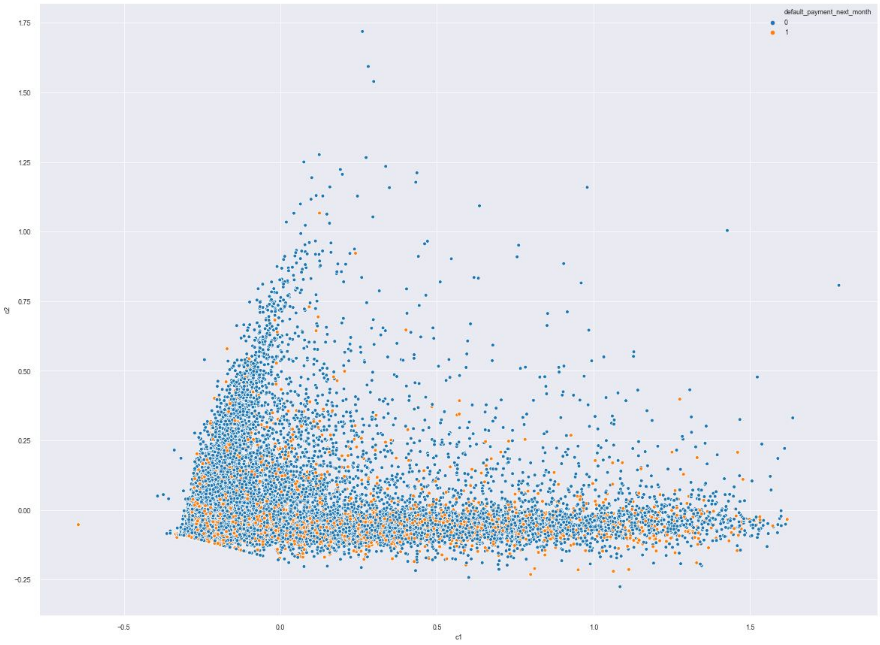


citibank

Logistic Regression

+ Principal Component Analysis

1	0.670292
2	0.102714
3	0.042986
4	0.039050
5	0.037658
6	0.035870
7	0.029988
8	0.027385
9	0.006803
10	0.003431
11	0.002136
12	0.001688



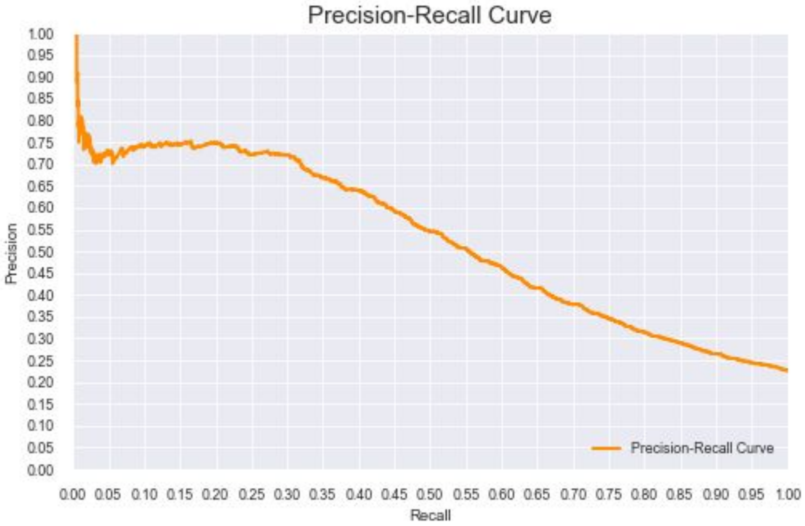
Logistic Regression

+ Principal Component Analysis

Classification Report:

	precision	recall	f1-score	support
0	0.87	0.83	0.85	5067
1	0.49	0.57	0.52	1474
accuracy			0.77	6541
macro avg	0.68	0.70	0.68	6541
weighted avg	0.78	0.77	0.77	6541

Precision-Recall AUC: 0.53



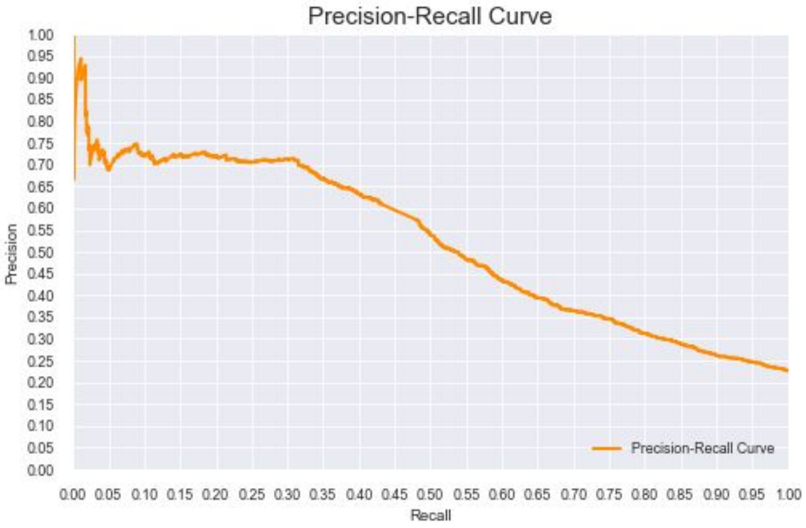
citibank

Support Vector Machines

Classification Report:

	precision	recall	f1-score	support
0	0.87	0.78	0.82	5067
1	0.44	0.59	0.51	1474
accuracy			0.74	6541
macro avg	0.65	0.69	0.66	6541
weighted avg	0.77	0.74	0.75	6541

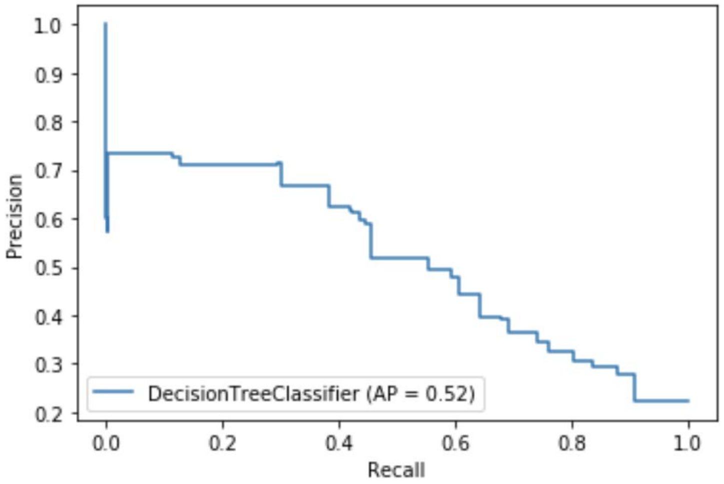
Precision-Recall AUC: 0.52



Random Forest

- Feature engineering:
    - weighted average of ‘pay’ columns for the past 3 and 6 months
    - bill amount as of September as percentage of the total credit limit
- Feature selection:
    - all features
- Class imbalance:
    - SMOTE

Random Forest



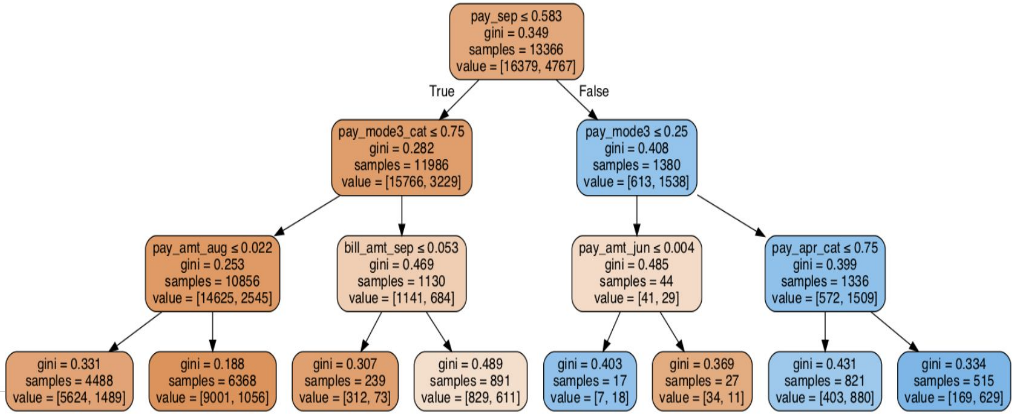

```
#calculating the auc
precision_recall_auc = auc(recall, precision)
precision_recall_auc
```

0.5940512167372005

	precision	recall	f1-score	support
0	0.84	0.95	0.89	4106
1	0.67	0.38	0.49	1181
accuracy			0.82	5287
macro avg	0.75	0.66	0.69	5287
weighted avg	0.80	0.82	0.80	5287

Random Forest

	rank_test_score	params	mean_test_score	std_test_score
0	7	{'max_depth': 3, 'min_samples_leaf': 5}	0.436625	0.020016
1	7	{'max_depth': 3, 'min_samples_leaf': 10}	0.436625	0.020016
2	7	{'max_depth': 3, 'min_samples_leaf': 15}	0.436625	0.020016
3	1	{'max_depth': 5, 'min_samples_leaf': 5}	0.468013	0.020908
4	2	{'max_depth': 5, 'min_samples_leaf': 10}	0.465494	0.020496
5	3	{'max_depth': 5, 'min_samples_leaf': 15}	0.464945	0.020169
6	4	{'max_depth': 7, 'min_samples_leaf': 5}	0.461309	0.025128
7	6	{'max_depth': 7, 'min_samples_leaf': 10}	0.458842	0.018647
8	5	{'max_depth': 7, 'min_samples_leaf': 15}	0.460636	0.017897



Conclusions

- Recall is more important than Precision
- None of the models are good in predicting defaulters
- Our best model: Logistic (recall = 0.59, f1= 0.51)
- We need to revisit and get more samples and features:
  - Income, estimate monthly expense
  - Related accounts (savings, loans, other credit cards)
  - Up to 12 or 24 month history

Thank you

Appendix

True positive (TP): the model predicts default and the client is indeed defaulting

False positive (FP): the model predicts default but the client is not defaulting

True negative (TN): the model predicts not defaulting and the client is not defaulting

False negative (FN): the model predicts not defaulting but the client is in fact defaulting

Precision = TP/TP+FP

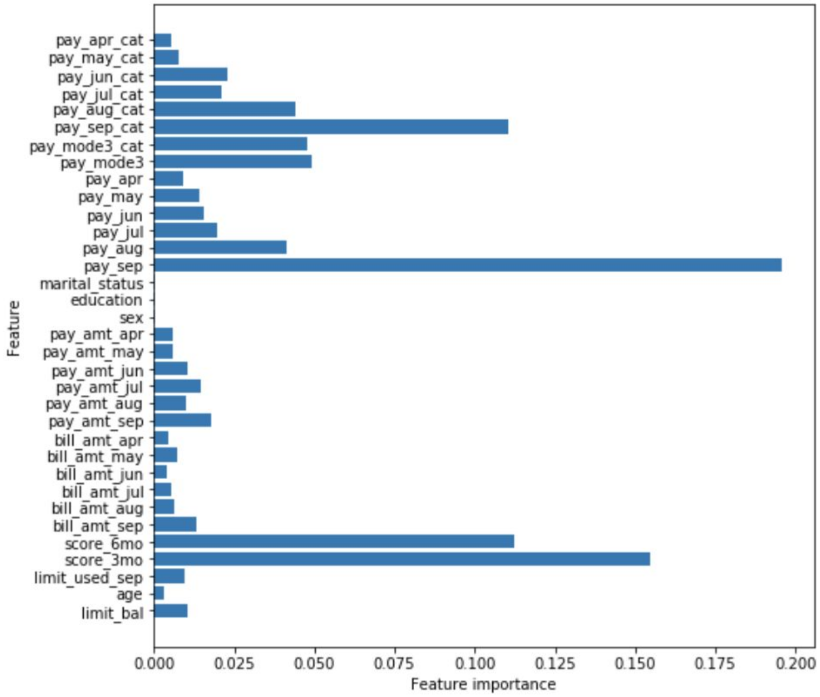
Recall / Sensitivity = TP/TP+FN = TPR (probability of detection)

Why recall matters to us is because of the False Negatives component i.e. the chance of knowing if someone has defaulted if they have actually defaulted. Not knowing this number will be a heavy cost to the bank and hence this is our key metric.

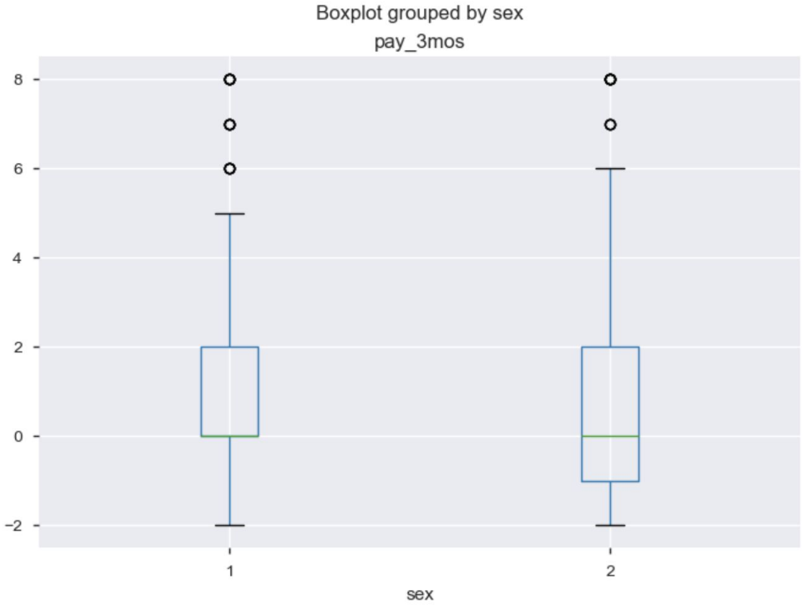
name	type	baseline	f1_weighted	f1_class_1	precision_recall_auc	recall
logreg model A	logistic, 18 features out of 58, default hyper...	0.66	0.75	0.51	0.52	0.6
logreg model B	logistic, manually selected features, default ...	0.66	0.8	0.53	0.52	0.51
logreg gridsearchcv #1	logistic, {'C': 1, 'penalty': 'l1'}	0.66	0.75	0.51	0.52	0.59
logreg gridsearchcv #3	logistic, {'C': 10, 'penalty': 'l2'}	0.66	0.75	0.51	0.52	0.59
pca logreg model A	pca logreg, default hyperparameters	0.66	0.77	0.52	0.53	0.57
pca logreg gridsearchcv #1	pca logreg {'C': 0.25, 'penalty': 'l1'}	0.66	0.77	0.52	0.53	0.57
pca logreg gridsearchcv #2	pca logreg {'C': 0.5, 'penalty': 'l2'}	0.66	0.77	0.52	0.53	0.57
svc model A	svm, hyperparameters set to default	0.7	0.78	0.51	0.52	0.52
svc gridsearchcv #1	svm {'C': 1, 'gamma': 1}	0.7	0.8	0.52	0.51	0.48
svc gridsearchcv #2	svm {'C': 0.1, 'gamma': 1}	0.7	0.79	0.52	0.51	0.49

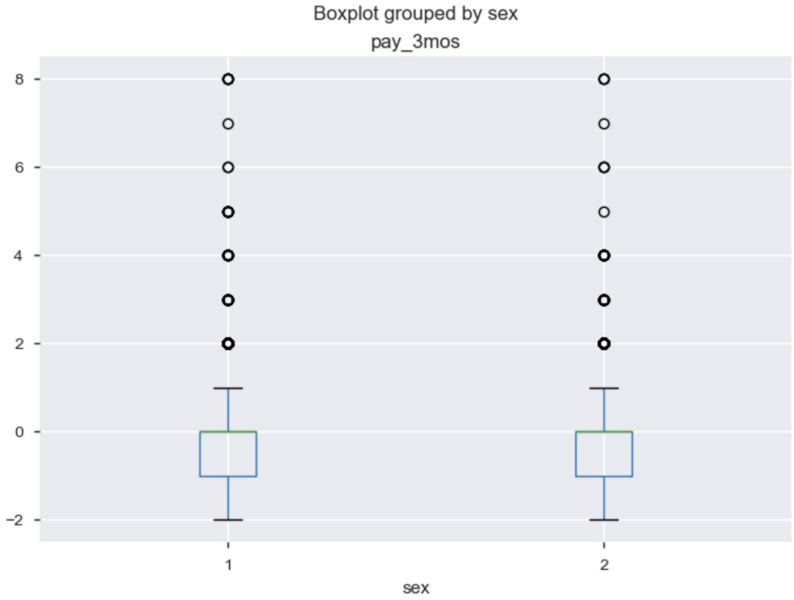
Note: We attempted to optimise SVM using several values of C and gamma and 3 different kernels but it was unsuccessful.

Random Forest

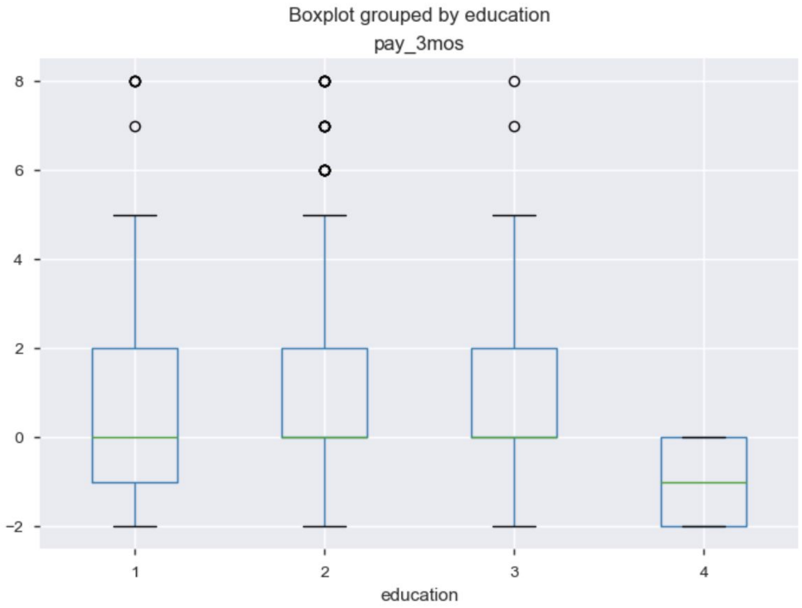


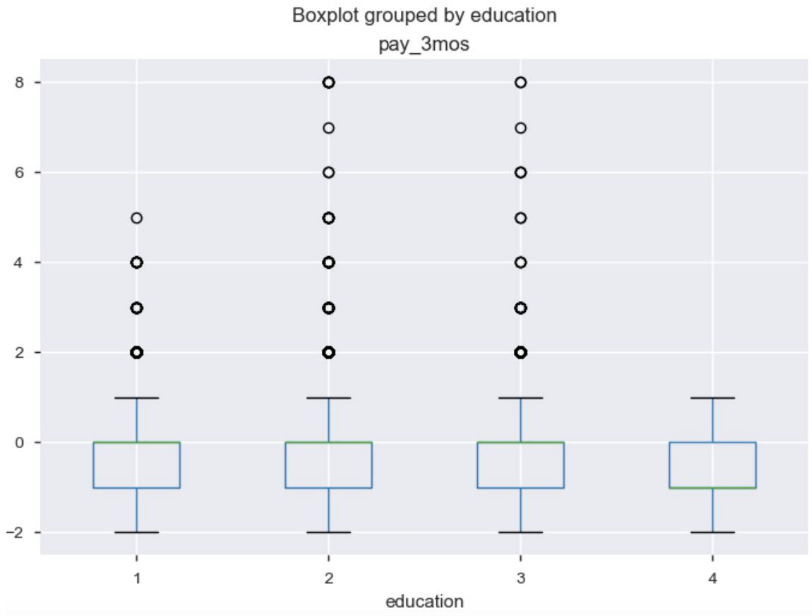
Class : default vs no default (by sex)



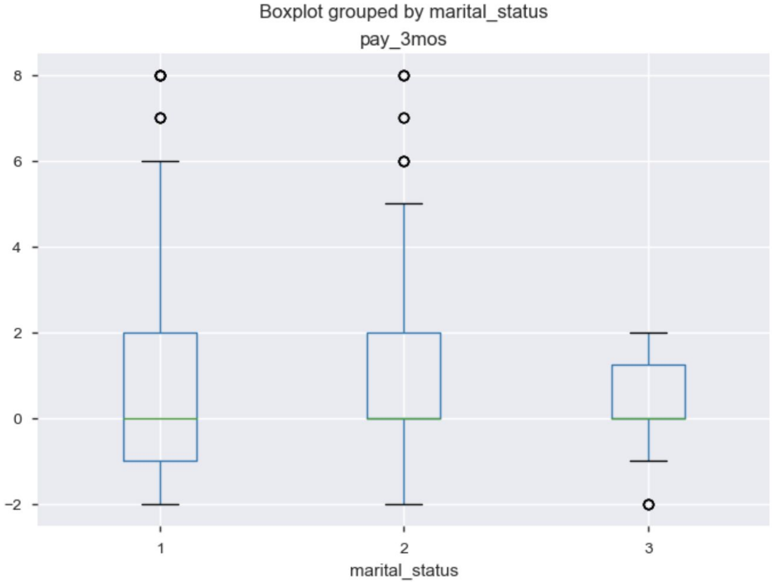


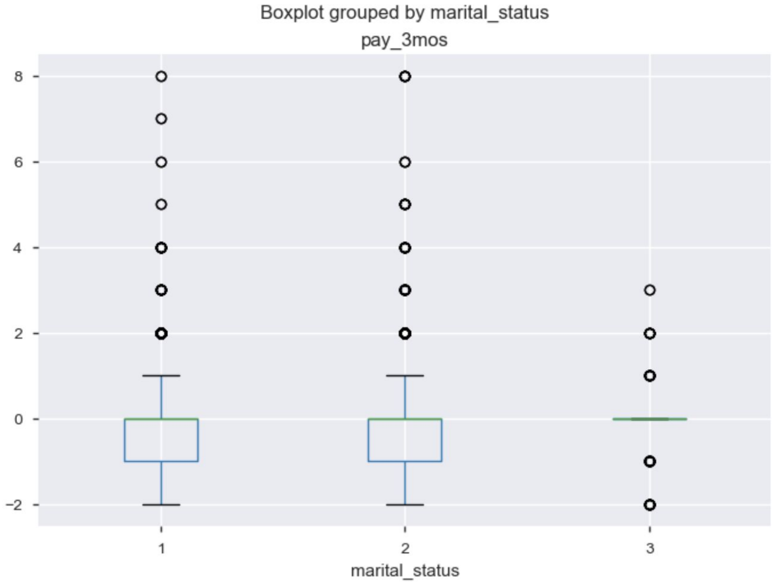
Class : default vs no default (by education)





Class : default vs no default (by marital status)





Class : default vs no default (by age)