PREDICTING LOAN DEFAULTS USING LOGISTIC REGRESSION

Sonja Tilly

November 17

Approach and method

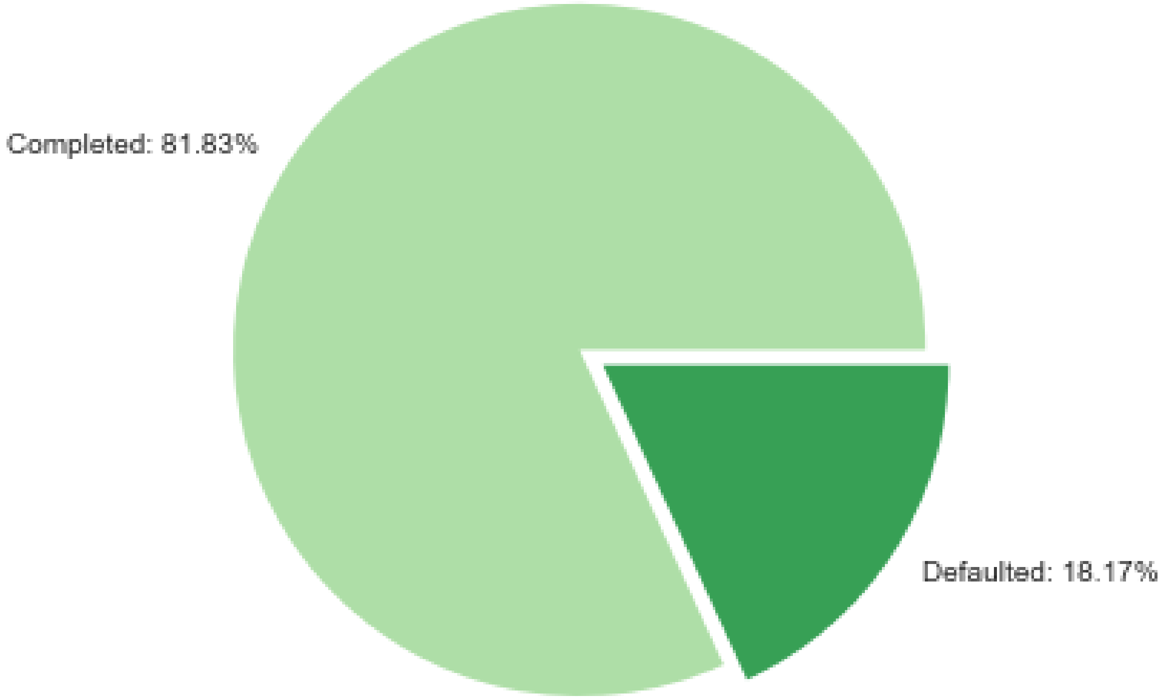
- Project
- Dataset
- Data exploration
- Feature engineering
- Classification
  - Building basic logistic regression model
  - Optimising model parameters
  - Feature selection
- Results
- Conclusion

Project

- Predict customers' creditworthiness
  (i.e. whether they will complete or default on a loan)
- Build and test a logistic regression model
- Use 70% of the data to build the model and then test the model using the remaining 30%

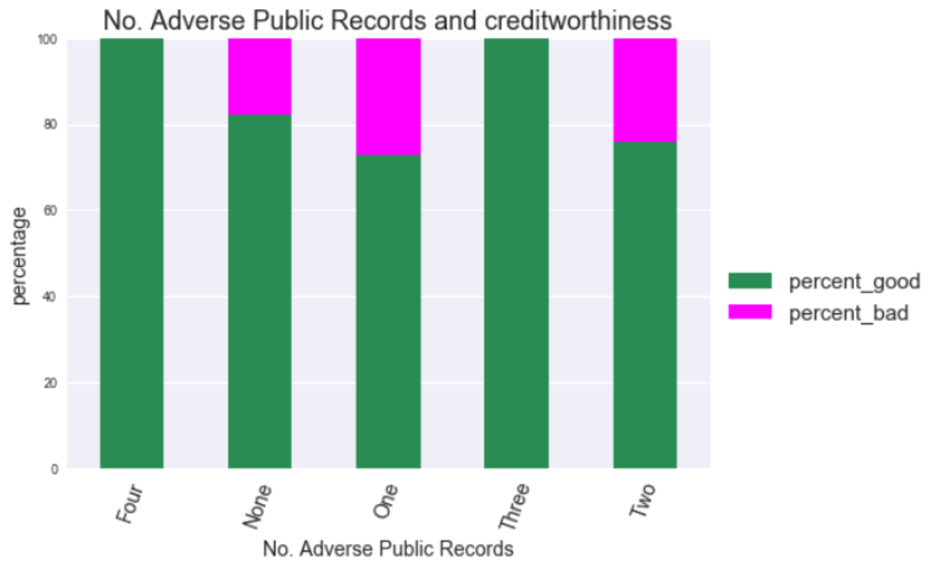
Lending Club dataset

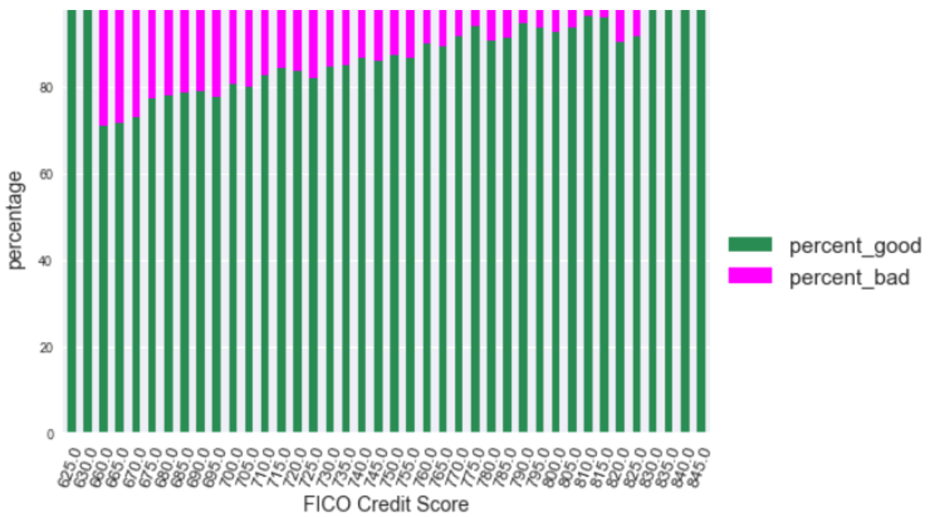
- Each row represents a customer.
- Each column shows the features for approved loan applications.
- The column ‘Class’ shows that each customer has defaulted or completed their loan.



- The dataset is unbalanced, with defaulted loans only accounting for 18.17%.
- The unbalanced nature of the data will have to be considered when making predictions.

Data exploration





- Relationships between predictor and target variable are not always linear.

- Observations of features ‘FICO Credit Score’ and ‘Address State’ can be grouped in order to increase predictive power.

Feature engineering

- Convert 'Class' feature (= target) into numerical values
- Summarise Address States and FICO Credit Score features
- Create dummy variables for categorical features
- Encode remaining columns containing object data types
- Drop redundant features
- Deal with nan values
- Remove outliers

Classification: Build model

```
# build logistic regression model
# address unbalanced classes with parameter 'class_weights'

from sklearn import linear_model
from sklearn.metrics import roc_auc_score

estimator = linear_model.LogisticRegression(random_state=42, class_weight={0:0.1817, 1:0.8183})

estimator.fit(X_train, y_train)

y_pred_regr = estimator.predict_proba(X_test)

y_pred_regr = pd.DataFrame(y_pred_regr, index=y_test.index)

# how does the basic model perform?

AUC = roc_auc_score(y_test, y_pred_regr[1])
print("The AUC score for a simple Logistic Regression model is {:.4f}.".format(AUC))
```

```
The AUC score for a simple Logistic Regression model is 0.7207.
```

Add class weights to account for unbalanced classes

Initial AUC score: 0.7207

Classification: Optimise model

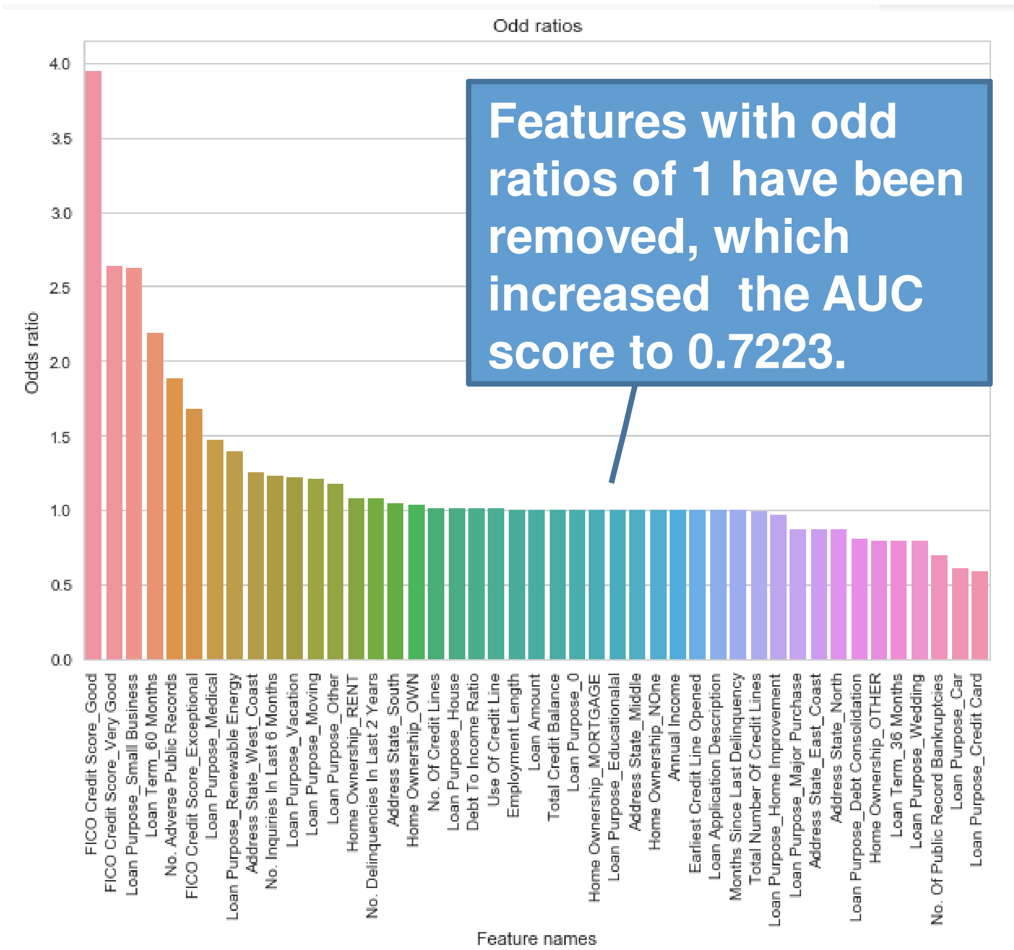
```
# tuning model parameters

from sklearn.model_selection import GridSearchCV


param_grid = {'C': [0.01, 1, 2, 3], 'penalty':['l1', 'l2']}

gsregr = GridSearchCV(estimator, param_grid, cv=9)

gsregr.fit(X_train, y_train)

y_pred_opt = gsregr.predict_proba(X_test)
y_pred_opt = pd.DataFrame(y_pred_opt, index = y_test.index)

# how does the optimised model perform?

AUC_opt = roc_auc_score(y_test, y_pred_opt[1])
print("The AUC score for an optimised Logistic Regression model is {:.4f}.".format(AUC_opt))
```

```
The AUC score for an optimised Logistic Regression model is 0.7216.
```

Use GridSearchCV to find best model parameters

Optimised AUC score: 0.7216

Results: Feature selection



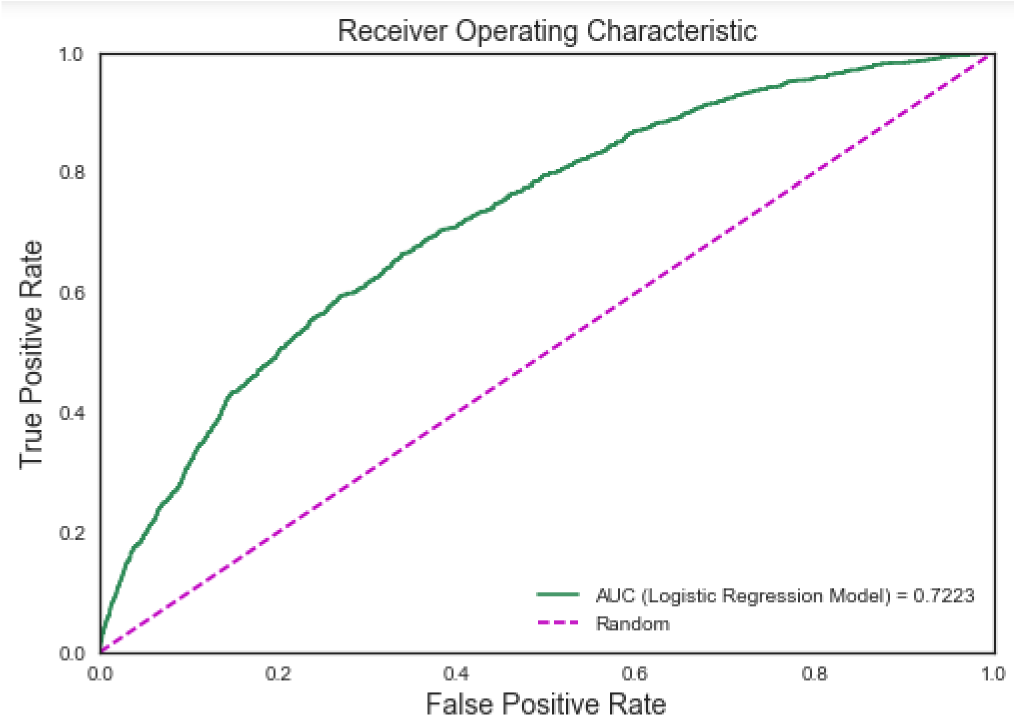
Highest odds for bad loans:

- FICO Credit Score Good
- FICO Credit Score Very Good
- Loan Purpose Small Business
- Loan Term 60 Months
- No. Adverse Public Records

Lowest odds for bad loans:

- Loan Purpose Credit Card
- Loan Purpose Car
- No. Of Public Record Bankruptcies
- Loan Purpose Wedding
- Loan Term 36 Months

Results: ROC Curve



- The optimised model delivered an AUC score of 0.7223.
- This can be interpreted as the probability that a randomly chosen positive example is deemed to have a higher probability of being positive than a randomly chosen negative example.
- Parameter tuning and feature selection improved the score.

Conclusion

- The model outperforms random guessing in predicting loan defaults.
- The model identifies features with high and low odds for bad loans.

Limitations

- Relationships between predictor variables and target are not necessarily linear.
- The amount of features and observations is not very large.

Recommendations

- Get more data.
- Build model using XG Boost, as this algorithm tends to perform well when variables exhibit non-linear relationships.