Assessing new loan applications using Machine Learning

Exploring machine learning methods for loan assessment

Jeremy Dang

Agenda

- Context
- Objectives
- Data
  - Understanding - Exploration
  - Preparation
  - Understanding - Correlation
- Modelling
  - Models Chosen
  - Considerations
  - Methodology
- Evaluation
  - Evaluation metrics
  - Model performance
- Conclusions
- Appendices

Context

- Loans provide a means of long term finance for several industries, in return, lenders' profit in the form of interest, payed on top of a loan

- However, loans are a huge risk to lenders', what if the borrower is unable to pay back their agreed loan?

- The business would incur a *loss* and would directly impact revenue streams.

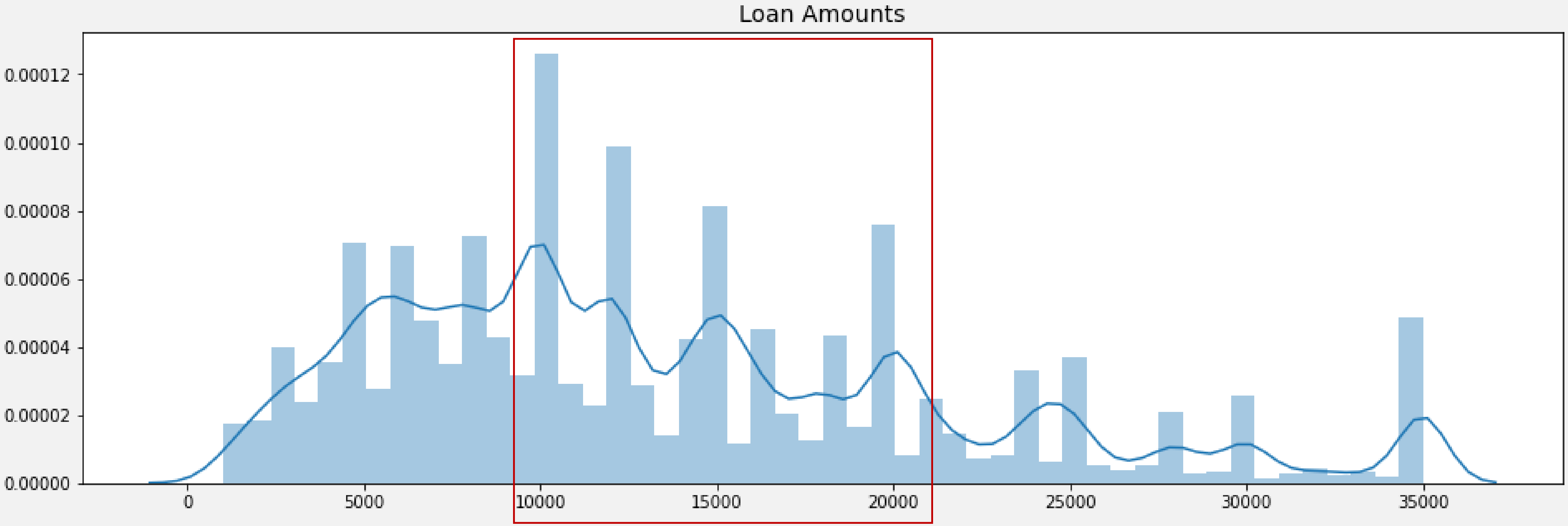
- Worse case scenario – A global recession
  - E.g – Financial crisis of 2007-2008

- Predicting if applicants default is an important task!

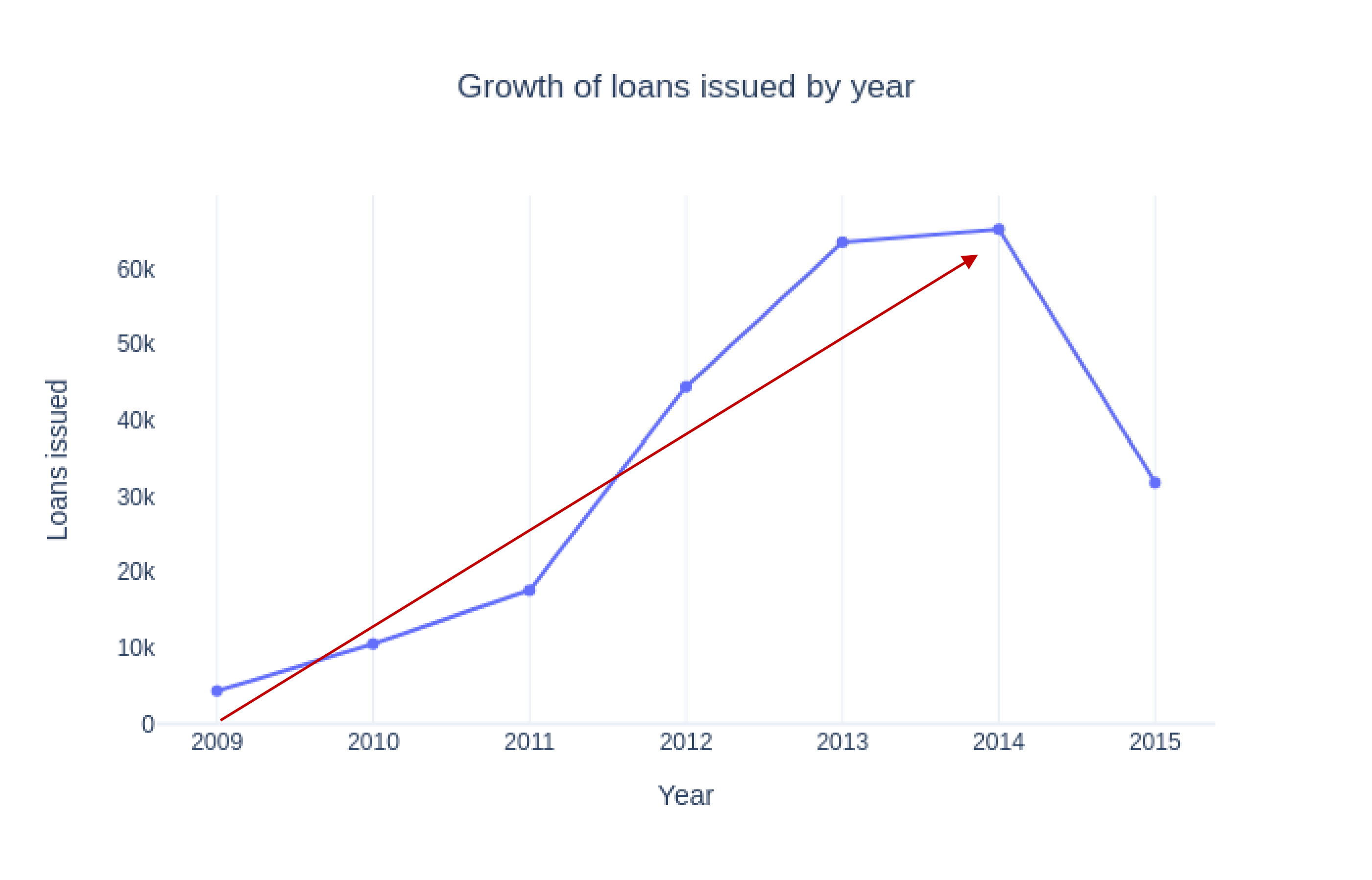
Objectives

- What do applicants who default look like?
- What are the factors correlated with default?
- How can we assist the analysts’ in minimising the cost of default?

Data Understanding – What are the amounts we're lending?



Data Understanding – Growth of loans over time



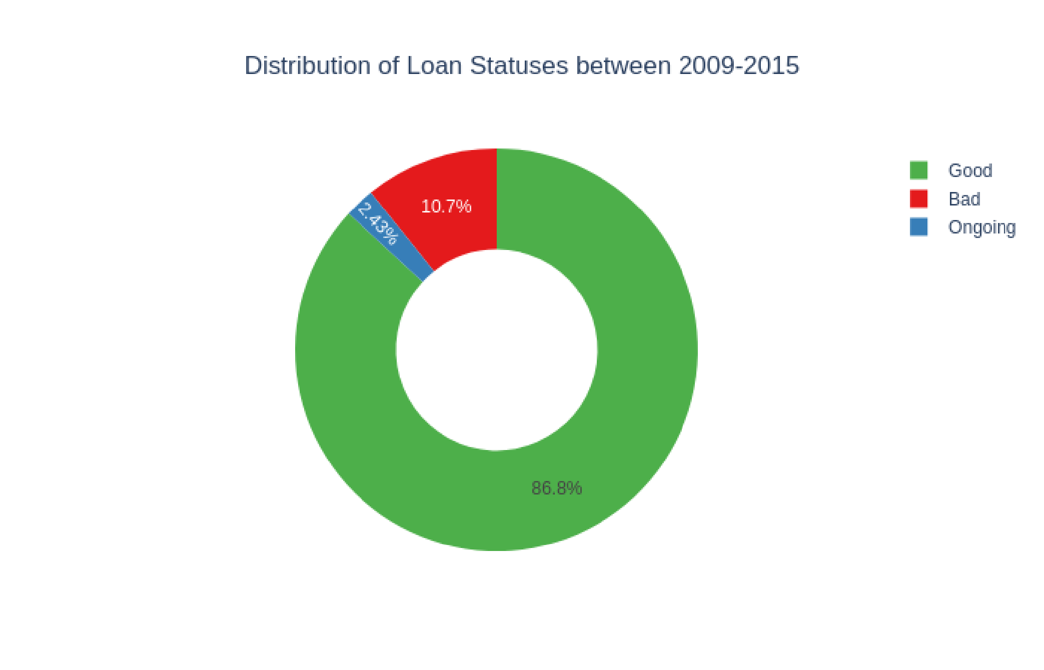
Data Understanding – What do we mean by default?

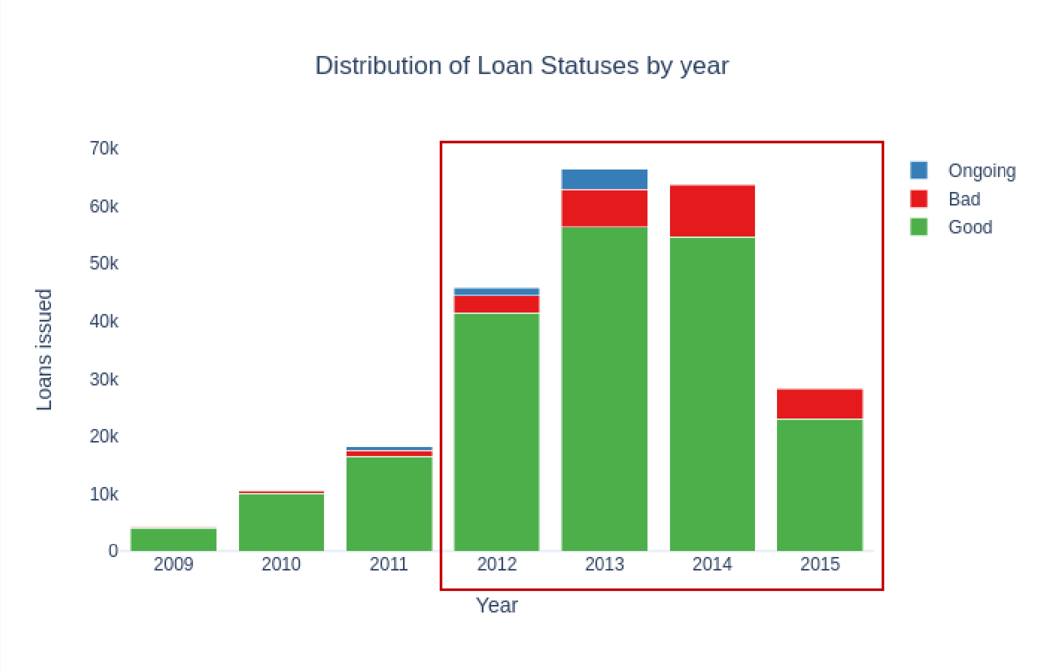
- For simplicity, three groups are defined on the *loan status* feature:
  - Good: An individual who has **fully paid** their loan back
  - Bad: An individual who has either **defaulted**, **charged-off** or **late by more than 90 days**
  - Ongoing: An individual who is still paying back their loan

- **Charged-off** represents a stage where we expect the loan to not be paid back → borrower has defaulted.

- **Late** is grouped according to the definition of *default from* IRB guidelines (defined by payments late by more than **than 90 or 180 days**)

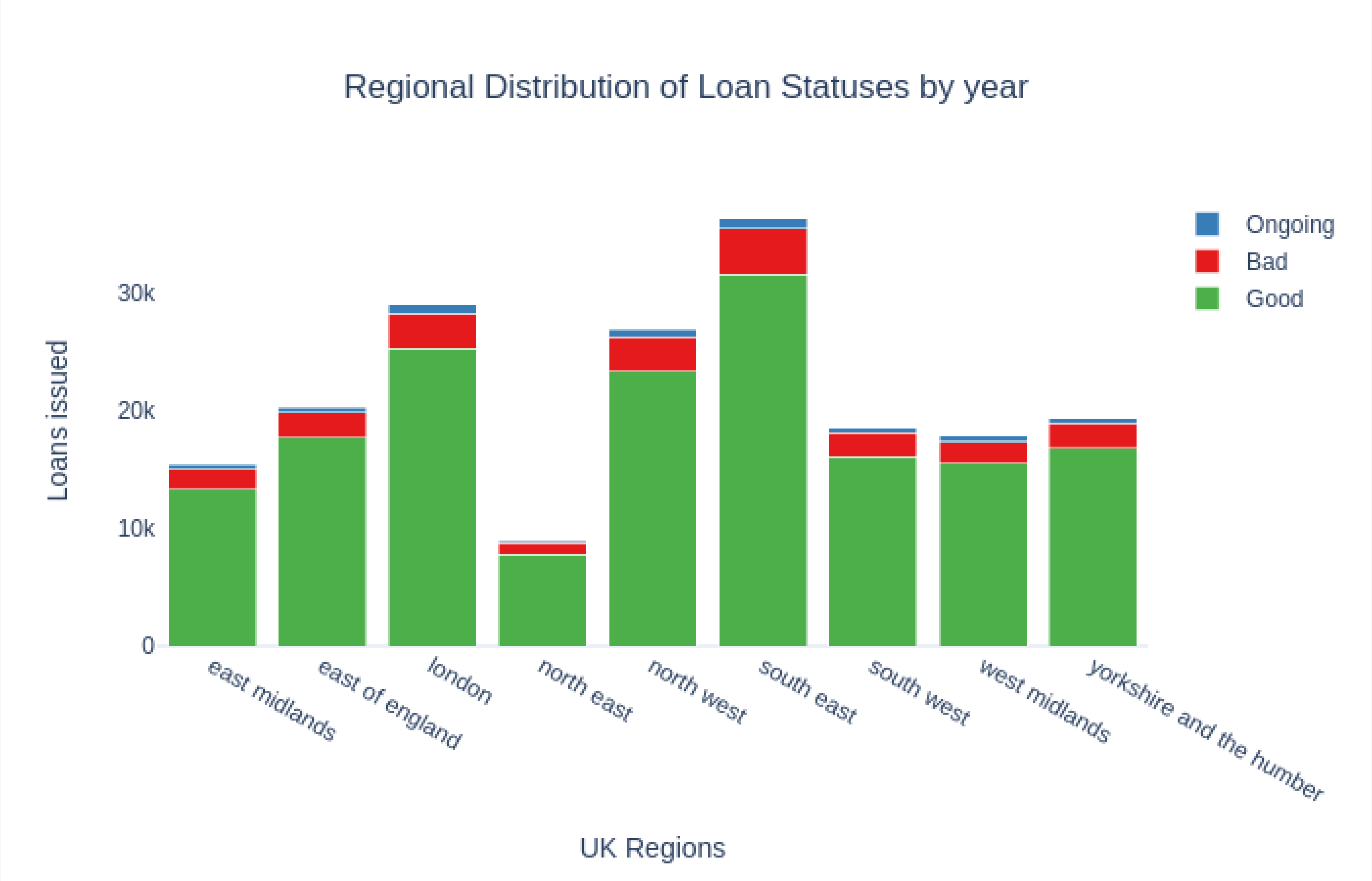
Data Understanding – What do applicants who default look like?





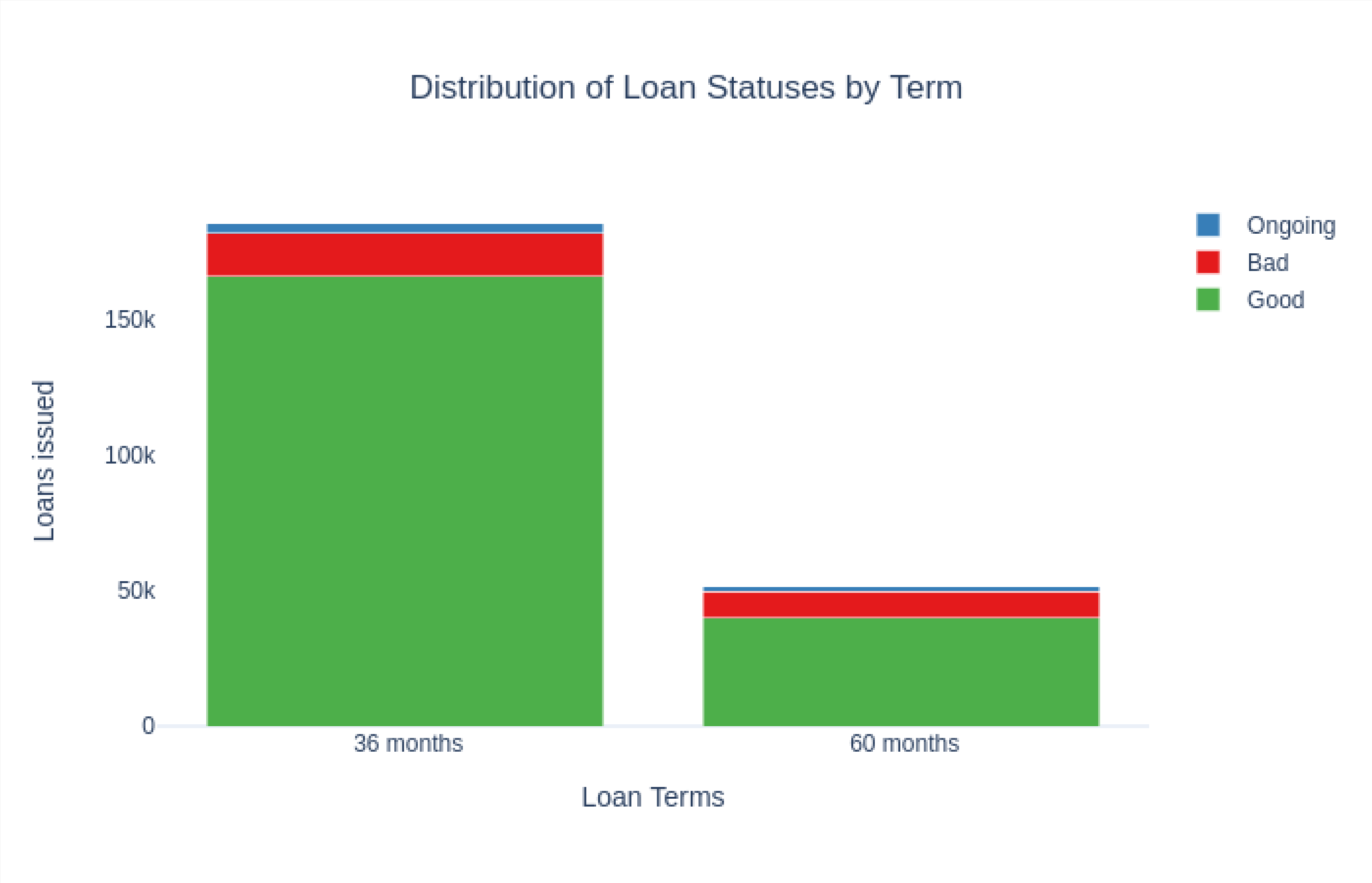
FINANCIAL CONDUCT AUTHORITY

Data Understanding – What do applicants who default look like?



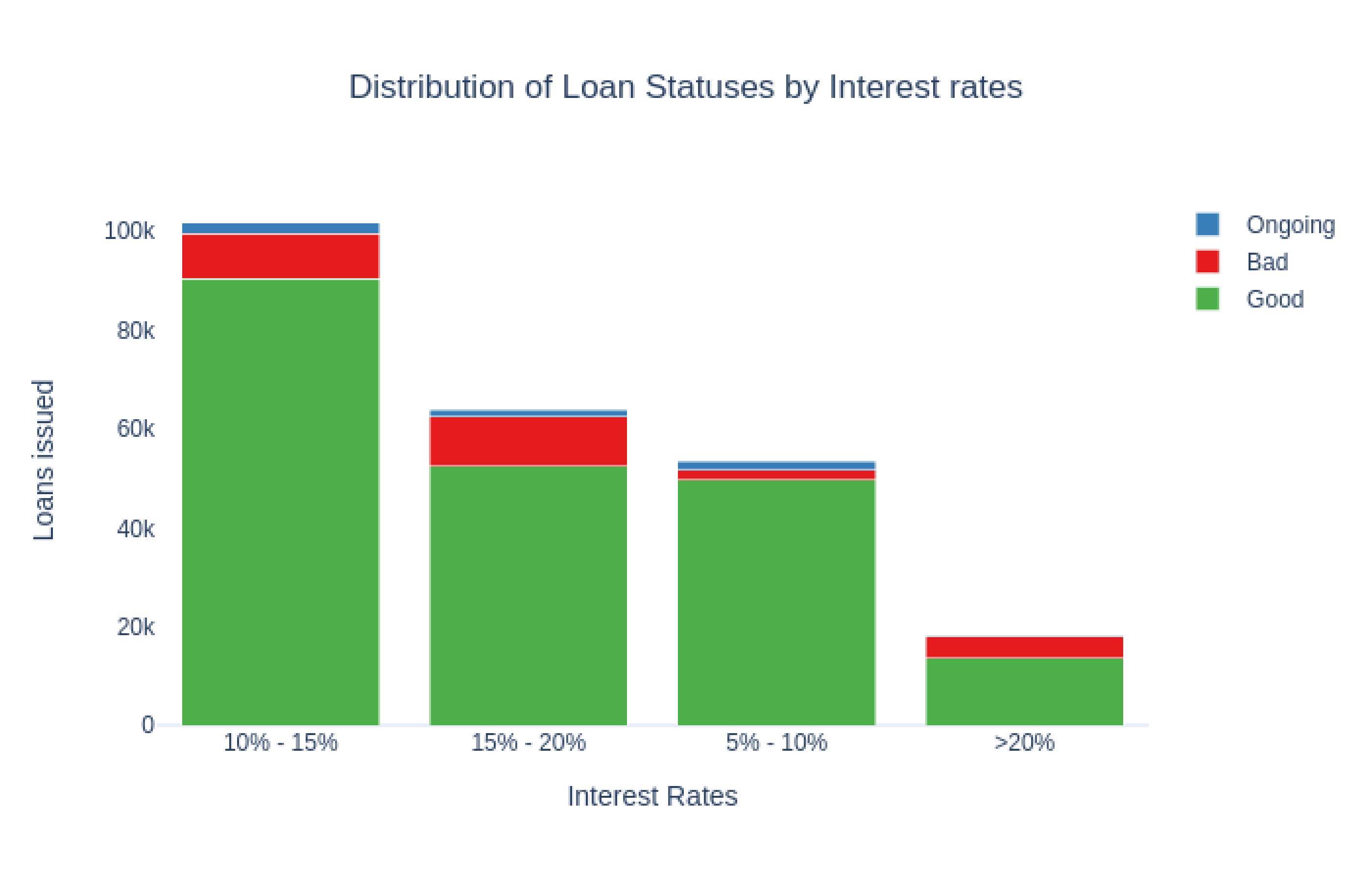
- The districts were mapped into regions where possible (82% coverage). See appendix for details.
- Regions of largest defaulted applicants: London, North West & South West

Data Understanding – What do applicants who default look like?



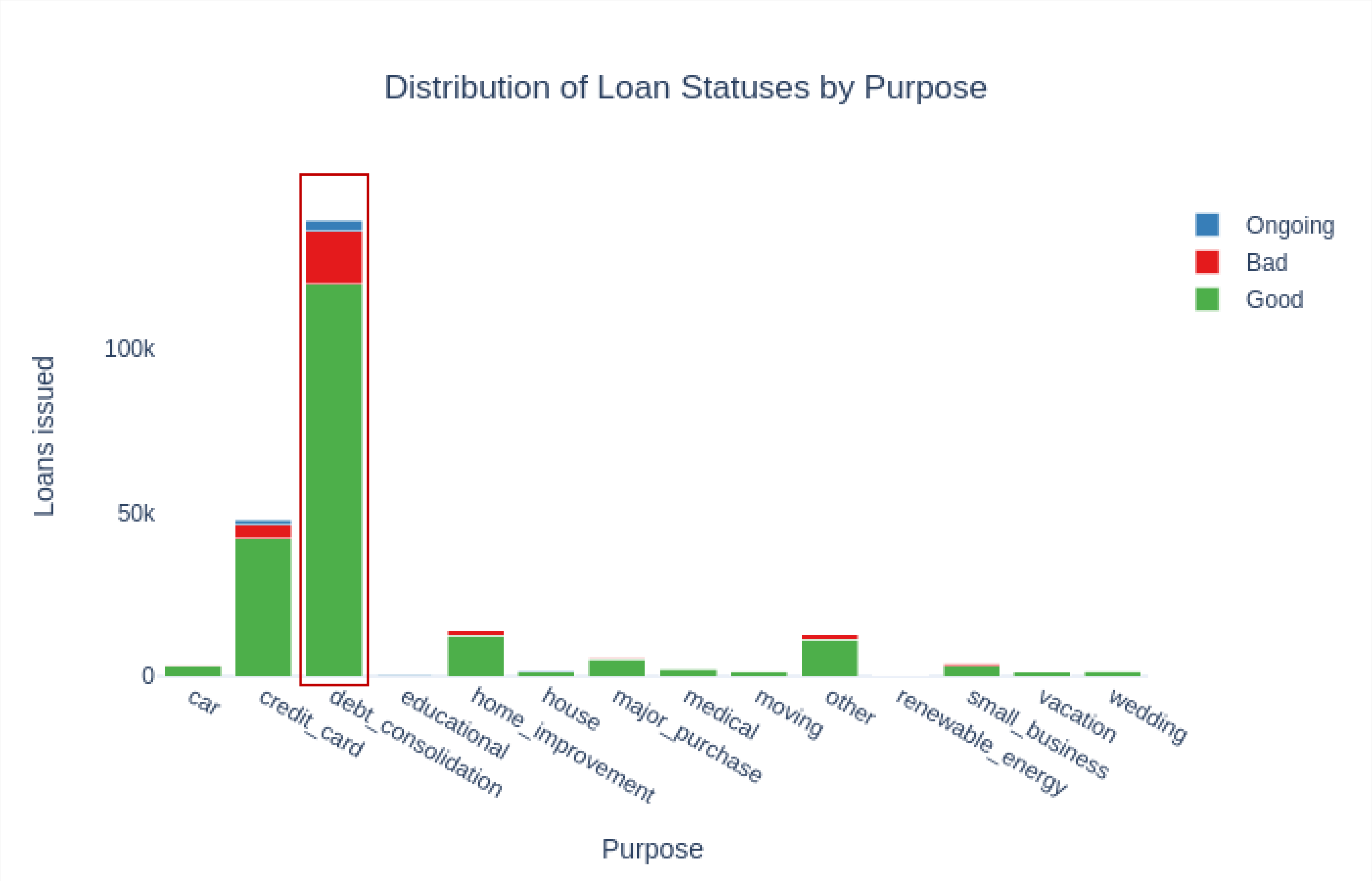
- 7% of loans defaulted were on 36-month terms
- 4% of loans defaulted were on 60-month terms

Data Understanding – What do applicants who default look like?



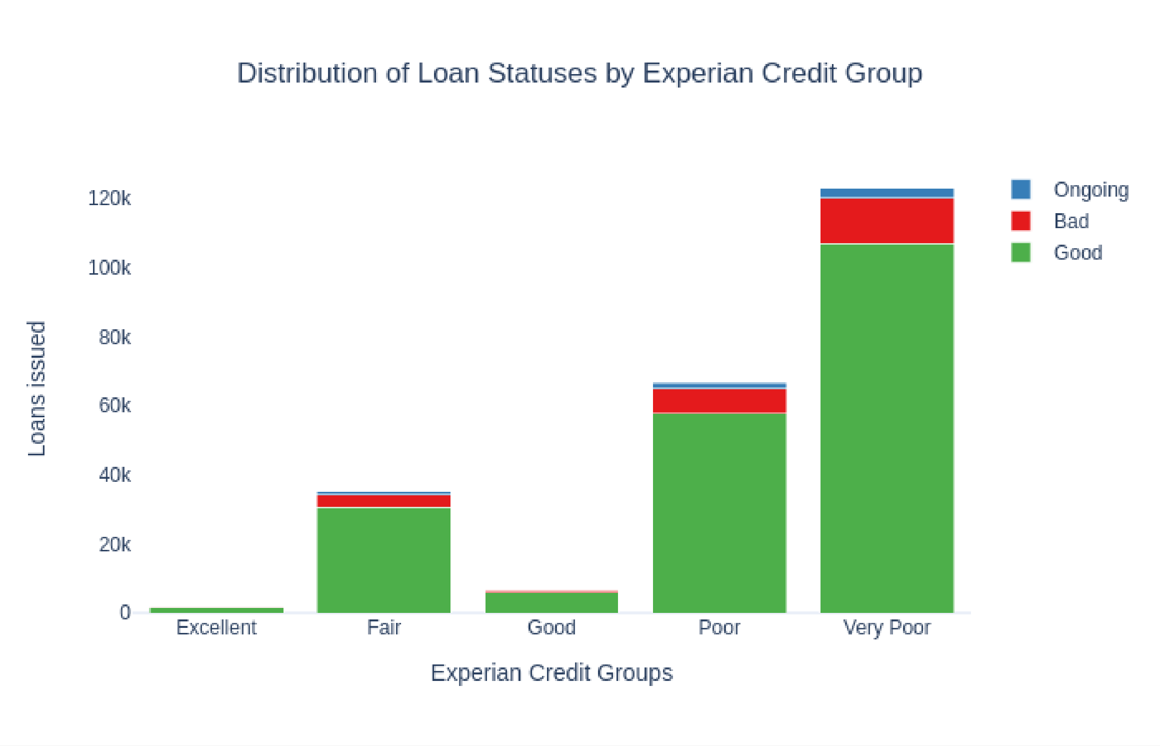
- 11% of loans which defaulted were paying an interest rate between 10%-20%

Data Understanding – What do applicants who default look like?



- 60% of loans were used to finance existing loans
- Of which 11% had defaulted

Data Understanding – What do applicants who default look like?



- Credit scores were mapped according to Experian Credit bands
- Credit score isn't everything!
- Many loans that were given, were fully-paid despite a low score

Excellent 961 - 999	You should get the **best credit cards, loans and mortgages** (but there are no guarantees).	Good 881 - 960	You should get most credit cards, loans and mortgages but the **very best deals may reject you.**	Fair 721 - 880	You might get OK interest rates but your **credit limits may not be very high.**	Poor 561 - 720	You might be accepted for credit cards, loans and mortgages but they may have **higher interest rates.**	Very Poor 0 - 560	You're more likely to be **rejected for most credit cards, loans and mortgages** that are available.

FINANCIAL CONDUCT AUTHORITY

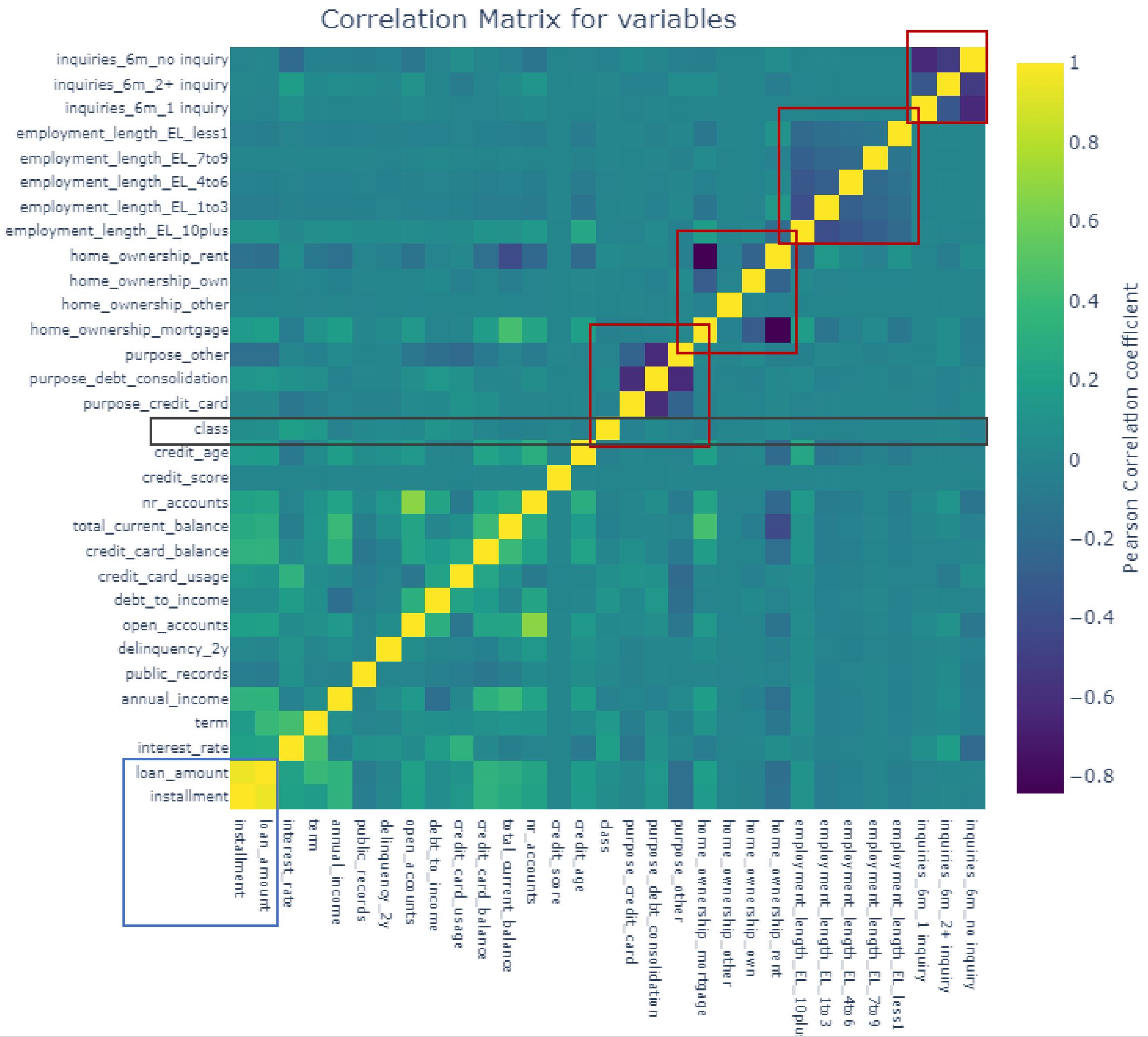
Data Preparation – Summary of data preparation

- Full details are available in the appendix and project files

- A binary response variable (named *class*) was created based slide 8, Ongoing samples were dropped
- Features that contributed to data leakage were removed i.e features that logically should not be present when there is a new application (e.g *amount payed)*
- Features that were text-based like *description* were removed – Natural language processing was not in scope
- Features with many groups such as *purpose* were re-categorised for simplicity and created as dummy variables
- Features that were ≥50% empty were removed – due to lack of information
- Features that were between 20% and 49% empty such as *total current balance* were imputed based on the median, segmented by *district*.

- Otherwise, samples were dropped (**5%** of samples were dropped in total)

Data Understanding – Correlations

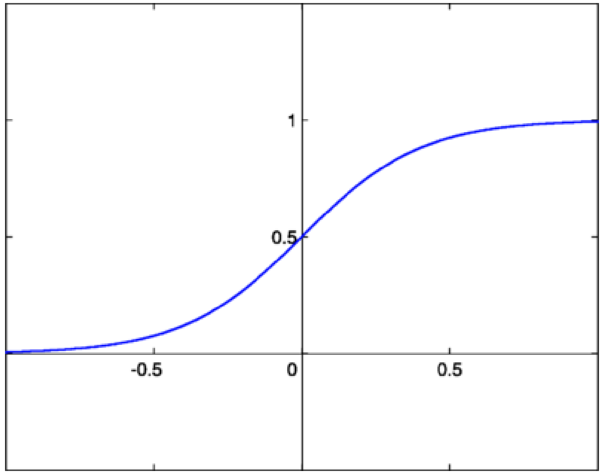


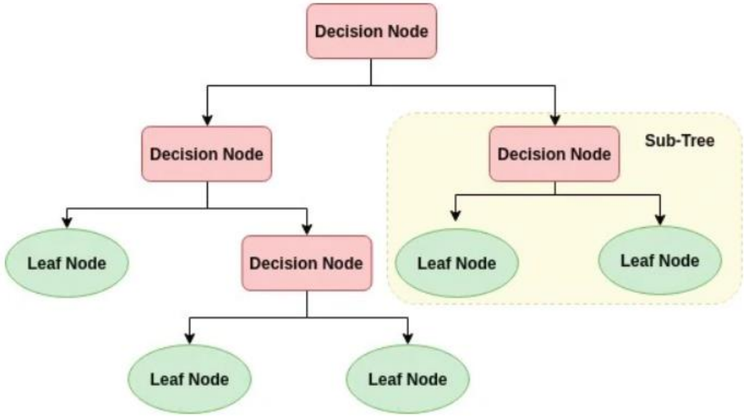
- Majority of features seem uncorrelated with each other and the *class* variable
- Some features are correlated such as *loan amount* & *installment*
- While others are negatively correlated due to the creation of dummy variables (creation of mutually exclusive variables)

FINANCIAL CONDUCT AUTHORITY

Modelling – Models chosen

- Three models were chosen due to their **simplicity**, ability to obtain a **probability of default** and **practical implementation**:
  - Logistic Regression
  - Huber Regression
  - Decision Tree



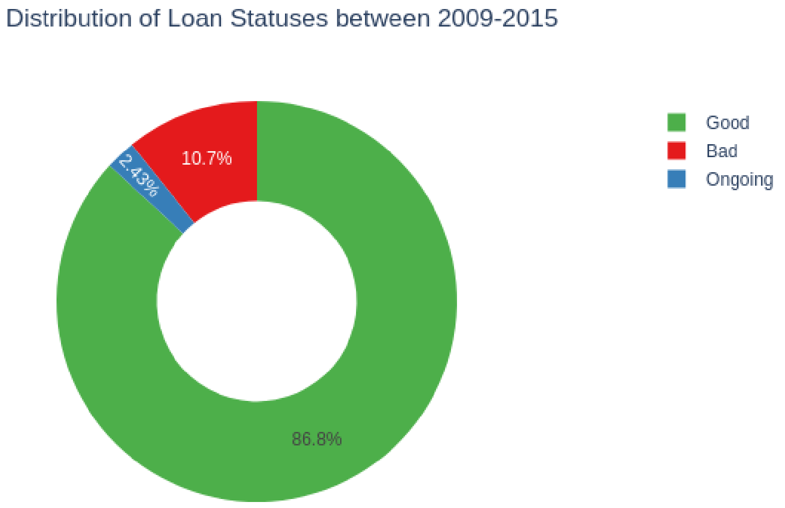


Modelling – Considerations

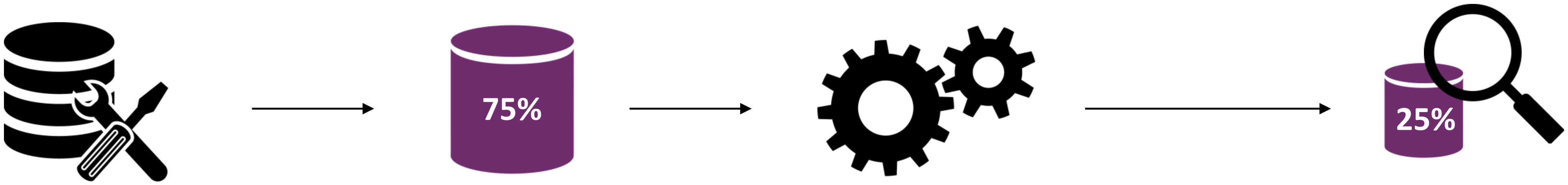
- Data is highly imbalanced and not equally distributed.

- Solution – Use Random Over-sampling
  - Initial experiments showed that using SMOTE encountered memory errors
  - Overfitting vs Underfitting → Apply regularisation to reduce overfitting

Method	Advantages	Disadvantages
Random Oversampling	• Intuitive • Easy to implement • No information loss	• May cause overfitting due to duplication of minority class
Random Undersampling	• Intuitive • Easy to implement	• May cause underfitting, due to information loss
SMOTE + variants	• Overcomes overfitting by creating samples • No information loss	• Impractical due to the calculations made to create samples • Harder to interpret



Modelling – Methodology



- Numerical features were normalised with the IQR method to handle any case of outliers (e.g *loan amount*)
- Samples were **shuffled** in and **minority class** was **oversampled** at random

- The data was split into 75% for model training and optimisation

- Models were trained and optimised on 75% of the data
- Where applicable, Gradient Descent was applied for efficient training and optimisation
- To reduce the number of features, Recursive feature elimination (RFE) was used.

- Models were evaluated on the remaining 25% of the data
- Metrics used to evaluate the model was *Accuracy, Recall & Area-under-the-curve (AUC)*

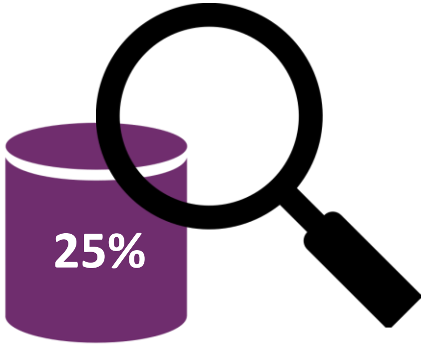
Evaluation – Evaluation metrics

- Accuracy, easy to understand - *alone* does not provide an adequate evaluation of the task at hand

- Recall attempts to answer the following question:
  - **What proportion of loan applicants *actually* defaulted?**
  - Having a low recall could lead to poor customer experience and decrease in revenue

- As a business, recall could help allocate funds to the minimum capital requirements

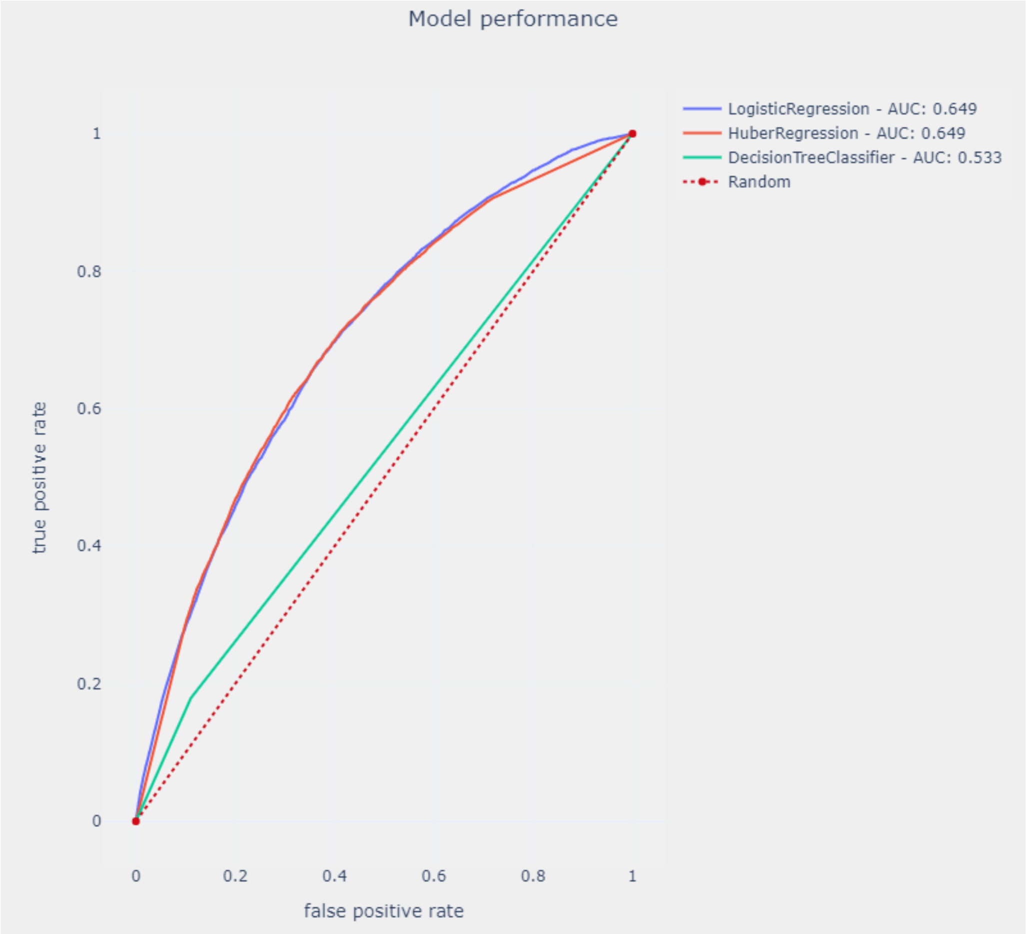
- AUC, visual representation of *recall* and *precision* and is a single metric defining how well the model separates defaulted and fully-paid loan applicants



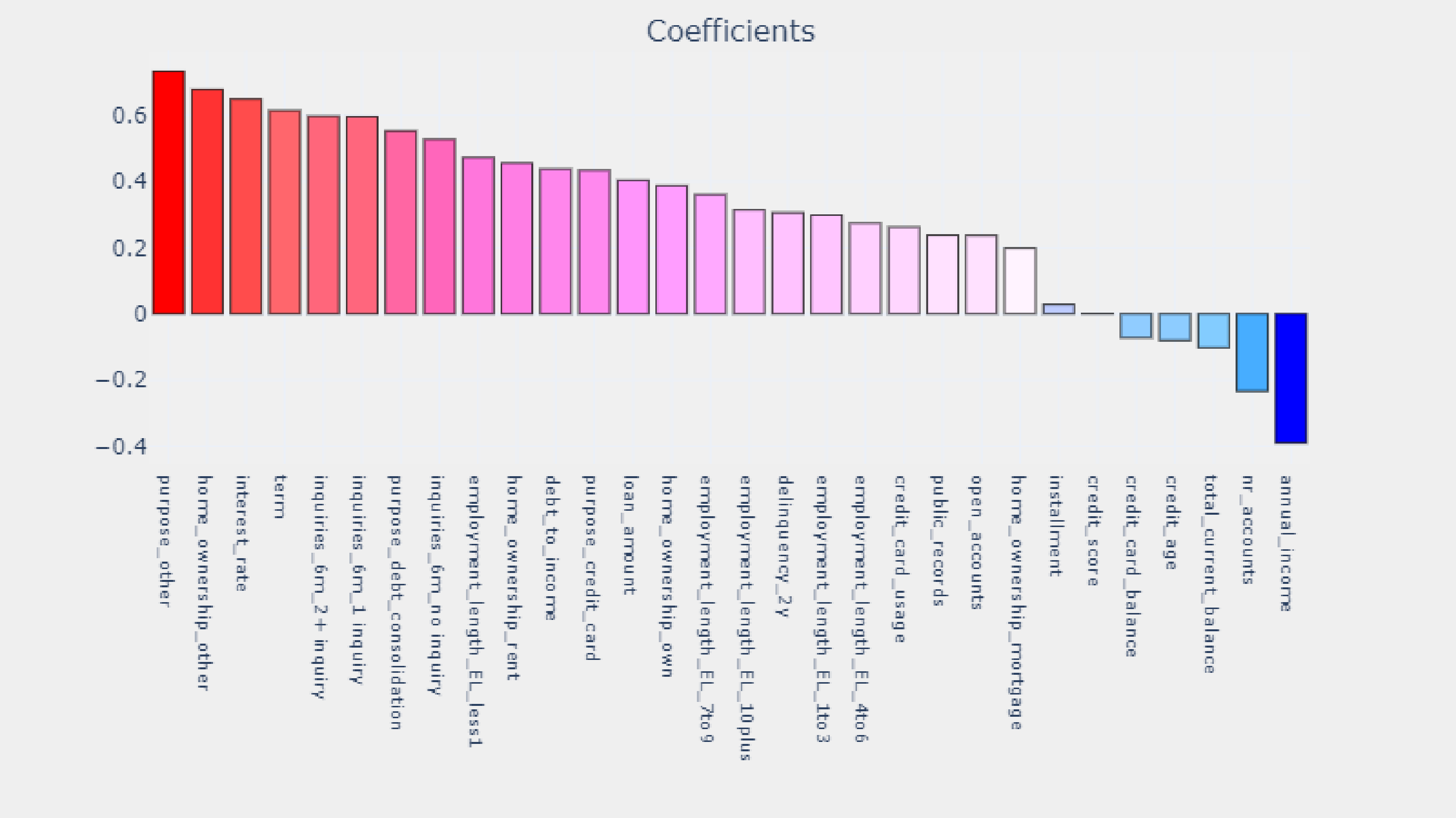
FINANCIAL CONDUCT AUTHORITY

Evaluation – Model performance using all variables

Model	Accuracy	Recall	AUC
Logistic Regression	65.3%	65%	64.91%
Huber Regression	**65.5%**	**65%**	**64.92%**
Decision Tree	81.3%	53%	53.3%

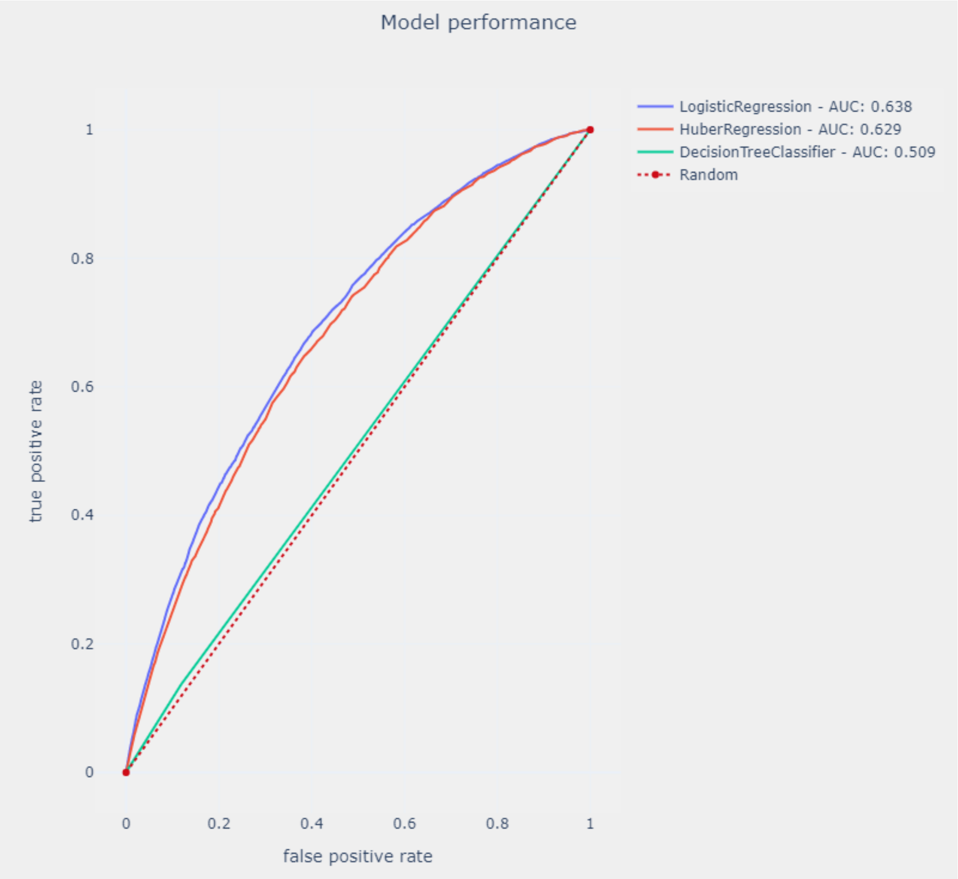


Evaluation – What are the factors correlated with default? (Huber Regression)



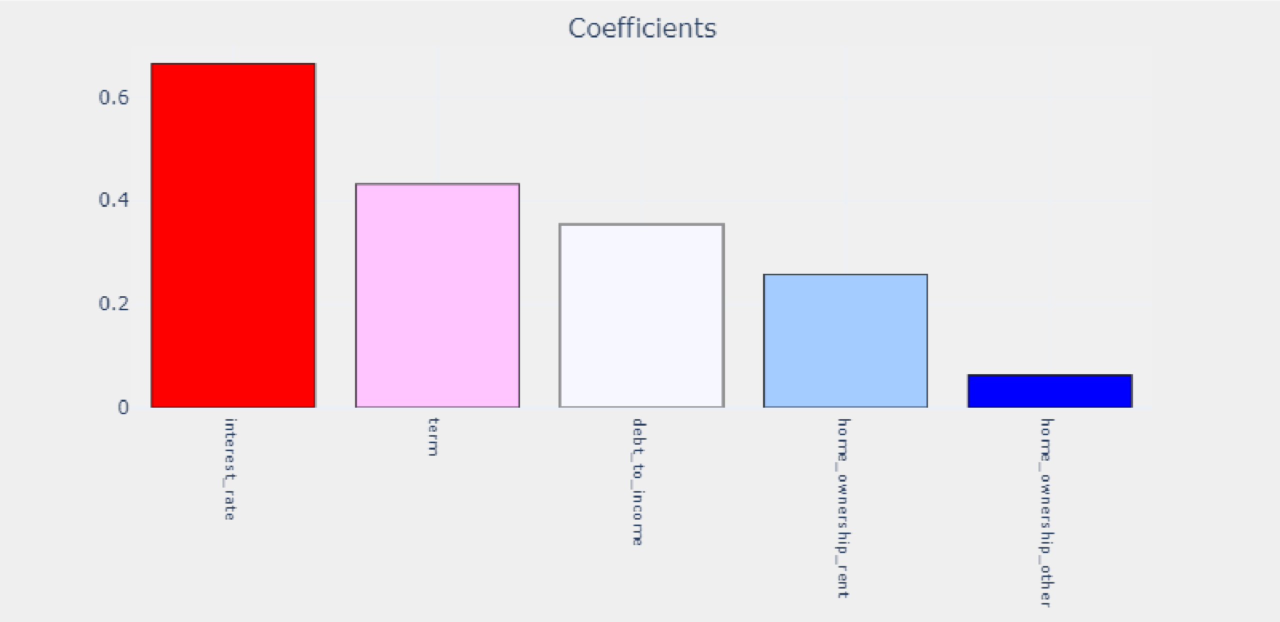
Evaluation – Model performance using five variables

Model	Accuracy	Recall	AUC
Logistic Regression	**65.5%**	**64%**	**63.83%**
Huber Regression	65.2%	63%	62.95%
Decision Tree	80.3%	51%	50.90%



FINANCIAL CONDUCT AUTHORITY

Evaluation – What are the factors correlated with default? (Logistic Regression)



Conclusions – Model to be used

# of Variables	Model	Accuracy	Recall	AUC
5	Logistic Regression	65.5%	64%	63.83%
31 (all variables)	Huber Regression	65.5%	65%	64.92%

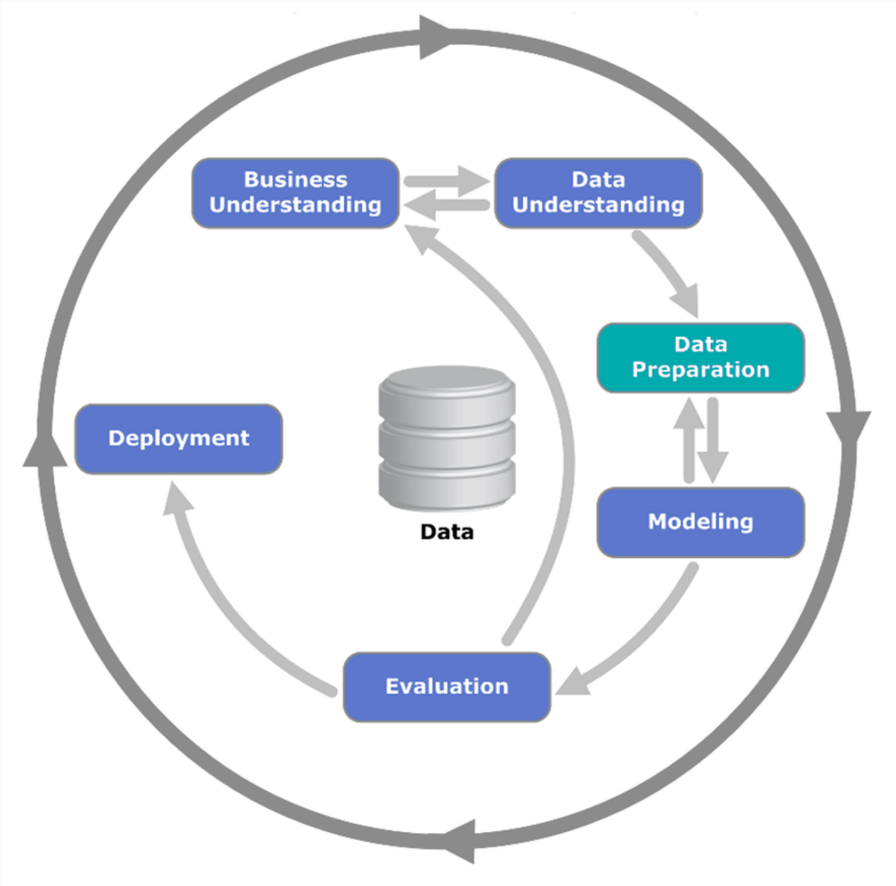
- The model that should be used is the simpler model, less is more
- Reduction from 31 features to 5 features, that is **83% reduction**
- From a practical perspective, it is much easier to collect 5 features than 31
- Trade-off is 1% pts difference in recall and AUC

Conclusions – Next steps

- Model should be used in **conjunction** with credit history of applicants to help the analysts' make an informed decision

- A traffic light system would be able to assist analysts' by obtaining their input on defining probability of default thresholds on the model
  - Allowing analysts' to prioritise applicants

Appendix - Approach



- `Cross-industry standard process for data mining` (CRISP-DM).
- A process for applying data science methodology to answer business questions

FINANCIAL CONDUCT AUTHORITY

Appendix – Documentation – Project Structure

Project Organization

```
├── LICENSE
├── README.md          <- The top-level README for developers using this project.
├── jupyter_setup.sh   <- Run this (`sh jupyter_setup.sh`) to enable QoL jupyter extensions, like codefolding.
├── run_docs.sh        <- Run this (`sh run_docs.sh`) to see your documentation locally.
├── data
│   ├── external       <- Data from third party sources.
│   ├── interim        <- Intermediate data that has been transformed.
│   ├── processed      <- The final, canonical data sets for modeling.
│   └── raw            <- The original, immutable data dump.
│
├── docs               <- A default Mkdocs project; see mkdocs.org for details
│
├── models             <- Trained and serialized models, model predictions, or model summaries
│
├── notebooks          <- Jupyter notebooks. Naming convention is a number (for ordering),
│                         the creator's initials, and a short `-` delimited description, e.g.
│                         `1.0-jqp-initial-data-exploration`.
│
├── references         <- Data dictionaries, manuals, and all other explanatory materials.
│
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures        <- Generated graphics and figures to be used in reporting
│
├── requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
│                         generated with `pip freeze > requirements.txt`
│
├── setup.py           <- makes project pip installable (pip install -e .) so src can be imported
├── src                <- Source code for use in this project.
│   │
│   ├── data           <- Scripts to download, generate data or clean data
│   │
│   ├── features       <- Scripts to turn raw data into features for modeling
│   │
│   ├── models         <- Scripts to train models and then use trained models to make predictions
│   │
│   └── visualization  <- Scripts to create exploratory and results oriented visualizations
```

Appendix – Documentation – EDA

- Mapping districts to regions:
  - https://geoportal.statistics.gov.uk/datasets/ward-to-local-authority-district-to-county-to-region-to-country-december-2017-lookup-in-united-kingdom-version-2

- Experian Credit Rating groups
  - https://www.experian.co.uk/consumer/mortgages/guides/credit-and-mortgages.html

Appendix – Documentation – Data cleansing process

Data Cleansing Process

- Features that had >= 50% nulls were dropped as there is not a significant amount of information to do anything with
- All textual data was lower cased
- All trailing and leading whitespaces were removed
- `credit_score` and `annual_income` was changed to numeric type
- `earliest_credit_line` was changed to datetime format
- `employment_length`, `homw_ownership`, `inquiries_6m` and `purpose` was re-categorised
- `total_current_balance` had its values imputed with the median, segmented by `district`
- `credit_age` was created from `earliest_credit_line` by taking subtracting 2015 (latest date in data) from `earliest_credit_line`
- `class` was created from `loan_status`, 1 if fully paid, 0 if defaulted, charged-off or late
- Textual and leaky features were dropped.
- Remaining samples were dropped, ~5% of data reduced.

Appendix – Documentation – Variables after data cleansing

Variables - Post cleansed

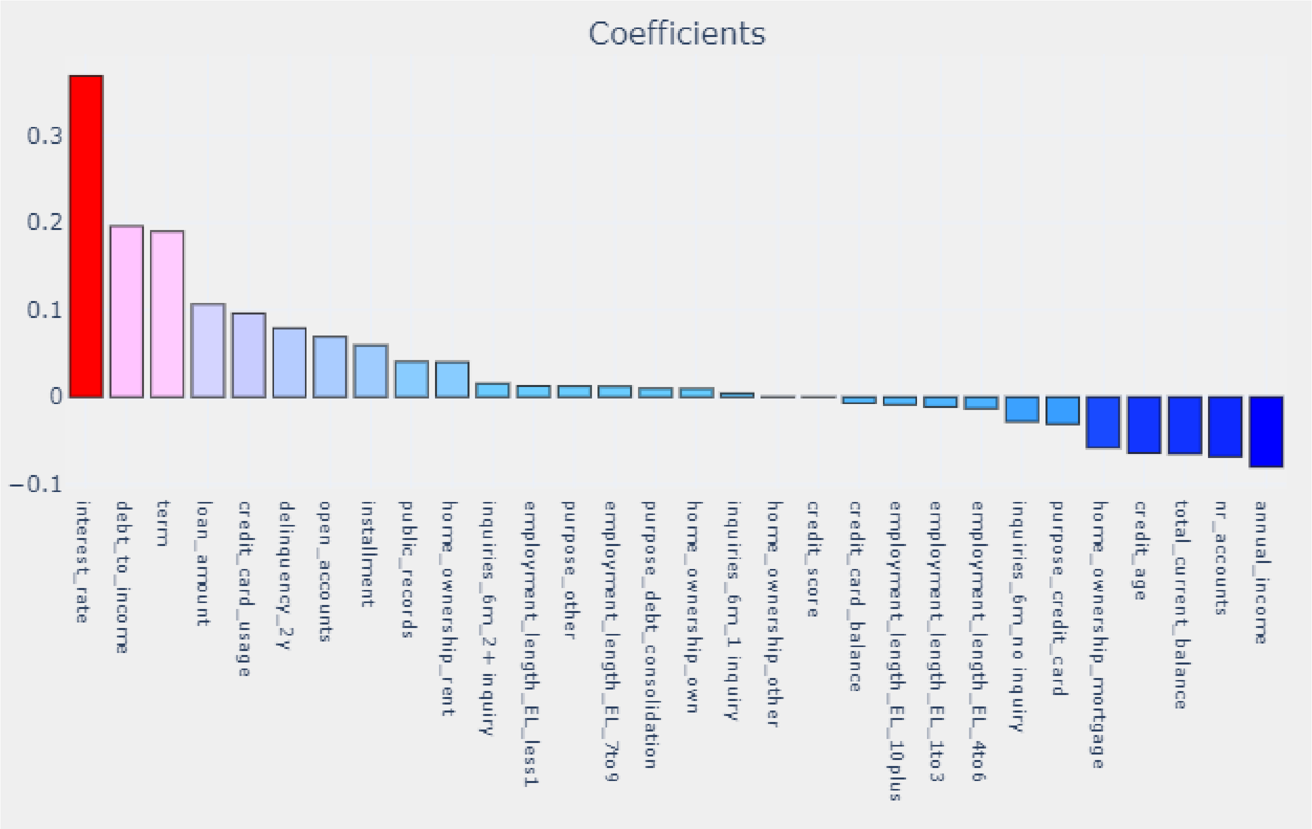
- `account_id`: See original data-dictionary
- `installment`: See original data-dictionary
- `loan_amount`: See original data-dictionary
- `interest_rate`: See original data-dictionary
- `term`: See original data-dictionary
- `purpose`: See original data-dictionary
- `home_ownership`: See original data-dictionary
- `annual_income`: See original data-dictionary
- `employment_length`: See original data-dictionary
- `public_records`: See original data-dictionary
- `delinquency_2y`: See original data-dictionary
- `inquiries_6m`: See original data-dictionary
- `open_accounts`: See original data-dictionary
- `debt_to_income`: See original data-dictionary
- `credit_card_usage`: See original data-dictionary
- `credit_card_balance`: See original data-dictionary
- `total_current_balance`: See original data-dictionary
- `nr_accounts`: See original data-dictionary
- `credit_score`: See original data-dictionary
- `credit_age`: Number of years from earliest_credit_line till 2015
- `class`: {0,1} 0: Customer paid back loan, 1: Customer defaulted loan payment

Appendix – Documentation – Variables dropped for Machine Learning

Variables that were dropped for Machine Learning

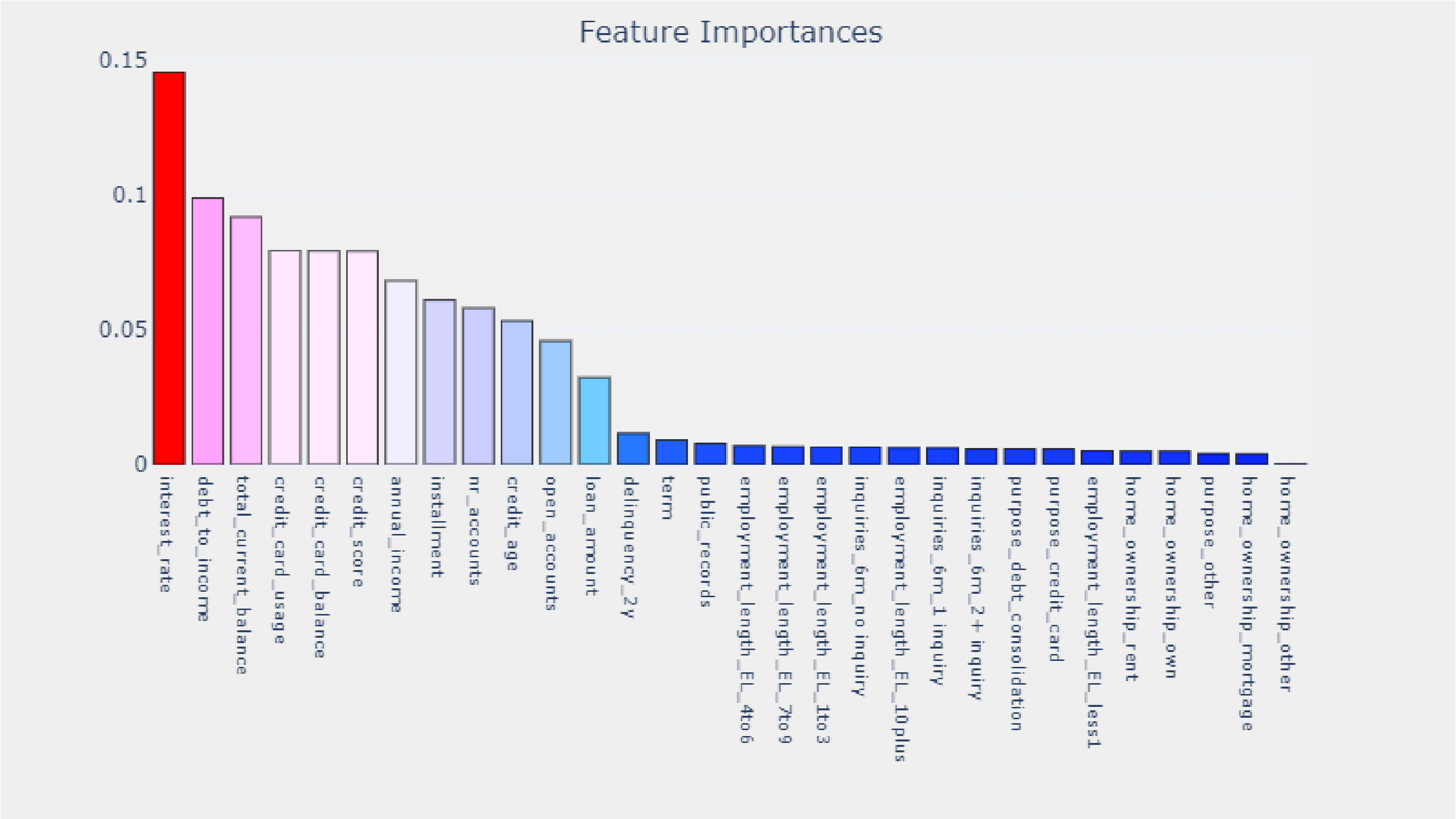
- `description`: >50% null
- `last_record_months`: >50% null
- `last_delinquency_months`: >50% null
- `last_derog_months`: >50% null
- `title`: text field, non-NLP task
- `job_title`: text field, non-NLP task
- `district`: text field, non-NLP task
- `postcode_district`: text field, non-NLP task
- `loan_status`: Data leakage, when predicting new application this would not be known
- `year`: Data leakage, when predicting new application this would not be known
- `amount payed`: Data leakage, when predicting new application this would not be known
- `earliest_credit_line`: Date field, used to create `credit_age`

Appendix – Model coefficient using all variables (Logistic Regression)



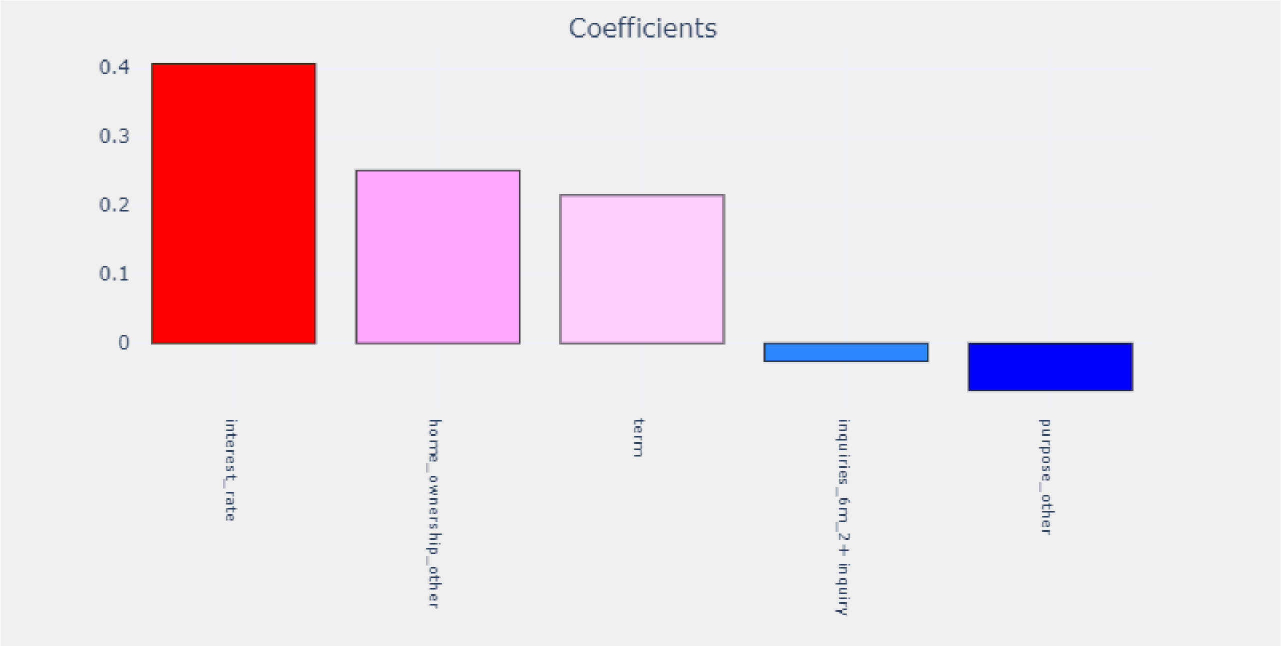
- See reports/figures/*.html for **all interactive reports and metrics**

Appendix – Model feature importances using all variables (Decision Tree)



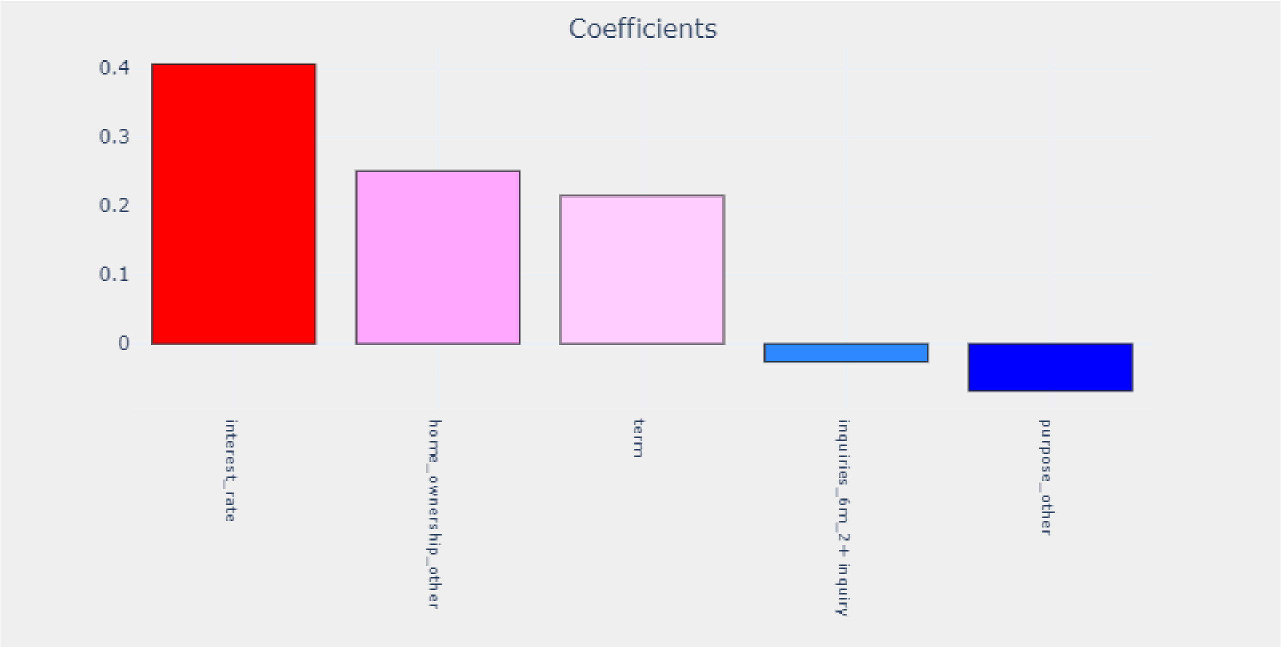
- See reports/figures/*.html for **all** **interactive** **reports and metrics**

Appendix – Model performance using five variables
Huber Regression



- See reports/figures/*.html for **all** **interactive** **reports and metrics**

Appendix – Model performance using five variables
Decision Tree



- See reports/figures/*.html for **all** **interactive reports and metrics**