## Methodological approach

Probability of default is the probability that a borrower will be unable to repay their debt obligation. Therefore it is extremely important for lenders to be able to calculate this risk as accurately as possible in order to prevent losses or miss out on opportunities. This analysis aimed to identify the factors that affect this risk the most, given the dataset provided, and to train a model to accurately predict the probability a company will default within a year.   
The analysis started with data exploration and validation i.e. identify incorrect feature values and outliers, check for data consistency within groups (i.e. whether the postcode or the industry code within a group is the same) as well as missing values and ways to deal with them. It relies on the assumption that if a company's status is "Default", this company defaulted during the latest period for which data was provided (latest accounts date) or the period right after that.  
A Random Forest Classifier was trained to predict the probability of default and its performance was evaluated using AUC score. Class imbalance was identified and a random oversampling technique was used. The analysis then used a forward feature selection algorithm to identify the best feature combination that maximizes AUC.  
  
Feedback to engineering team

* There are many missing values in the data that have no value for modelling - think of ways to avoid that
* Add data validation checks to improve data quality (i.e. avoid having values such as "-100000" in a datetime field)
* Look for other data sources to increase the number of examples and/or features

Guidance to the business

* Overall the model shows good performance with 84% accuracy but there are still some misclassified examples that should be considered
* It is a good starting point and there's much room for improvement - so it is a promising model (approach)
* Combine model's results (especially when probabilities of default are close to the threshold) with other business insight for validation and better accuracy

Two things I would improve  
Train and compare more models (logistic regression, SVM...) and use a grid search technique to tune hyperparameters  
Try other techniques to deal with class imbalance (SMOTE)