# Project Summary

## Executive Summary

- The best predictions were obtained from a Gradient Boosting based model with a ROC-AUC of 0.771.
- Another model based on Random Forest also gave a comparable result. This model would also have been equally good in prediction accuracy, but the gradient boosting model was used because it was faster to train and obtain predictions.
- Preprocessing: Missing values were handled by replacing them with global mean. Categorical features were one hot encoded.
- Feature Selection: Highly correlated(Pearson correlation>0.8) features were removed. Features with more than 80% of the values missing were removed before modelling.
- Different models like Logistic regression, Decision trees, Random forest, Gradient boosting trees and Extreme gradient boosting(XGBoost) were trained on the data.
- Extensive hyperparameter tuning and evaluation were performed with cross-validation.
- The introduction of interaction features(all pairwise products of numerical features) and oversampling didn't improve the performance of the models.

## Exploratory Data Analysis

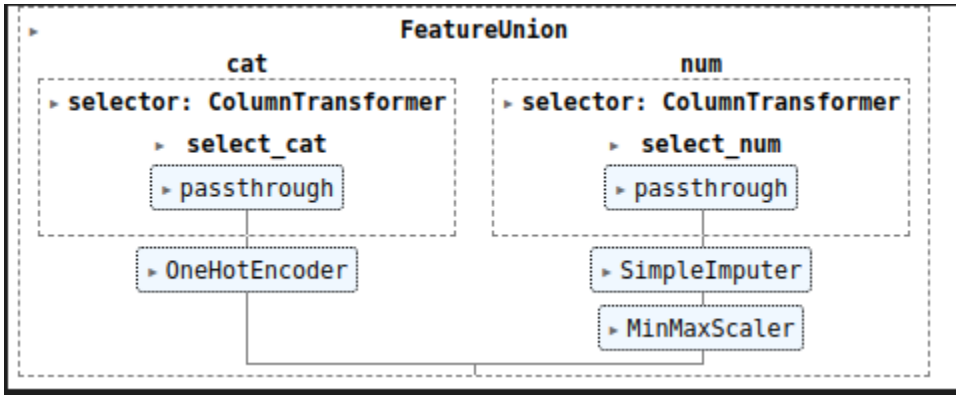
### Key takeaways

- All numerical features had missing values. Features N25 - N32 had more than 80% of their values missing.
- The following pairs of numerical features had high Pearson correlation
    - N3 - N6
    - N4 - N5
    - N4 - N8
    - N5 - N8
    - N7 - N20
    - N33 - N34

The full analysis can be found at 01_eda.ipynb

## Pre-processing Steps

- 20% of the training data was set aside before any analysis was performed to evaluate the models.
- Categorical variables were one hot encoded
- Missing values were handled by SimpleImputer with “mean” strategy
- Numerical variables were scaled with MinMaxScaler
- Pipelines were used in all pre-processing steps to avoid any data leakage during cross-validation.
- The following is the pre-processing pipeline used.



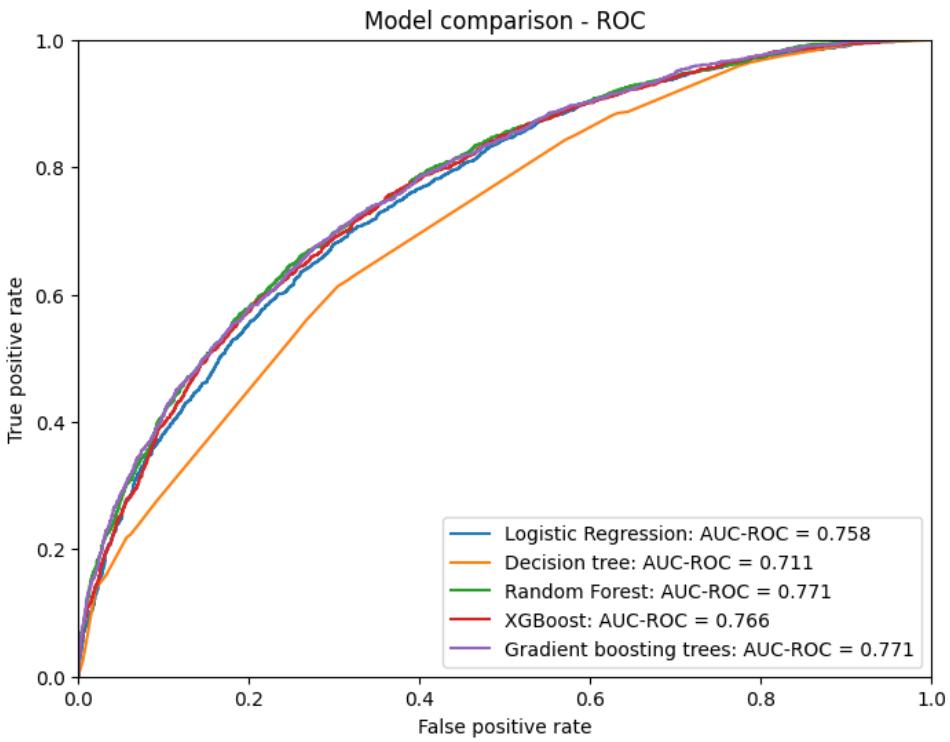

## Model Selection

Models were trained with cross-validation and hyperparameter tuning was done with the help of GridSearchCV.

- Both Random forest and Gradient boosting performed the best with an AUC of 0.771.
- Gradient boosting with the following hyperparameters was chosen for the final prediction.
    - n_estimators = 300
    - learning_rate = 0.1

- This choice is due to faster training and inference time observed for the Gradient boosting model with this dataset
- The selected model was retrained with the whole training set before final predictions were made on the test data.

The following plot gives a comparison of the model performance.




## Other Experiments

- Oversampling with SMOTENC was tried since the dataset had a slight imbalance. Model performance didn't improve
- Interaction features for numerical variables also failed to give any appreciable result except in the case of the Logistic regression model.

## Future Directions

- Feature Engineering informed by domain knowledge will be critical in improving the model.
- If the feature names were known, a subset representing the age, education, employment, location etc (i.e demographic information) could be used to cluster the candidates and a downstream model could use the cluster label as a feature.
- Similar to the point above, customer segmentation can be performed first and separate models can be built for each customer segment.