Glossary:

- **Accuracy**- total correctly classified examples by the model divided by the total number of classified examples.
- **Bias**- a systematic error that occurs in the machine learning model itself due to incorrect assumptions in the ML process.
- **Bias-variance trade-off**- the property of a model that the variance of the parameter estimated across samples can be reduced by increasing the bias in the estimated parameters.
- **Binning-** a way to group a number of more or less continuous values into a smaller number of "bins".
- **Collinear-** correlation between predictor variables (or independent variables), such that they express a linear relationship in a regression model.
- **Confusion matrix**- a table that summarizes how successful the classification model is at predicting examples belonging to various classes.
- **Correlation**- a measure of how well two or more variables are related.
- **F1-score-** a weighted average of precision and recall
- **Feature Space**- the n-dimensions where your variables live excluding the target variable.
- **Hyperparameters**- parameters whose values control the learning process and determine the values of model parameters that a learning algorithm ends up learning.
- **Imbalanced data**-a lack of balance between the target variable's classes.
- **OOB error**- the average error for each calculated using predictions from the trees that do not contain in their respective bootstrap sample
- **Precision-** actual correct prediction divided by the total prediction made by the model.
- **Predictors**- predictor variables help the model choose which target variable to apply to each case.
- **Recall-** calculated as the number of true positives divided by the total number of true positives and false negatives.
- **Regularization**- techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting.
- **ROC-curve-** a graph showing the performance of a classification model at all classification thresholds
- **Sparse data**- an array of data where numerous components have an estimation of zero value.
- **Supervised learning**- An ml approach, in which the algorithm is trained on input data that has been labelled for a particular output.
- **Variance**- changes in the model when using different portions of the training data set.
- **Word cloud**-Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance.

Errors experienced:

- I tried to use mnrfit on my dataset but it threw me the error stating “Error using eps CLASSNAME argument must be a class that supports EPS, such as 'double' or 'single’.

Solved using:

Tried to solve this error and came across a Linear model for binary classification of high-dimensional data that either uses SVM or logistic regression as the base model.

- Another error that I experienced was with the datatype because I was extracting the table columns directly and the data type was a table, whereas the needed data type was double

Solved using:

The error got solved by using table2array which converts the type table to double.

- Was unable to plot the roc curve using perfcurve. Gave me a lot of errors.

Solved using

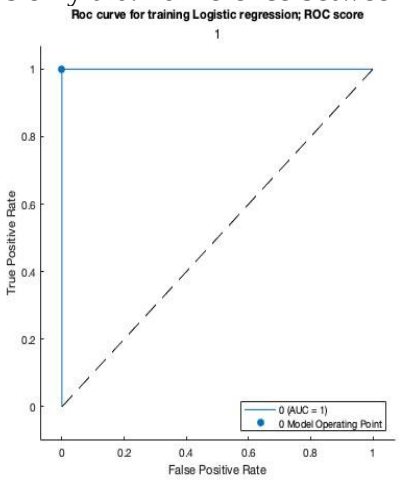
Roc.metrics

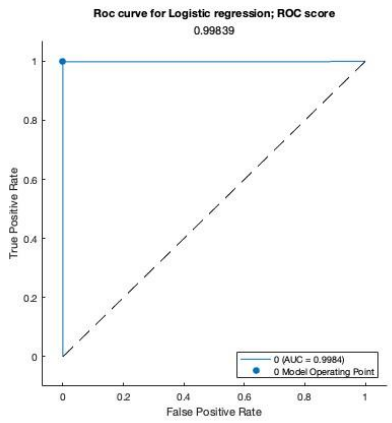
Negative results:

The bias for my optimised Linear regression model is negative, leading to sigmoid output near 0.

I am getting a way too high score for my AUC for LR i.e. around 0.99 which looks unreal. One possible reason might be that I have an imbalanced dataset, high recall, and low precision. I think the model is predicting most of the ones at the higher end of prediction probabilities, but most of the outcomes at the higher end of prediction probabilities are still 0.

There is only a 0.1 difference between training and testing AUC.





Choice of parameters:

Tried fitting random forest with 50 bagging trees at first but the fit time was too high so changed them to 20.

Tried Setting the optimisehyperparameter to auto but it took a significant time to run, so removed it because it wasn’t bringing any significant results.

The main thing that helped in improving the Logistic regression model was the ridge regularization with the solver combination of “Stochastic gradient descent”, and “Limited-memory BFGS”.

Supplementary notebook

Data visualisation was done in python. Attaching the colab link below.
https://colab.research.google.com/drive/1-ZrBky9sRYKcuTkf6U2hNjeaFN1PyTRd?usp=sharing