CMT307 Coursework

Introduction

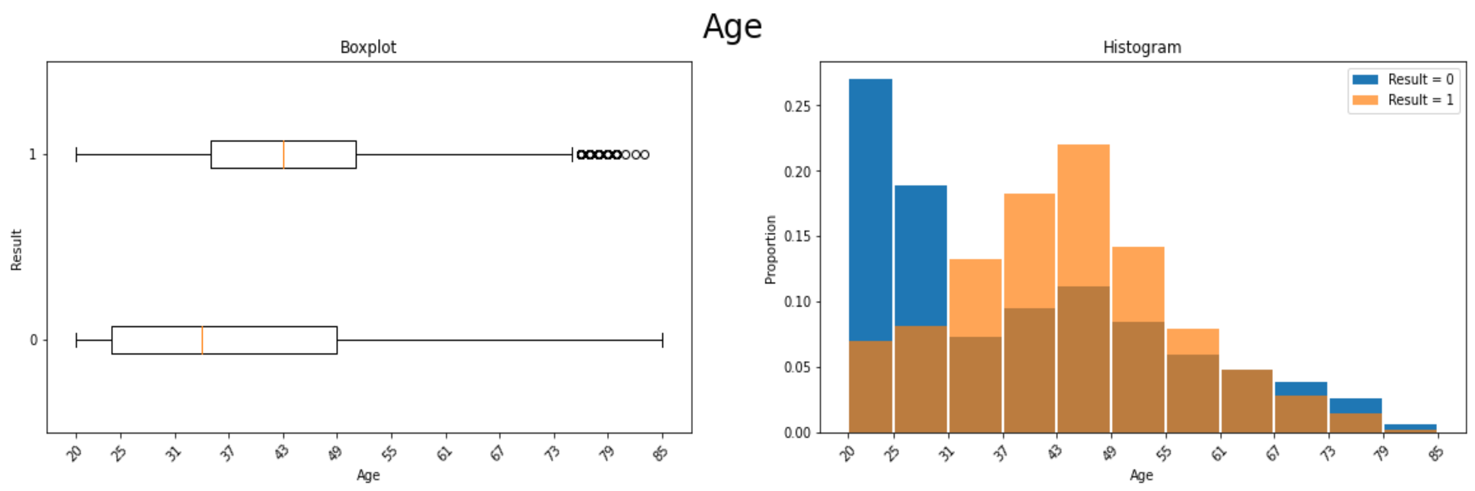
We have been given a data set from a health insurance provider which contains information on patients who were offered car insurance by the same provider. Our task was to develop machine learning models to predict whether customers did or did not buy car insurance. The data included a variety of features, including binary and non-binary categorical variables, and numerical features.

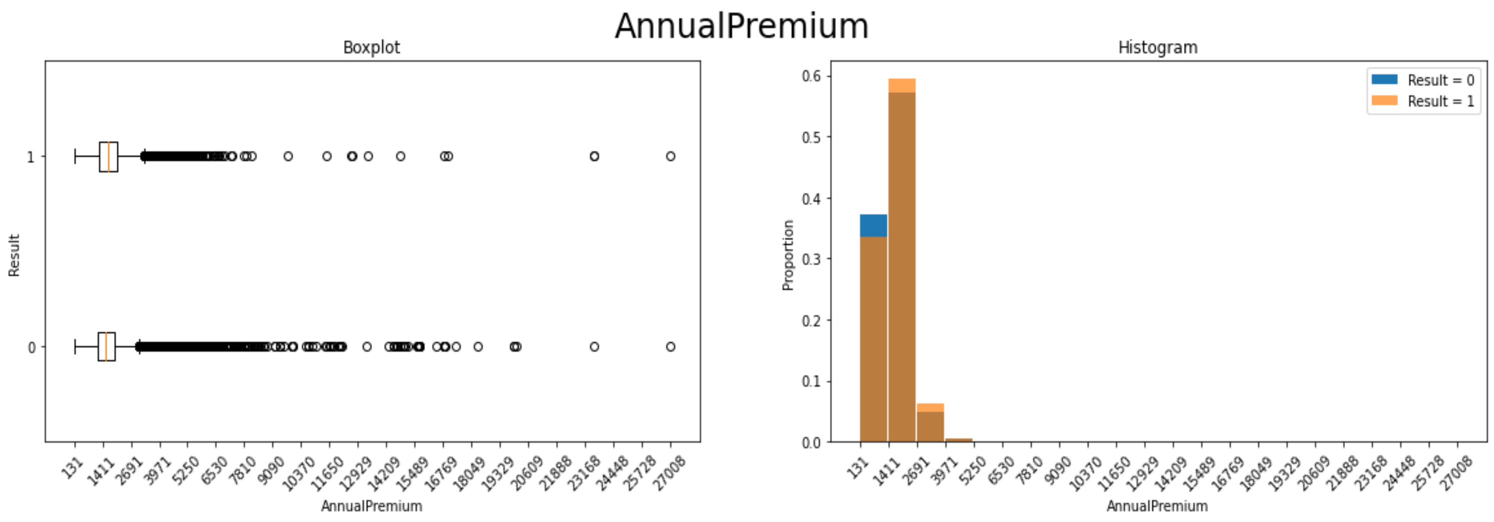
This report outlines the methods used in the process of developing the models used for classification and provides justification where necessary for any decisions made.

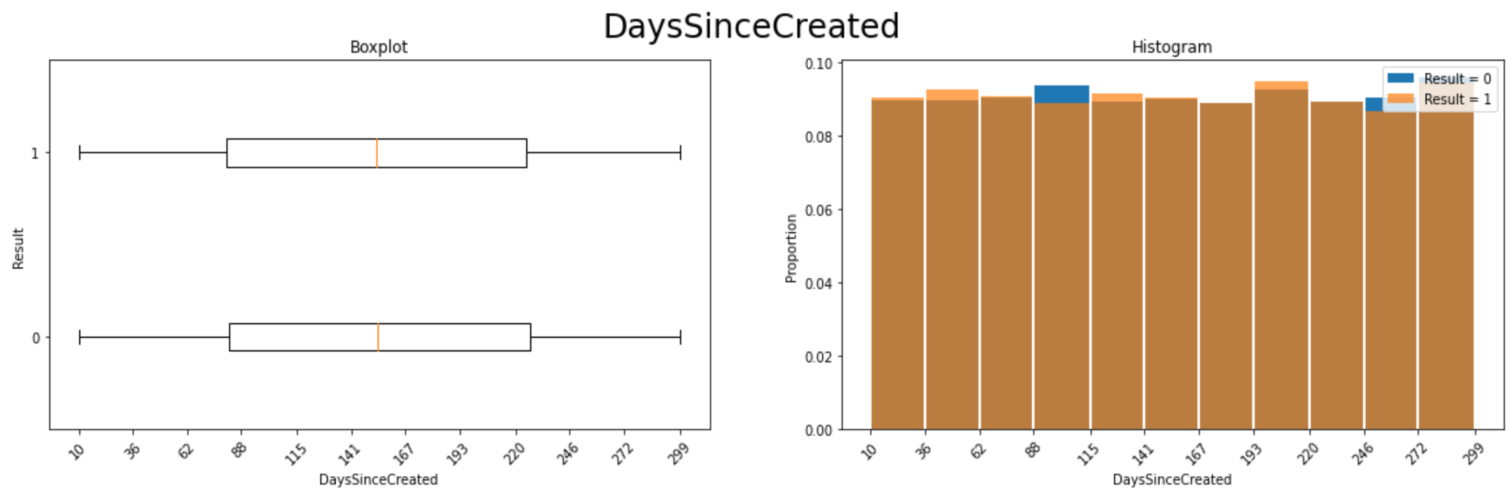
Data Exploration

The data contains 12 columns, including the target variable Result, and before splitting contains just over 300,000 rows. I continue the exploration using a sampled 70% of the data as training data.

I start by looking at the numeric features: Age, AnnualPremium, and DaysSinceCreated. For each variable, I construct a boxplot and a histogram of the variable split by Result. These plots are shown below.







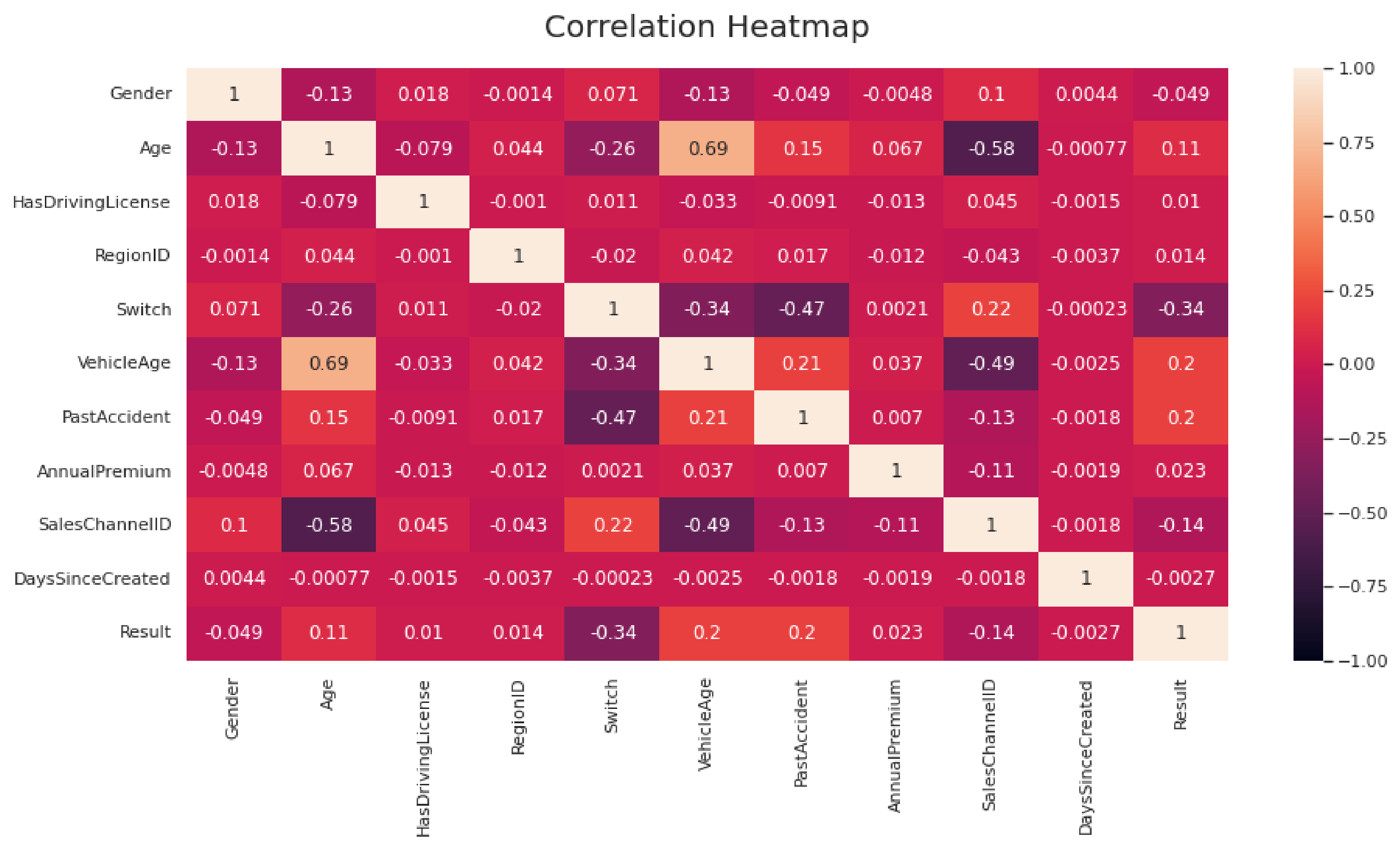
For Age, the mean is 38.87 and the median is 36, with an inter quartile range (IQR) from 25-49 and a range from 20-85. Both plots show a potential difference in the distribution of age by Result, especially the histogram. The histogram suggests that the average age is greater for those customers who purchased car insurance.

For AnnualPremium, it is hard to see if there are large differences, especially in the histogram, since the upper tail is very long and low. 75% of values are below £1,970, but the maximum value is around £27,000. I also plotted a histogram of the feature for values greater than £5,000, but it showed the same results (large proportion between 5000-7000 and then a long low tail.

Finally, for DaysSinceCreated, there doesn't seem to be a difference in the distribution by Result and the values are very uniformly spread in the data.

There are 7 categorical variables, with varying amounts of levels. Gender, HasDrivingLicense, Switch, and PastAccident are binary variables. VehicleAge has 3 leves, while RegionID and SalesChannelID have many more.

I also constructed a correlation heatmap of all the features (excluding id, which is a row identifier), shown below.



	Gender	Age	HasDrivingLicense	RegionID	Switch	VehicleAge	PastAccident	AnnualPremium	SalesChannelID	DaysSinceCreated	Result
Gender	1	-0.13	0.018	-0.0014	0.071	-0.13	-0.049	-0.0048	0.1	0.0044	-0.049
Age	-0.13	1	-0.079	0.044	-0.26	0.69	0.15	0.067	-0.58	-0.00077	0.11
HasDrivingLicense	0.018	-0.079	1	-0.001	0.011	-0.033	-0.0091	-0.013	0.045	-0.0015	0.01
RegionID	-0.0014	0.044	-0.001	1	-0.02	0.042	0.017	-0.012	-0.043	-0.0037	0.014
Switch	0.071	-0.26	0.011	-0.02	1	-0.34	-0.47	0.0021	0.22	-0.00023	-0.34
VehicleAge	-0.13	0.69	-0.033	0.042	-0.34	1	0.21	0.037	-0.49	-0.0025	0.2
PastAccident	-0.049	0.15	-0.0091	0.017	-0.47	0.21	1	0.007	-0.13	-0.0018	0.2
AnnualPremium	-0.0048	0.067	-0.013	-0.012	0.0021	0.037	0.007	1	-0.11	-0.0019	0.023
SalesChannelID	0.1	-0.58	0.045	-0.043	0.22	-0.49	-0.13	-0.11	1	-0.0018	-0.14
DaysSinceCreated	0.0044	-0.00077	-0.0015	-0.0037	-0.00023	-0.0025	-0.0018	-0.0019	-0.0018	1	-0.0027
Result	-0.049	0.11	0.01	0.014	-0.34	0.2	0.2	0.023	-0.14	-0.0027	1

There appear to be some fairly strong connections between many of the features in the data, though one should be careful when interpreting the coefficients for the ID variables (RegionID and SalesChannelID), because the matrix treats these as numeric variables. For example, the coefficient for SalesChannelID and Age, which is -0.58, means that there is a fairly strong negative relationship between age and the increasing number associated with the level of SalesChannelID.

Data Pre-Processing

There were seven features that contained missing values, with two of them missing around 50% of their values. The first action I performed was removing some features from the data. I removed id because it is just a row identifier, Gender because the correlation matrix shows that it has little to no impact on Result while still being roughly equally represented in the data (unlike HasDrivingLicense which has mostly 1), as well as Switch and PastAccident because they are missing around 50% of their data points. While these last two could potentially be quite impactful on Result, I didn’t think that a basic imputation method would be sufficiently accurate at replacing missing values without increasing the bias of the features’ effects too much. For the remaining features containing missing values, I used a simple imputer to impute the categorical features with the mode and the remaining numeric feature, Age, with the median.

Then, I began on feature transformation, starting with simply converting AnnualPremium from a string starting with “£” to a float. I scaled the numeric features to be between zero and one and using OneHotEncoder and a binary encoder for the categorical variables. I used the former for variables with two or three levels and the latter for those with many levels (RegionID and SalesChannelID).

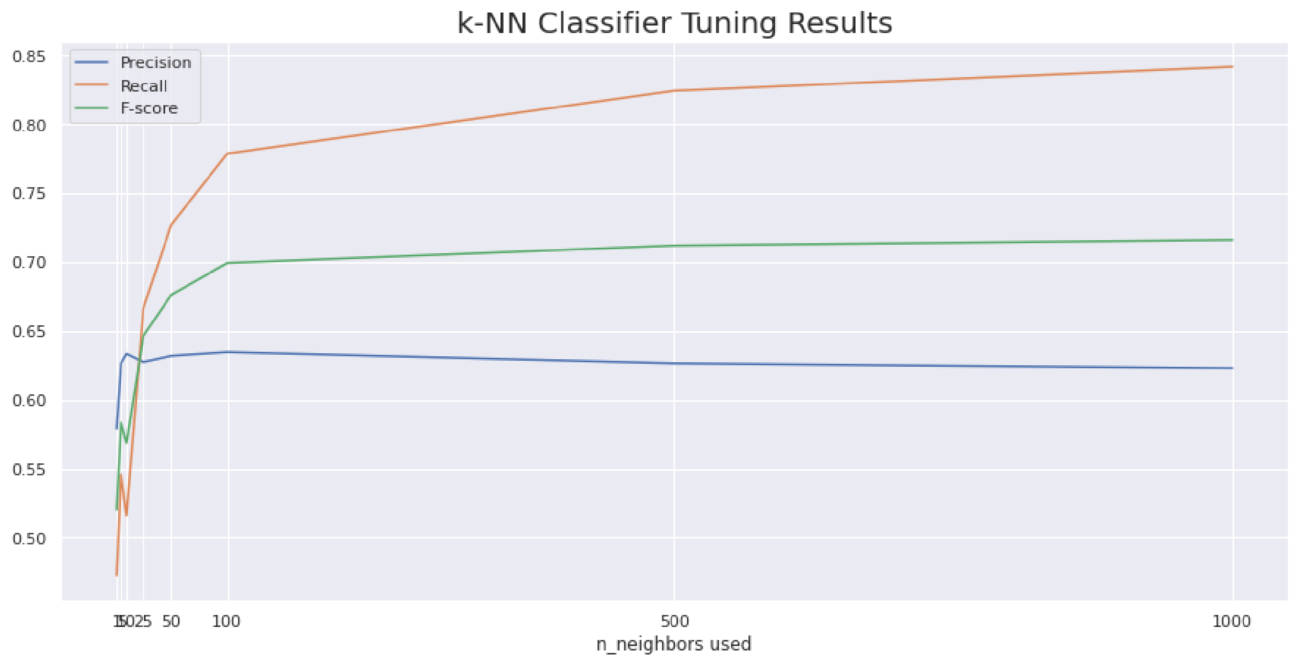
Given how unbalanced the data was (roughly 85% 0 and 15% 1), I decided to use oversampling to balance the data since it would otherwise be feasible for a model to predict only 0 and get a good accuracy score. I used a naïve oversampling method by randomly sampling from the positive observations until the data was balanced.

Modelling

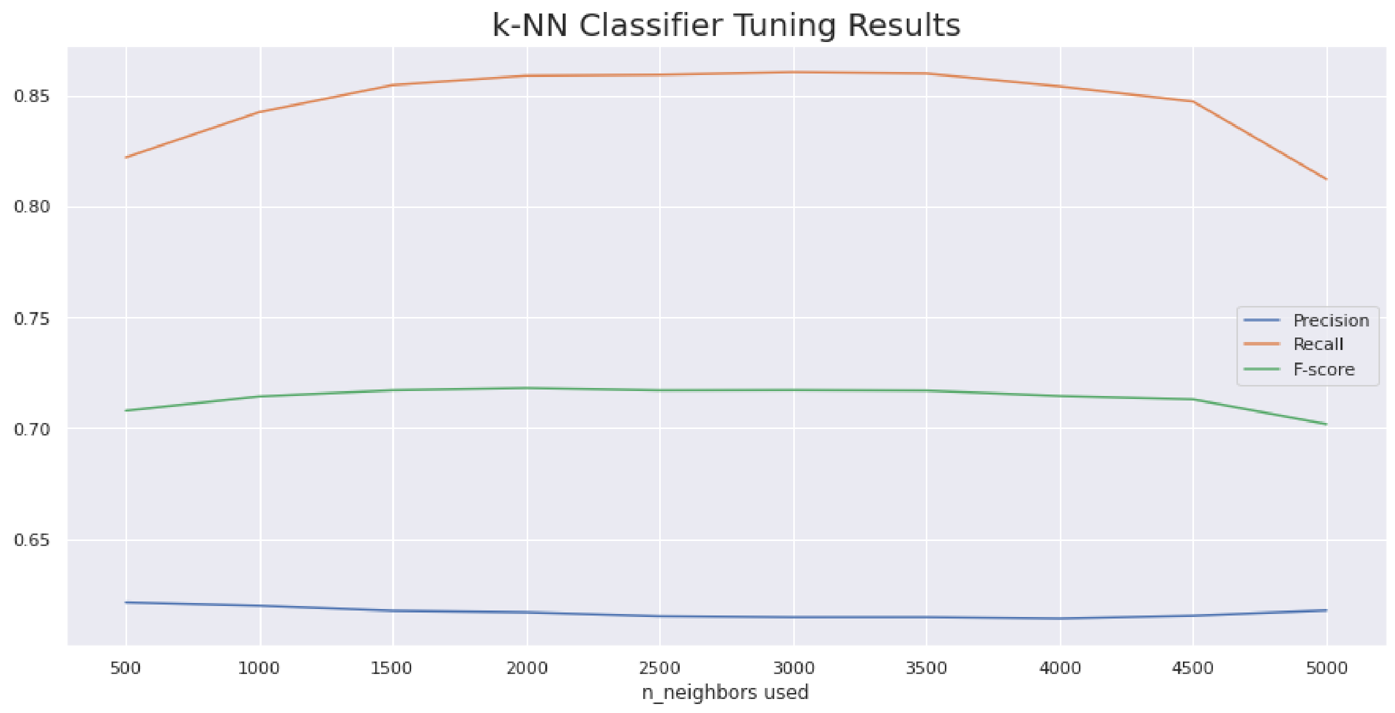
For optimising the models, I decided to take a stratified sample of 10,000 observations from the training and test data so that optimisation would not take too long. I would then use the optimised model fitted on the full data and evaluated against the full test data. The three models that were selected were: k-nearest neighbours, logistic regression, and decision trees. These models cover linear and non-linear, simple and complex models. I provided some brief descriptions of the methods in the code file.

k-NN:

I optimised the number of neighbours used by the model by comparing precision, recall and F-scores for each model, starting with a variety of values from 1-1000.



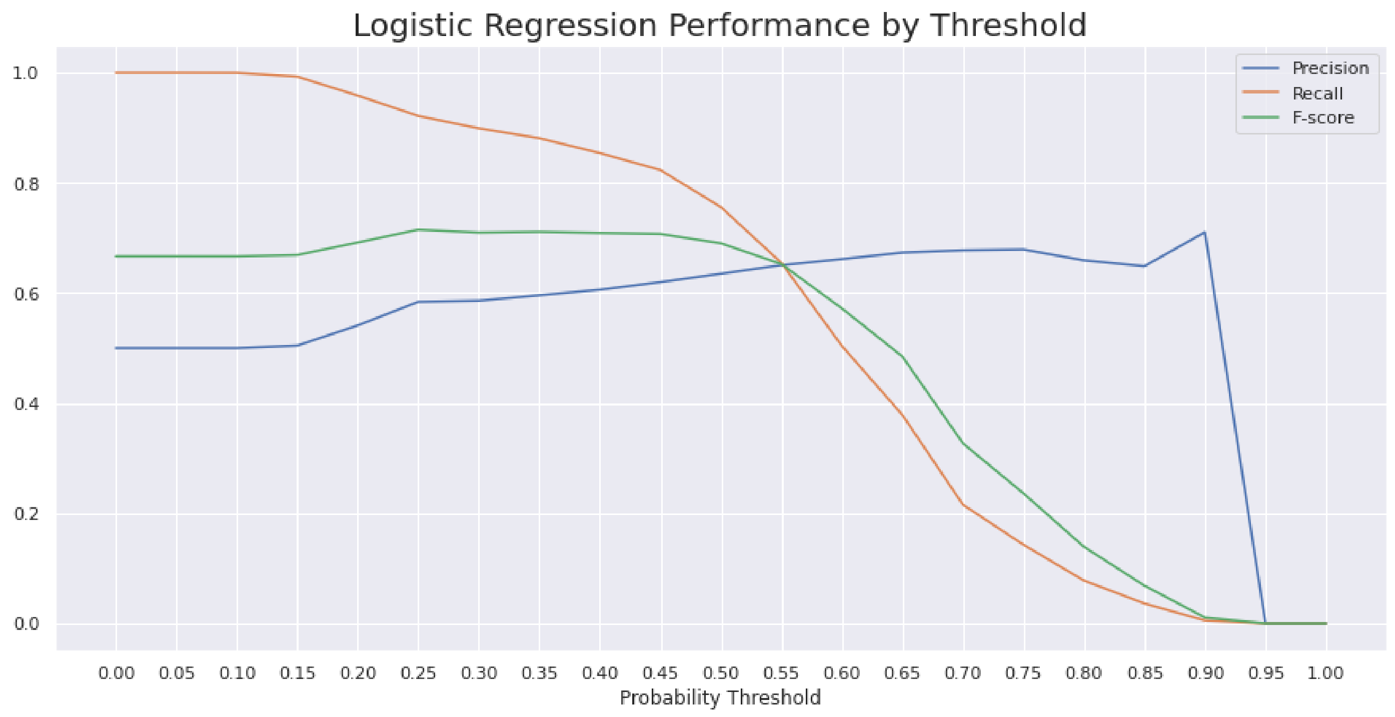
The recall seems to keep increasing, so I extended the search by selecting n from 500-5000 in increments of 500, shown below.



Recall is maximised around 1500 neighbours, while precision and the F-score are relatively stable. Therefore, I use 1500 neighbours for the final k-NN model and fit it on the whole data set.

Logistic Regression:

For the logistic regression, I optimise the probability threshold used for classification. I again look at precision, recall, and the F-score for probabilities from zero to one in increments of 0.05. The plot for these trials is shown below.



From this, I chose to use a cut-off of 0.25 as this maximised the F-score.

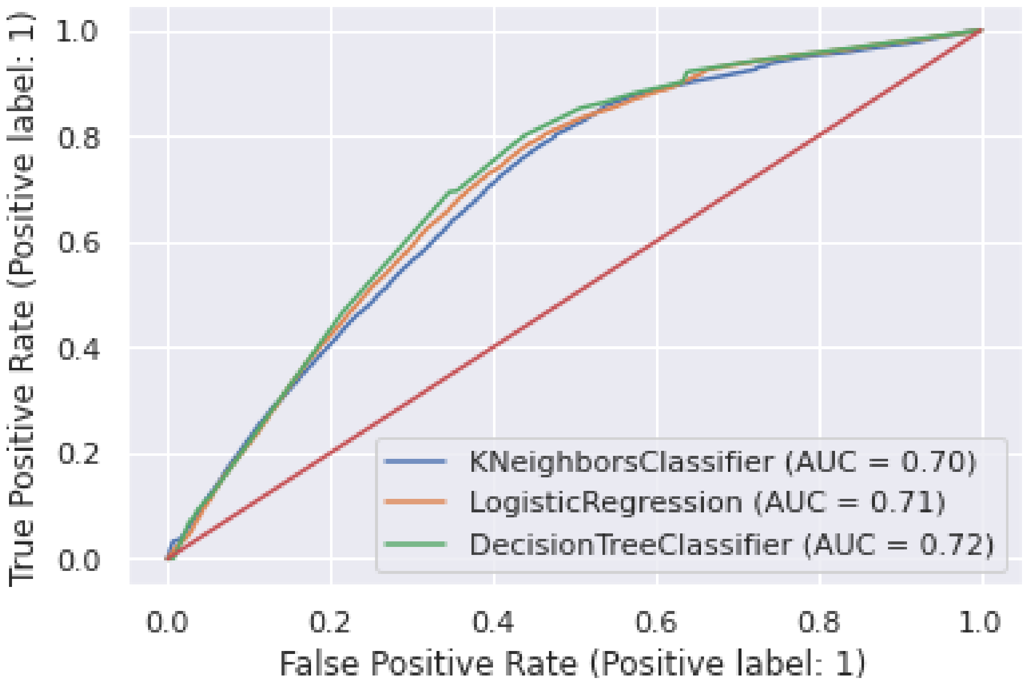
Decision Trees:

I used a randomised search to optimise a decision tree by randomising: criterion, maximum tree depth, minimum samples split, and maximum leaf nodes. I also used repeated stratified k-fold cross validation and compared results using the F-score of each tree. The final used the gini criterion, with a max depth of 13, max leaf nodes of 19, and min samples split of 171.

Model Performance

I compared the precision, recall, and F-scores of the models, before also comparing ROC curves and AUC scores. The decision tree has the best F-score, but only just, with a similar score to k-NN. The logistic regression is also not far away. All the models also have precision and recall between 0.6-0.7 which is not great but is an improvement over a trivial model which predicts zero and one with a 50:50 split.

The plot below shows the ROC curves and AUC scores for each of the models.



All three models have very similar ROC curves and AUC scores, with the decision tree being marginally the best. My suggestion would therefore be to use the decision tree as it performed the best of the models, but only just. Moreover, it might be worth using a random forest classifier, which combines multiple trees for classification, to improve model performance. Also, the use of Switch and PastAccident could improve performance if more advanced, reliable imputation methods were used.