YouNa Kim

Stat 724

Accurately Predicting the Car Insurance Claims

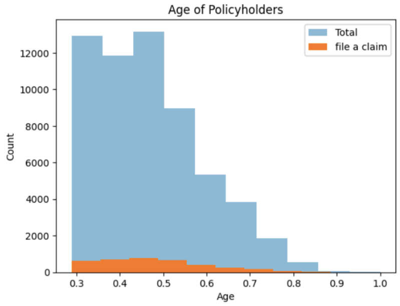
Introduction:

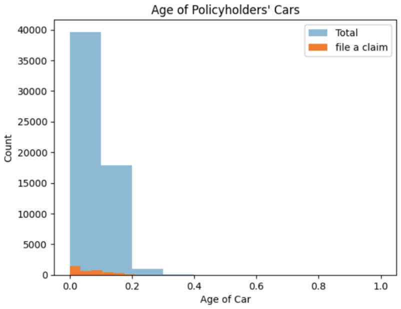
According to PolicyAdvice.net, about 105 auto accidents occur every day in the United States from the 2022 data, and more than 77,000 vehicles get stolen each year. In order to manage risks and calculate insurance rates, it is indispensable for auto insurance companies to predict whether their drivers will file a claim or not. The sample dataset is obtained from Kaggle's "Car Insurance Claim Prediction" to classify whether the policyholder will file a claim in the next 6 months or not using statistical learning methods in machine learning. There are several methods used to classify observations that include logistic regression, random forest, and KNN. The best method with the highest ROC AUC score (used instead of accuracy due to data imbalance) will be selected to aid the auto insurance company keep its profit loss at bay.

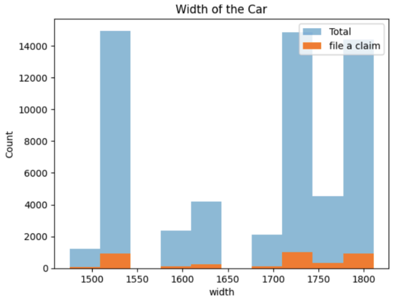
Data Description:

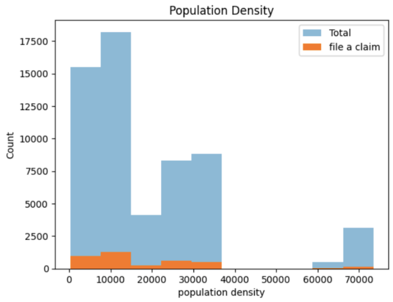
The source of Kaggle's "Car Insurance Claim Prediction" is from Data verse Hack, a data science platform that is based in India, and there is no indication on their website whether the data was collected from the real world. The dependent variable is a binary column called is_claim that is assigned 1 for policyholders who filed a claim in six months and 0 for those who did not. The independent variables are mainly categorical. One important note is that two columns, age_of_policyholder and age_of_car are given in normalized values which frustrate interpretation of the true ages. Additionally, the source does not elucidate what the categorical variables from area_cluster, make, or model columns are referring to. For instance, "1" from the make column could be referring to any makes from Honda to BMW to Ferrari. These variables can still nevertheless be used in the classification methods for the sole purpose of prediction with the given testing set data.

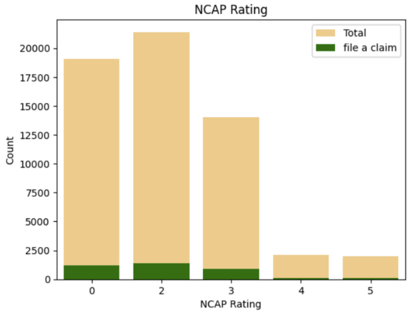
Without duplicated or missing values, it is easy to perform descriptive analysis. The percentage of policyholders who filed a claim in 6 months is 6.40% from the training dataset and some of the independent variables are graphed to observe the distributions of the population. The policyholders' and their cars' ages in the population are skewed right. Also, the policyholders who filed a claim are randomly distributed across different population density of their area and car width, and thus, it is expected that these variables do not change the likelihood of filling a claim. Moreover, approximately 90% of the policyholders have cars with low NCAP rating that are either 0, 2, or 3, which is rational considering that their car segments range from A to C. Segment C, the maximum sized cars from the data, have the size of a small sedan (length 4,300mm x width 1,811mm) and these cars have higher number of policyholders filing a claim. In addition, policyholders with car models M6, M1, and M4 and engine types K Series Dual Jet, F8D Petrol Engine, 1.5L U2 CRDi are also more prone to filing a claim. Lastly, the number of policyholders who filed a claim for 3 different engine types are consistent, indicating that engine types likely will not be useful in the classification method.

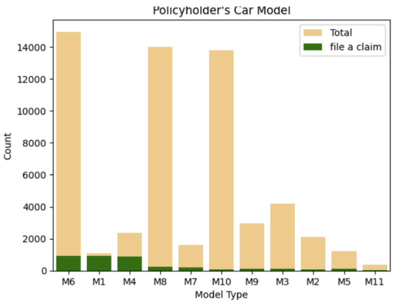


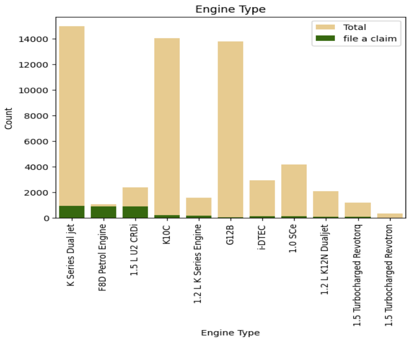


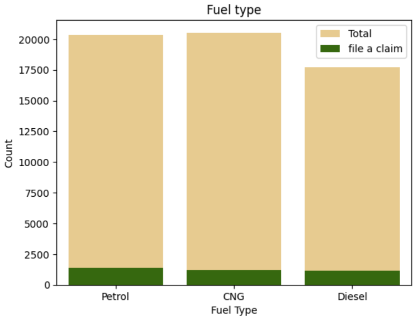








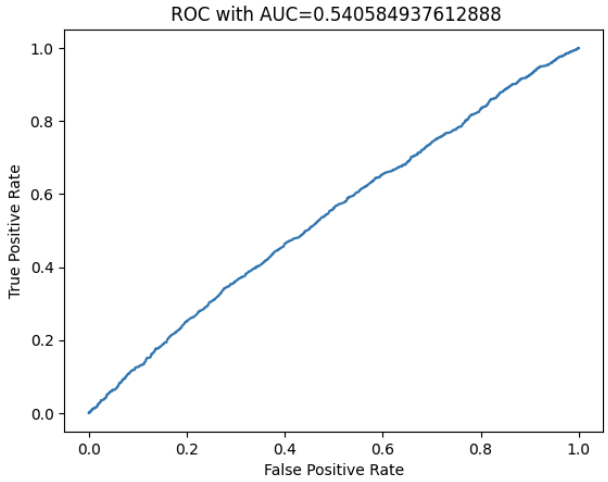


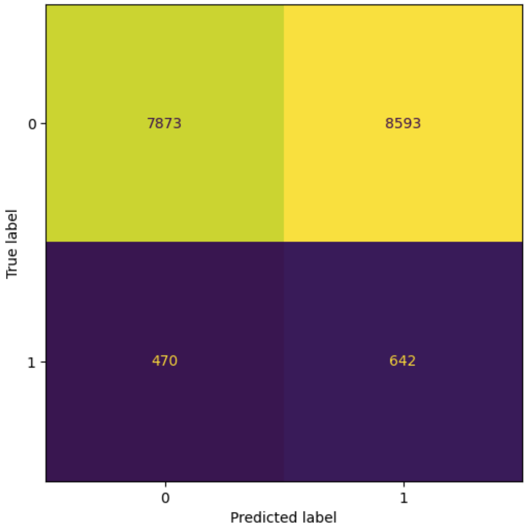


Analysis:

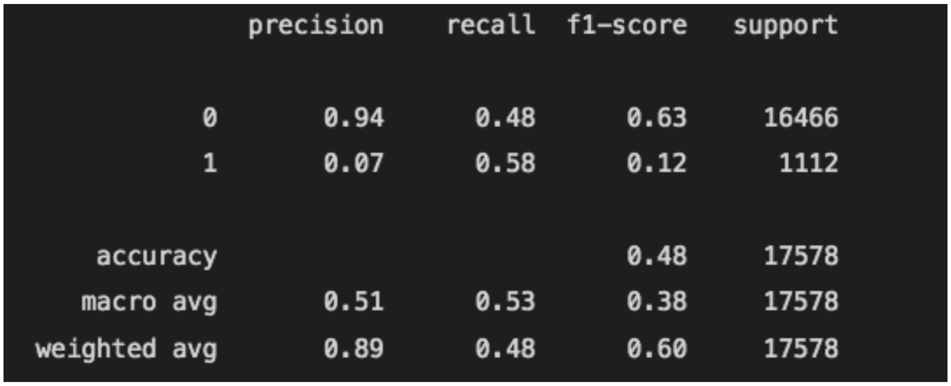
Logistic regression:

Logistic regression is a type of linear regression that predicts that a claim will be made within 6 months if the conditional probability of success is higher than the threshold (.5). The assumptions of no multicollinearity, continuous variables that are linearly related to the log odds, and binary dependent variable are checked during the process of building the model. Due to large numbers of independent variables, feature selection with Lasso Regularization is first used to reduce redundant slopes of predictors to zero. This method also helps avoid overfitting. The Lasso Logistic Regularization works by applying the penalty term, lambda times the sum of absolute value of coefficients, to the RSS, where a large value of lambda shrinks the coefficients to zero. During the process of building our model, the data is balanced by setting the class_weight= ’balanced’ and cross validation is utilized to find the hyperparameter that will deliver the highest AUC score. In sklearn, hyperparameter C represents 1/lambda. After the feature selection, the model’s AUC, the area under the ROC curve that determines how good of a classifier the model is, is .54 which insinuates that the model is slightly more competent at detecting policyholders who claim within 6 months than a random guess. Lastly, the average weighted recall is .48, which is low. Recall is more preferable than precision since we want to predict as many policyholders who will claim as possible.





	0	1
0	7873	8593
1	470	642

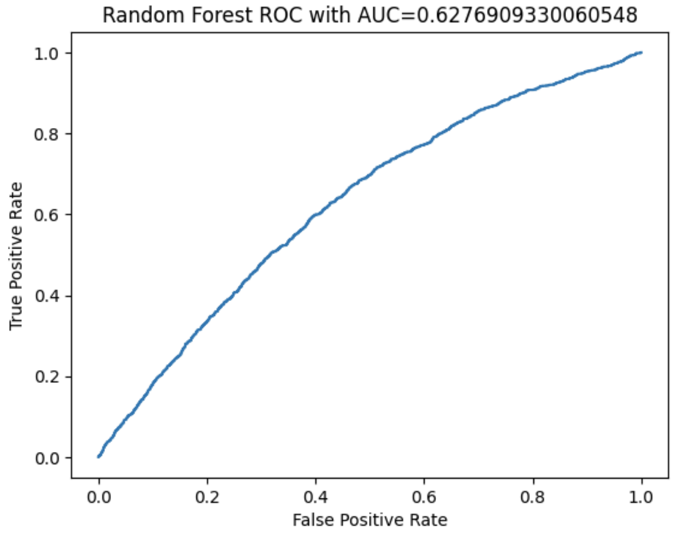


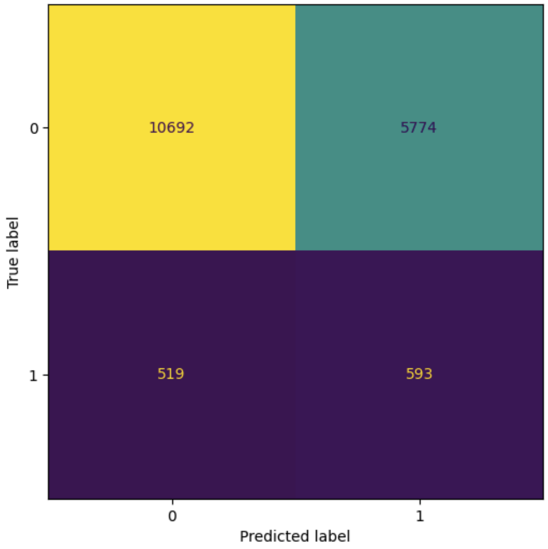
```
              precision    recall  f1-score   support

           0       0.94      0.48      0.63     16466
           1       0.07      0.58      0.12      1112

    accuracy                           0.48     17578
   macro avg       0.51      0.53      0.38     17578
weighted avg       0.89      0.48      0.60     17578
```

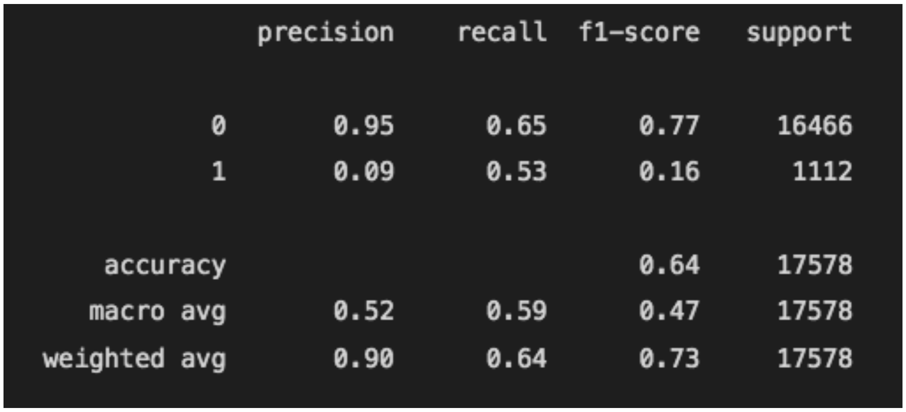
Random Forest:

A Random Forest of many trees are built from bootstrapped training dataset and one of random m predictors are chosen at each split to be the node. The purpose of using a subset of predictors is to prevent choosing the most influential predictor continuously. The predictor with the lowest impurity is chosen to be the node. In Random Forest, all trees get equal votes that are used to determine the classification outcome. During the process of building Random Forest, imbalanced data is dealt by using class_weight= 'balanced' and cross validation is used to determine the best hyperparameters for maximum node expansion, number of trees built, amount m features, and minimum samples in the leaf before expanding the tree.

Before using the selected features, the AUC of Random Forest Classifier is .63 and its average weighted recall is .64.

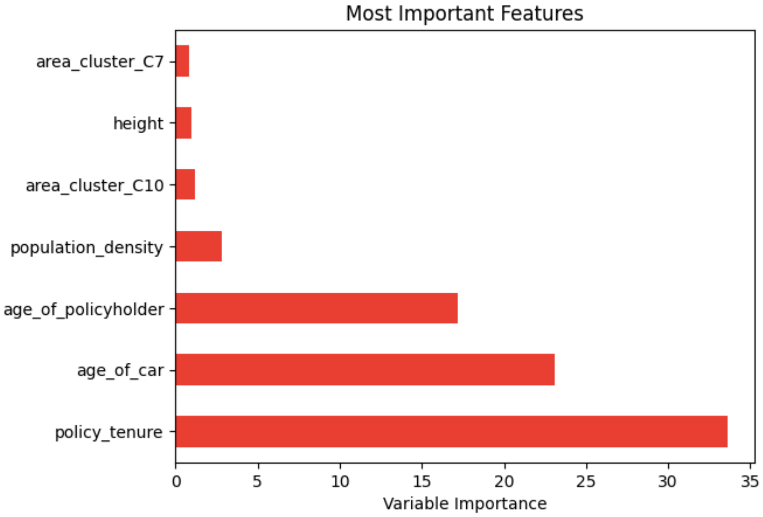




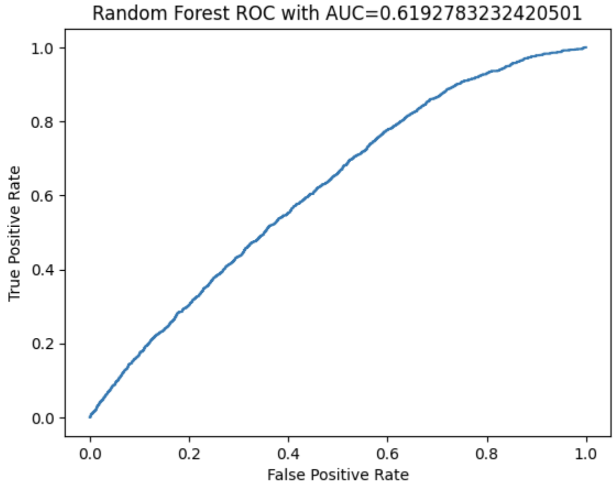


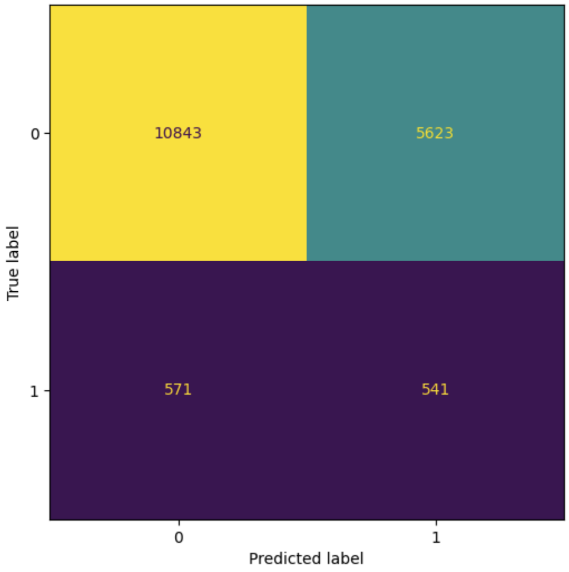
```
              precision    recall  f1-score   support

           0       0.95      0.65      0.77     16466
           1       0.09      0.53      0.16      1112

    accuracy                           0.64     17578
   macro avg       0.52      0.59      0.47     17578
weighted avg       0.90      0.64      0.73     17578
```

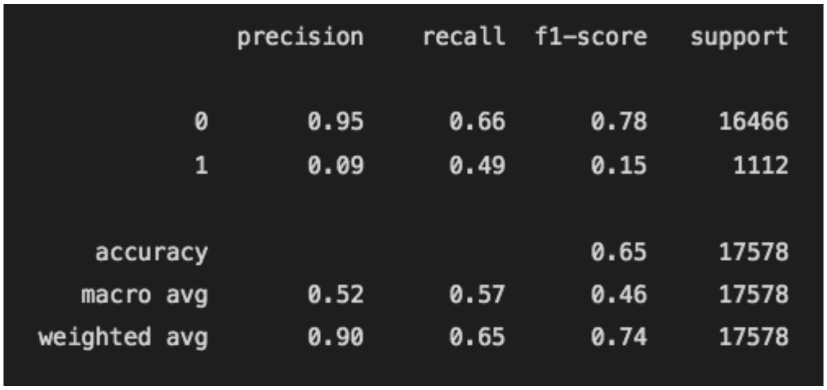
The features used which tree-based model selected as important are listed:



After the feature selection, there is not much improvement from the previous model as AUC slightly decreased to .62, while weighted average recall increased only slightly by .01. Based on lack of improvement, the above model where all features are fitted can be used instead.


Random Forest ROC with AUC=0.6192783232420501

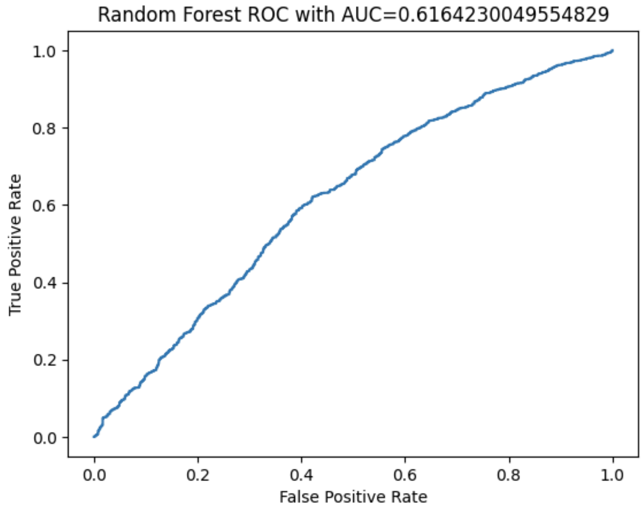


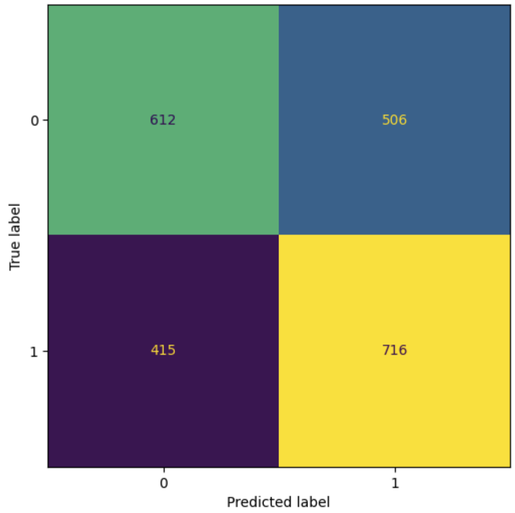


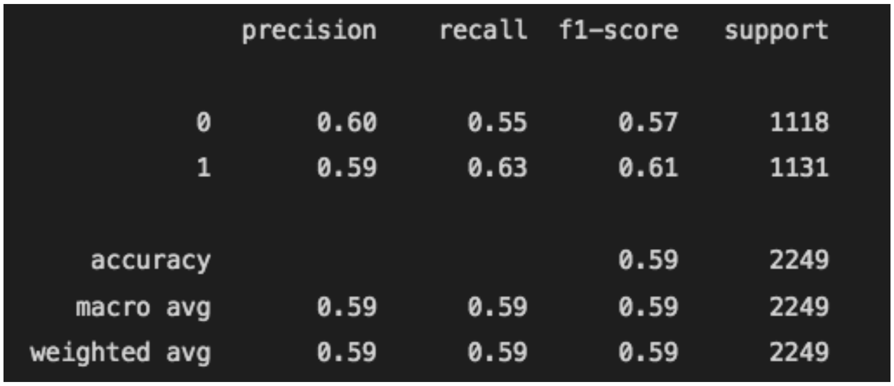
```
              precision    recall  f1-score   support

           0       0.95      0.66      0.78     16466
           1       0.09      0.49      0.15      1112

    accuracy                           0.65     17578
   macro avg       0.52      0.57      0.46     17578
weighted avg       0.90      0.65      0.74     17578
```

K Nearest Neighbors:

The K-Nearest Neighbors is a non-parametric method that can deal with any distribution or shape of data. KNN searches through all data points and calculates the distance. Thus, it is important to normalize the points first and determine the right value for k, the number of nearest points, for accurate result. The ideal starting value for k is square root of sample size and cross validation can be used to find the k with the best score. The KNN classifier will then takes the average response for k points and classify the average based on the threshold. One disadvantage of using KNN is that this method cannot select features. Thus, the important features selected from Random Forest are used. Moreover, since KNN algorithm does not have a quick method to balance data using class_weight= ’balanced’, one type of resampling method called under sampling is applied. It under samples the rows where is_claim=0 to match the number of rows where is_claim=1. After performing the KNN, the AUC is .62 and the weighted average of the recall is .59.







```
              precision    recall  f1-score   support

           0       0.60      0.55      0.57      1118
           1       0.59      0.63      0.61      1131

    accuracy                           0.59      2249
   macro avg       0.59      0.59      0.59      2249
weighted avg       0.59      0.59      0.59      2249
```

Conclusion:

In conclusion, all three methods are compared using AUC score and recall instead of accuracy because only 6.4% of the data make up the minority class where is_claim=1. The reason for choosing to focus on recall instead of precision is that from the insurance companies' perspective, they would prefer to estimate the worst case of financial loss possible. This situation is known as precision recall tradeoff, where either one can be high but not both. Of the three models that are graphed, logistic regression has the lowest performance of all (recall=.48, AUC=.54) and KNN has the second-best performance (recall=.59, AUC=.61). Lastly, Random Forest has the best performance (recall=.64, AUC=.63) and is used to classify policyholders found on new data. Using the Random Forest model, 14,350 out of 39,063 policyholders (36.7%) are expected to file a claim within the next 6 months.

Although scores of all three models are lower than expected, several methods that could enhance the outcome include boosting logistic regression, trying different cross validation methods like stratified, and trying different thresholds besides .5. Additionally, the insurance company could include more information on each policyholder’s driving record, credit scores (albeit this is banned in states like California and Massachusetts), and past auto insurance history for better classification of policyholders to calculate suitable premiums.

Work Cited

Najnin, Iftesha. “Car Insurance Claim Prediction.” *Kaggle*, 14 Nov. 2022, https://www.kaggle.com/datasets/ifteshanajnin/carinsuranceclaimprediction-classification?select=train.csv.

Statsha, Smiljanic. “Notable Auto Insurance Statistics for 2021: Policy Advice.” *Notable Auto Insurance Statistics for 2021 | Policy Advice | Policy Advice*, 2022, https://policyadvice.net/insurance/insights/auto-insurance-statistics/.