Assignment 2:
Car Insurance Claim Prediction

Jiayang Bao (CSE 158R)
Jiayi Zhao (CSE 158R)
Yinglan Chi (CSE 258)

Literature Review

Insurance companies have put great emphasis on predicting their customers' behaviors such as the number of potential insurance claims they will file. An accurate forecasting model can provide insurance companies the opportunity to optimize their financial gains and minimize their financial losses. According to the Insurance Information Institute report, the claims frequency, amount of claims per car, and claim severity have increased by 35% between 2010 and 2019 for US auto insurance, and the average US car insurance expenses increased from $78,665 to $100,458 in 8 years.[1] This increase indicates the need for a high-accuracy prediction model in predicting auto insurance claims under the increasing claim frequency conditions. Moreover, these increasing needs make our auto insurance claims prediction studies more meaningful. Many insurance companies have long recognized the power of artificial intelligence and machine learning to help companies optimize services more precisely. For instance, Allstate has started a Claim Prediction Challenge with a winner prize of up to $10,000, and the goal of this competition is to better predict Bodily Injury Liability Insurance claim payments based on the characteristics of the insured customer's vehicle.[2]

Similar to this challenge, we want to predict whether the policyholder will file a claim in the next 6 months or not. We aim to create an effective approach and a reliable ML model to assess different features such as policy tenure, age of the car, age of the car owner, the population density of the city, make and model of the car, power, engine type, etc. We want to predict the probability of filing a claim in the coming 6 months, and create a model that can interpret vast databases containing more than fifty thousand consumer details provided by Analytics Vidya DataVerse Hack Competition. This dataset is also available on Kaggle.[3] Similar datasets have been studied in the past. The paper Machine Learning Approaches for Auto Insurance Big Data by Hanafy, Mohamed, and Ruixing Ming analyzed insurance customers' metadata provided by the Porto Seguro insurance company.[5] According to this paper, Porto Seguro is one of the biggest car insurance firms in Brazil. The dataset they used contained 59 variables with 1,488,028 observations.[5] In their paper, they used regression analysis, Decision Tree, XGBoost, Random Forest, K-Nearest Neighbor, and Naïve Bayes to create their model. Then, they used the Confusion Matrix and Kappa Statistics to do the model evaluation. The random forest model has been concluded with the highest accuracy model.[5] The regression analysis, Random Forest, and KNN methods are the state-or-art methods currently employed to study this type of data, so we also decided to use the above model in our prediction. It turns out that the conclusions from this paper are highly similar to our own findings, in which the Random Forest model has the highest accuracy.

Dataset & Exploratory Analysis

By introducing the Car Insurance Claim Prediction dataset on Kaggle[3], we utilized its training dataset (train.csv) to study since its other datasets like the testing dataset (test.csv) and sample submission dataset (sample_submission.csv) lack information for us to test the future model prediction accuracy. There are 58592 rows in the dataset which is reasonably large enough for us to run the kinds of methods we use for our model. Each row of the dataset contains one year's worth of information for insured vehicles having 43 features (28 categorical features and 15 numerical features) like policy tenure, age of the car, age of the car owner, the population density of the city, make and model of the car, power, engine type, etc, and the target variable

indicating whether the policyholder files a claim in the next 6 months or not.

By performing an exploratory analysis of the data, we found that the distribution of the age of policyholders who files a claim (Figure 1) is skewed to the right which means that younger policyholders are more common to file a claim. The distribution of the age of the car from the claimed policyholder (Figure 2) is also right-skewed, which represents that policyholders who own newer cars are more likely to file a claim.

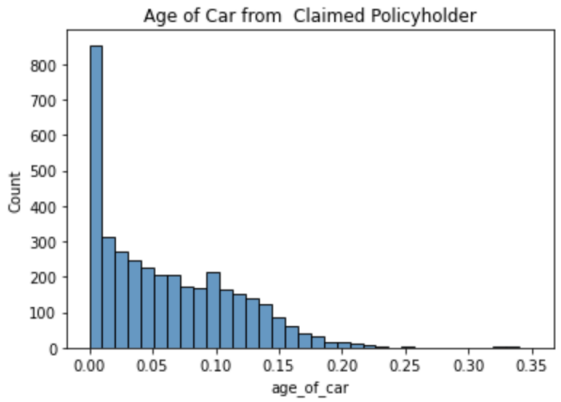

Figure 1. Distribution of Age of Car
From Policyholders Who File A Claim

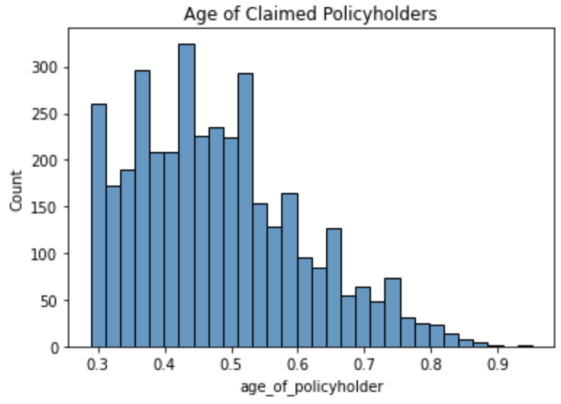

Figure 2. Distribution of Age of Policyholders Who File A Claim

In addition, we saw that policyholders are mostly from areas of lower population density by the following distribution graph (Figure 3).

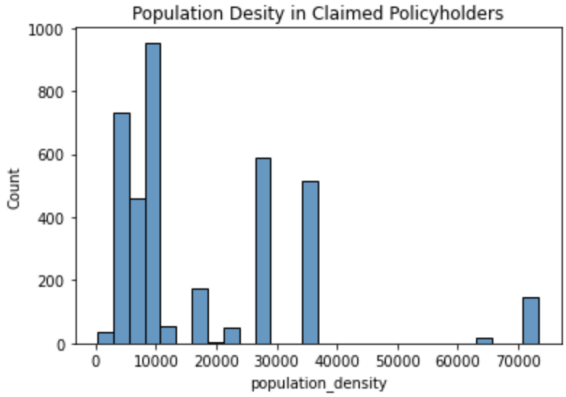

Figure 3. Distribution of Population Density in the City of Policyholders Who File A Claim

The following graph (Figure 4) shows the distribution of whether or not policyholders file a claim. Interestingly, the frequency of not filing a claim is much higher than the frequency of filing a claim, which might cause an unequal distribution of classes when we split the data into a training set and a testing set and resulting in models that have poor predictive performance, specifically for the minority class.

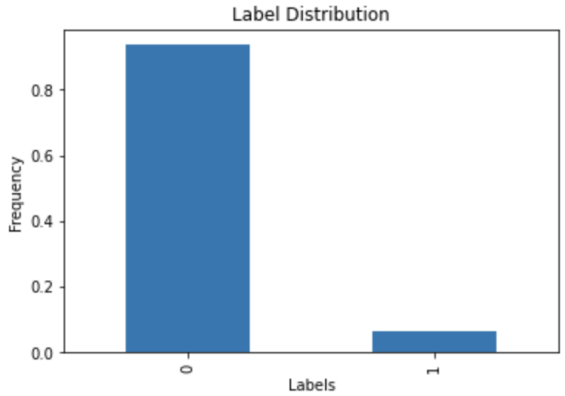

Figure 4. Distribution of Whether Policyholders File A Claim

Predictive Task

The predictive task that we studied for this dataset is predicting whether the policyholder will file a claim in the next 6 months or not.

Evaluation Methods

In the context of our dataset, we predicted the "is_claim" feature. To evaluate the performance of our different models on the predictive task, we splitted our dataset into a train set and a test set; specifically, we divided our dataset into an

80% train set and a 20% test set. We trained our models individually on the train set and computed the accuracy and confusion matrix for each.

We applied the confusion matrix, which gave us a matrix as output and described the complete performance of the model. In the confusion matrix, every row is the ground truth and every column is the prediction. We calculated the recall and precision value to evaluate the relationship between the true is_claim output and predicted output. Recall is the measure for how many true is_claim outputs get predicted out of all the is_claim outputs in the dataset while precision is the measure of the correctness of an is_claim output prediction.

Feature Selection

There are a total of 43 features in the dataset, including attributes about the cars, attributes about the policyholders, and attributes about the location that the policyholders are in. Among the 43 features, we dropped the features "policy_id" and "area_cluster" since"policy_id" does not have any cause-and-effect relationships with whether a policyholder will file a claim within the next 6 months and the dataset contains the feature "population_density" in our dataset, which is more informational than the "area_cluster" feature. Therefore, we trained our models with the following 41 features (Figure 5).

Variable	Description
policy_tenure	Time period of the policy
age_of_car	Normalized age of the car in years
age_of_policyholder	Normalized age of policyholder in years
area_cluster	Area cluster of the policyholder
population density	Population density of the city (Policyholder City)
make	Encoded Manufacturer/company of the car
segment	Sefment of the car(A/B1/B2/C1/C2)
model	Encoded name of the car
fuel type	Type of fuel used by the car
max_torque	Maximum Torque generated by the car (Nm@rpm)
max_power	Maximum Torque generated by the car (Nm@rpm)
engine_type	Type of engine used in the car
airbags	Number of airbags installed in the car
is_esc	Boolean flag indicating whether Electronic Stability Control (ESC) is present in the car or not.
is_adjustable_steering	Boolean flag indicating whether the steering wheel of the car is adjustable or not.
is_tpms	Boolean flag indicating whether Tyre Pressure Monitoring System (TPMS) is present in the car or not.
is_parking_sensors	Boolean flag indicating whether parking sensors are present in the car or not.
is_parking_camera	Boolean flag indicating whether the parking camera is present in the car or not.
rear_brakes_type	Type of brakes used in the rear of the car
displacement	Type of brakes used in the rear of the car
cylinder	Number of cylinders present in the engine of the car
transmission_type	Transmission type of the car
gear_box	Number of gears in the car
steering_type	Type of the power steering present in the car
turning_radius	The space a vehicle needs to make a certain turn (Meters)
length	Length of the car (Millimetre)
width	width of the car (Millimetre)
height	Height of the car (Millimetre)
gross_weight	The maximum allowable weight of the fully-loaded car, including passengers, cargo and equipment (Kg)
is_front_fog_lights	Boolean flag indicating whether front fog lights are available in the car or not.
is_rear_window_wiper	Boolean flag indicating whether the rear window wiper is available in the car or not.
is_rear_window_washer	Boolean flag indicating whether the rear window washer is available in the car or not.
is_rear_window_defogger	Boolean flag indicating whether rear window defogger is available in the car or not.
is_brake_assist	Boolean flag indicating whether the brake assistance feature is available in the car or not.
is_power_door_lock	Boolean flag indicating whether a power door lock is available in the car or not.
is_central_locking	Boolean flag indicating whether the central locking feature is available in the car or not.
is_power_steering	Boolean flag indicating whether power steering is available in the car or not.
is_driver_seat_height_adjustable	Boolean flag indicating whether the height of the driver seat is adjustable or not.
is_day_night_rear_view_mirror	Boolean flag indicating whether day & night rearview mirror is present in the car or not.
is_ecw	Boolean flag indicating whether Engine Check Warning (ECW) is available in the car or not
is_speed_alert	Boolean flag indicating whether the speed alert system is available in the car or not.
ncap_rating	Safety rating given by NCAP (out of 5)

Figure 5. 41 Model Features with Description

We used the numeric features directly from the dataset, and we executed the LabelEncoder function to encode the categorical features as numerical labels before using them.

For the labels, we discovered a problem of label imbalances as shown in the exploratory analysis, in which the number of negative labels far exceeds the number of positive labels. This means that even if we have a bad model that predicts all labels as negative, we can still achieve high accuracy. To solve this problem, we utilized the RandomOverSampler function from the imblearn library on our labels. As a result, we were able to make our label distribution more evenly, roughly about 50% positive and 50% negative.

Baseline Model

For the model construction, we first constructed a simple baseline model with features "age_of_car", "population_density", and "age_of_policyholder". We chose these three features based on the results we got from the exploratory data analysis. From the EDA, we discovered that the lower the population density, age of cars, and age of policyholders, the higher the number of claimed policyholders. We took the median of the age_of_cars feature, population_density feature, and age_of_policyholder feature as our baseline. All policies with age_of_car, population_density, and age_of_policyholder greater than our baseline were predicted as is_claim = 1. This baseline model is somewhat similar to the read prediction baseline that we discussed in the lecture and assignment1: this baseline model is predicting the claims with population density,

age of cars, and age of policyholders less than the median of all claims as filed, and the read prediction baseline model was predicting all books with popularity greater than the median of all popularity counts as read. After constructing the baseline model, we constructed a logistic regression model, K-NN model, and random forest model and compared them with the baseline. The resulting accuracy score for the baseline model is 81.85%. Based on the confusion matrix, we got a precision score of 0.93 and a recall score of 0.87 for the baseline model.

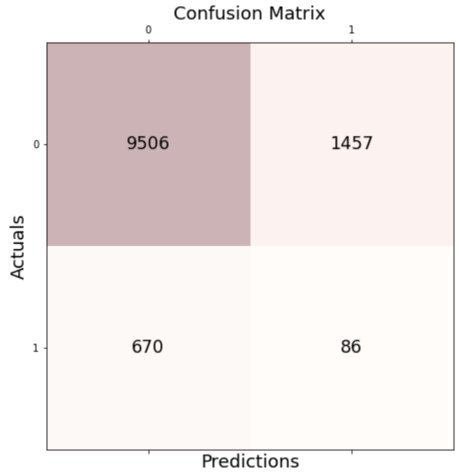

Figure 6. Confusion Matrix for Baseline Model

Models

In order to optimize the baseline model, we employed 3 different machine learning models: logistic regression, K-NN, and random forest. The deliberate decision of choosing these 3 models stems from the fact that the dataset contains numeric variables and a binary output variable.

Logistic Regression

The statistical analysis technique known as logistic regression is applied when our dependent variable is binary or dichotomous. Considering the binary nature of output values, logistic regression should be a suitable classification for our large dataset with its advantage in speed and it fulfills our requirement of predicting binary purchase decisions with multiple features. Compared to other models, one of the disadvantages of the logistic regression model is easier overfitting the training data since the dataset contains a large number of features. On the other hand, logistic regression is easier and faster to implement.

We implemented Pipeline and GridSearchCV to search for the optimal hyperparameters that could generate the best performance of the logistic regression classifier over our dataset. We explore those hyperparameters among different solvers, different penalties, and different C values. The metric we are using in our hyperparameters selection process is accuracy. In order to make the best use of our dataset and get an unbiased accuracy while comparing different hyperparameters, we are comparing them with the average validation set accuracy from k-fold cross-validation. The logistic regression model with the highest accuracy from grid search contains the parameters with 0.1 C value, ‘sag’ solver with l2 penalty. However, the accuracy of the optimal logistic regression model is only 56.57% which is worse than what we got from the baseline model. The precision and recall score for the logistic regression model are 0.95 and 0.57. The low value of the recall score might explain why the logistic regression model has low accuracy.

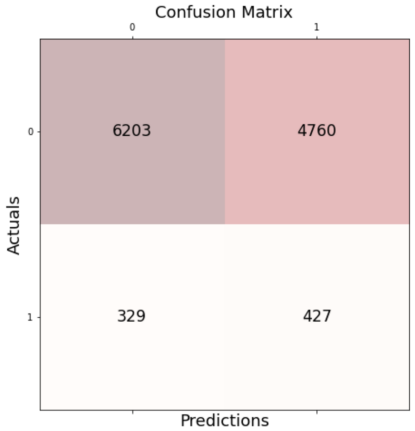

Figure 7. Confusion Matrix for Logistic Regression Model

K-NN model

The supervised learning-based K-nearest neighbor model contrasts the previously unexplored data with the newly discovered data. The category that is closest to the available classes includes the new data. We intended to use this model for classification. K-NN is one of the state-of-art methods in dealing with

insurance claims datasets according to the previous literature review, so we decided to implement this model as well. The strength of the K-NN algorithm is that it doesn't require training time. K-NN does not build any model. The classifier immediately adapts as we collect new training data. It allows the algorithm to respond quickly to changes in the input during real-time use. Also, it's very simple and easy to use since it simply chooses the neighbors based on distance criteria. Although the K-NN algorithm is straightforward, it's hard to determine the right k (number of neighbors) and different k can have a significant impact on the model accuracy and whether the model is underfitting or overfitting. Thus, one challenge we had for the K-NN model is to select the best hyperparameters. We decided to do a grid search on different hyperparameters and selected the best model from the grid and evaluated the best model on the test set. We decided to tune the K-NN model with neighbors ranging from 1 to 10, and then checked the accuracy for different neighboring values and found the one that produces the best accuracy. A confusion matrix check (Figure 8) had also been performed for this model to evaluate the performance of the classification models. The precision for the K-NN model is 0.94 and the recall is 0.94. The final accuracy score for the optimized K-NN model is 88.26%, which is a significant improvement compared to the baseline model.

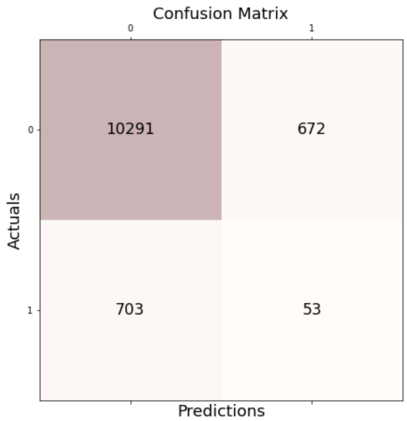

Figure 8. Confusion Matrix for K-NN Model

Random Forest Model

The random forest classifier consists of multiple decision trees that will execute as an ensemble. For decision trees, we are organizing the dataset into a tree structure which means that we are creating a node in the decision trees when we ask a question. The result of the question is either 1 or 0, i.e. yes or no, and the result will help us split our dataset into two subsets. While the node connecting to the result no will be a leaf node in the tree, the node connecting to the result yes can be split further. We stop splitting until no leaves can be split further. After constructing the tree, we assign a class to each leaf node. The reason that we chose to use random forests is for two main reasons. Firstly, from our exploratory data analysis and the baseline model, we already knew that there exists several features in the dataset that have at least some predictive power. Secondly, we have a lot of features in the dataset, and using the random forest classifier can help us select the ones that are most useful for prediction and the ones that do not have a strong correlation with the label will be disregarded by the model.

The strength of the random forest classifier when compared with K-NN and logistic regression is that it performs better for large datasets with higher dimensionality, which suits our dataset. Another advantage against logistic regression is that the random forest model does not get overfitted easily. The main disadvantage of the random forest model against the other two models is that it consumes more time and resources for computation. However, in this scenario, the time and resource consumption are tolerable.

To optimize the random forest model, we used the GridSearchCV, a cross-validation method, to help tune the hyperparameters. The accuracy for the optimized model on the test set is around 92.03%, which is the highest among all models that we've tried. The precision is approximately 0.94, and the recall is approximately 0.98, which is the highest among all models.

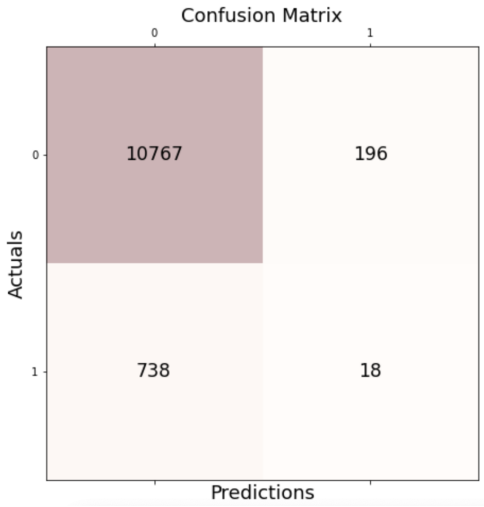

Figure 9. Confusion Matrix for Random Forest Model

Results & Discussion

In our study, we show that data re-sampling is essential in handling an imbalanced dataset to prevent bias prediction. We also build different machine learning models to predict the likelihood of filing a claim using different models and metrics. The result of our analysis shows that insurance claims can be predicted using machine learning methods with a high-accuracy model. Also, according to the results from figure 10, we can see that Random Forest is the best model with an accuracy of 92%. Compared with other alternatives, such as logistic regression and K-NN, the Random Forest model shows higher reliability in classifying and predicting.

====== Model Summary ======

	Algorithm	Accurracy
0	Baseline Model	0.818500
1	Logistic Regression	0.565748
2	KNN	0.882669
3	Random Forest	0.920300

Figure 10. Model Accuracy Summary Table

Since Random Forest is the most accurate model for predicting policyholders filing a claim, we explored the feature weights on the random forest model. According to figure 11, we see the population density of the city of policyholders and the age of car are the two most weighted features in the random forest model. They negatively correlated with the policyholder's decision which means that the less the population density of the city that policyholder is in and newer the car is, the higher probability the policyholder will file a claim. In addition, the feature of the age of the policyholder which is also one of the standards in our baseline model also ranks 19 in the weight list. On the other hand, the features of whether power steering is available in the car or not and whether the speed alert system is available in the car or not are the two least weighted features in the random forest model, which means that the existence of power steering and speed alert system in the car hardly affect the decision of the policyholder to file a claim for their vehicle (Figure 12).

Weight	Feature
-0.0031 ± 0.0022	population_density
-0.0030 ± 0.0019	age_of_car
-0.0029 ± 0.0004	model
-0.0027 ± 0.0004	height
-0.0027 ± 0.0007	displacement
-0.0026 ± 0.0004	length
-0.0025 ± 0.0004	width
-0.0024 ± 0.0007	segment
-0.0023 ± 0.0005	gross_weight
-0.0022 ± 0.0007	max_torque
-0.0019 ± 0.0002	is_adjustable_steering
-0.0019 ± 0.0002	cylinder
-0.0017 ± 0.0006	max_power
-0.0017 ± 0.0003	turning_radius
-0.0013 ± 0.0004	engine_type
-0.0013 ± 0.0003	ncap_rating
-0.0011 ± 0.0003	fuel_type
-0.0010 ± 0.0002	steering_type
-0.0009 ± 0.0020	age_of_policyholder

Figure 11. Most Weighted Features For Random Forest Model Top 19

Weight	Feature
0 ± 0.0000	is_power_steering
-0.0000 ± 0.0001	is_speed_alert
-0.0000 ± 0.0014	policy_tenure
-0.0000 ± 0.0001	is_parking_sensors
-0.0002 ± 0.0001	rear_brakes_type
-0.0002 ± 0.0002	is_rear_window_wiper
-0.0002 ± 0.0002	is_tpms
-0.0002 ± 0.0001	is_rear_window_washer
-0.0002 ± 0.0001	transmission_type
-0.0003 ± 0.0003	is_day_night_rear_view_mirror

Figure 12. Least Weighted Features For Random Forest Model Top 10

The random forest succeeds because, as mentioned in the model section above, this model suits our dataset more than the logistic regression model and K-NN model since our dataset is large and contains a lot of different features. Moreover, our dataset contains a lot of categorical features, which the random forest model works very well with. The reason that the logistic regression model does not work very well is that it tends to overfit when we have too many features in the dataset. K-NN models perform fairly well, but it outputs a lower accuracy value compared to the random forest because K-NN models tend to perform better on a large dataset with lower dimensionality.

In the future, we should implement the random forest model in different insurance claims datasets to see if the model still works well. It's also essential to constantly improve the model with new training data and testing data.

References

[1]"Facts+Statistics:Auto Insurance." *III*, https://www.iii.org/fact-statistic/facts-statistics-auto-insurance.

[2] "Allstate Claim Prediction Challenge." *Kaggle*, https://www.kaggle.com/competitions/ClaimPredictionChallenge.

[3] Najnin, Iftesha. "Car Insurance Claim Prediction." *Kaggle*, 14 Nov. 2022, https://www.kaggle.com/datasets/ifteshanajnin/carinsuranceclaimprediction-classification.

[4] Hanafy, Mohamed, and Ruixing Ming. 2021. "Machine Learning Approaches for Auto Insurance Big Data" Risks 9, no. 2: 42. https://doi.org/10.3390/risks9020042

[5]Cristini, Aline. "Atividade2-PortoSeguro." *Kaggle*, 18 Feb. 2019, https://www.kaggle.com/datasets/alinecristini/atividade2portoseguro.