Evaluating the Impact of Feature Engineering on Auto Insurance Claim Prediction Models

Wenyi Fang

Department of Mathematics and Statistics, McMaster University, Hamilton, Ontario, L8S 4K1, Canada
fangw10@mcmaster.ca

Abstract. Predicting automobile insurance claim is essential for risk assessment, premium calculating, and fraud detection. All of which help ensure the profitability and stability of insurance companies. This study introduced a novel engineered feature, the Collision Index, designed to quantify localized automobile collision risk, which is aggregated into an insurance claim dataset to evaluate its impact on model performance. Four machine learning models, Random Forest, Extreme Gradient Boosting (XGBoost), Support Vector Machine and Neural Network, were trained using the dataset with and without the engineered feature. Their F1 scores are compared using the paired Student's t-test. Results indicate that only the Random Forest model witnessed a statistically significant improvement with the inclusion of the collision index at 0.05 significance level. The other models fail to observe significant performance gain, or imply a decisive conclusion due to computational constraints. Limitations of this paper include the imbalanced dataset, the estimated nature of collision index and possible incorrect assumption made during paired t-test.

Keywords: Insurance Claim Prediction, Feature Engineering, Machine Learning Models.

1 Introduction

Insurance companies play an important role in the economy, not only by protecting clients from financial loss due to accidents, but also by investing capital and spurring broader economic growth. To remain profitable, insurers must build strong investment portfolios, improve fraud detection, and make accurate claim predictions [1, 2].

Higher premiums are typically charged to high-risk drivers, who are more likely to file claims, while lower-risk drivers receive reduced premium for insurance companies to stay competitive. Since this practise is common across the insurance industry, it is essential to make accurate and precise predictions of claims. This task can be framed as supervised machine learning problems with a binary outcome (claim or no claim).

Prior studies have explored various approaches for predictions in the auto insurance industry. Saikia et al. identified that driving experience, credit score and age strongly influence claim prediction, while cautioning about biases from historical data [2]. Hanafy and Ming evaluated 13 machine learning models on auto insurance fraud data

A. J. Moshayedi (ed.), *Proceedings of the 2025 International Conference on Hybrid Commerce, Human Capital, and Economic Dynamics (ICHCH 2025)*, Advances in Economics, Business and Management Research 374,
https://doi.org/10.2991/978-2-38476-585-0_17

and found Stochastic Gradient Boosting had the best performance under resampling technique [1]. Moonoo and Hosein proposed a composite approach to predict claim severity, with Linear Regression witnessed the lowest Mean Squared Error but a more complex model [3]. Jonkheijm notified XGBoost Regressor achieved the lowest Mean Absolute Error across experiments though feature engineering was limited, and performance gains were modest [4].

This paper uses a Kaggle car insurance dataset to evaluate the impact of an engineered feature, the Collision Index, on model performance. The feature is designed to capture hidden localized patterns linked to postal codes and aims to enhance prediction performance. Four models, Random Forest (RF), Neural Network (NN), XGBoost (XGB) and Support Vector Machine (SVM), are assessed by comparing their F1 scores with and without the feature. Each model is trained 10 times with different random seeds, followed by a paired Student's t-Test (paired t-Test) for evaluating the significance of improvement. Results offer insights into establishing accurate, efficient models with meaningful feature engineering.

2 Methodology

2.1 Machine Learning Algorithms

Random Forest. RF is an ensemble algorithm that combines many weak learners (Decision Tree) using a method like bagging to create a powerful model. Each tree is trained on a randomly sampled partition of the training set [5]. Samples in the test set are passed through all trees, each of which makes a decision. The final prediction is determined by aggregating these decisions through voting, improving overall model accuracy. RF is known for its robustness to noise and outliers, effectiveness in capturing complex variable relationships, and high computational cost. It also offers strong adaptability through flexible hyperparameter tuning, helping to control overfitting or underfitting.

Neural Network. A one-layer neural network processes input variables by applying a linear transformation, where each input is multiplied by a corresponding weight and combined with a bias. The result is then passed through a nonlinear activation function, typically a ReLU function, to produce the final output. In this project, a neural network with two hidden layers was used, containing five neutrons in the first layer and two in the second. The ReLU activation function was applied in the hidden layers to enable the model to capture nonlinear relationships between inputs and outputs [6].

Support Vector Machine. SVM is a supervised algorithm that recognizes patterns in input data and finds the optimal hyperplane to separate classes. It maximizes the margin between classes and performs well with linearly separable data. For non-linearly separable cases, kernel functions map data to higher-dimensional space for linear separation [7]. SVM is favored for high-dimensional data due to its strong generalization when properly tuned.

Extreme Gradient Boosting. XGB is a supervised learning model that combines tree-based models with boosting techniques for strong predictive performance. It builds decision trees sequentially, where each new tree corrects errors from the previous ones and captures difficult cases [5]. A loss function measures how closely predictions match true values, and the model optimizes it by gradient descent. To prevent overfitting, the number of estimators, tree depth, and data sampling can be controlled.

2.2 Data Overview

The dataset used in this paper consists of 10000 entries and 18 variables spanning demographic, socioeconomic, driver profile, and behavioral risk categories [8]. Key demographic features include Age ("16-25", "26-39", "40-64", "65+"), and Gender ("Famale", "Male"). Socioeconomic categories comprise Education ("none", "high school", "university"), Income ("poverty", "working class", "middle class", "upper class"), and a numeric category Credit Score.

Driver profile attributes include Driving Experience ("0-9y", "10-19y", "20-29y", "30y+"), Vehicle Year ("before 2015", "after 2015"), Vehicle Ownership, and Annual Milage, with the latter two being numeric. Behavioral risk attributes include Speeding Violations, DUI, and Past Accidents, all are numeric data. Additional personal information, such as marital status, number of children and Postal Code ("10238", "21217", "32765", "92101") are also included in the dataset.

The last column of the dataset indicates the prediction outcome which contains binary values, indicating 1 if the client has filed a claim, otherwise it shows 0. While most of the variables of the dataset are well-balanced, others such as Race, Vehicle Type and Postal Code are extremely imbalanced. As shown in Fig. 1, "majority" in Race, "sedan" in Vehicle Type and "10238" in Postal Code occupied the dataset predominately.

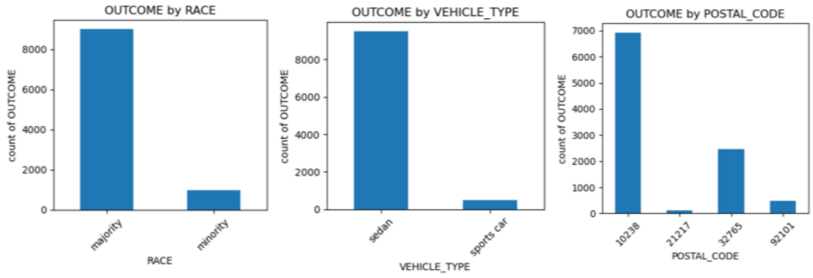

Fig. 1. Outcome by Race, Vehicle Type and Postal Code (Picture credit: Original).

Within the Postal Code category, the imbalance is even more severe, as shown in Table 1.

Postal Code "10238" occupies almost 70% of the entire dataset, while 100% of the data that had postal code "21217" has filed a claim. These two postal codes are expected to have a higher ranking in feature importance, which could mislead the model and fail to reflect an accurate relationship between features and outcomes in a real-life perspective.

Table 1. Counts cases grouped by Outcome and Postal Code.

Postal Code	Class Positive	Class Negative
"10238"	4154	1530
"21217"	0	102
"32765"	1230	746
"92101"	229	158

2.3 Feature Engineering

Due to the imbalance in the dataset, feature engineering was performed based on the Postal Code. The four unique values in Postal Code "10238", "21217", "32765" and "92101" serve as proxies for New York City, Orlando, San Diego, and Baltimore, respectively [9, 10]. The engineered feature, named "Collision Index" was constructed to capture the relationship between collision risk and geographic location. It is defined as:

$$Collision\ Index = \frac{Number\ of\ collisions\ in\ the\ city\ in\ a\ year}{City\ area\ (square\ miles)} \quad (1)$$

This index measures normalized collision density, allowing the model to learn subtle relationships and establish a more balanced interpretation of real-world collision likelihood. The number of collisions for each city was collected from open city databases, police department reports and Kaggle [11-14]. The area for each city was obtained from U.S. Census data [15]. The engineered feature was added as an additional column to the dataset for the following model training. The specific index values are shown in Table 2.

Table 2. Collision Index and corresponding Postal Code.

Postal Code	Collision Index
"10238"	163.5819
"32765"	366.8261
"92101"	61.0462
"21217"	208.3807

2.4 Data Preprocessing and Workflow

Both datasets, original and the one with the new feature, underwent a series of preprocessing steps. Missing values were removed row-wise, categorical variables were encoded, and numerical variables were standardized for consistent scaling, as illustrated in Fig. 2.

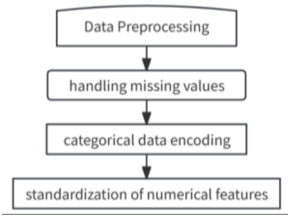

Fig. 2. Data preprocessing pipeline for both datasets (Picture credit: Original).

Fig. 3 illustrates the workflow designed to assess the impact of the newly engineered Collision Index on model performance. Two datasets, the original and the engineered one, were considered. Both datasets underwent identical preprocessing and were split into training (80%) and testing (20%) sets using the same random seed. After the models made their prediction, F1 scores were recorded. This process was repeated 10 times and yielded 10 F1 scores for each dataset. A paired t-Test was then conducted to determine whether the engineered feature made a statistically significant improvement in model performance. Detailed hypothesis testing and evaluation metrics will be discussed in Section 2.5.

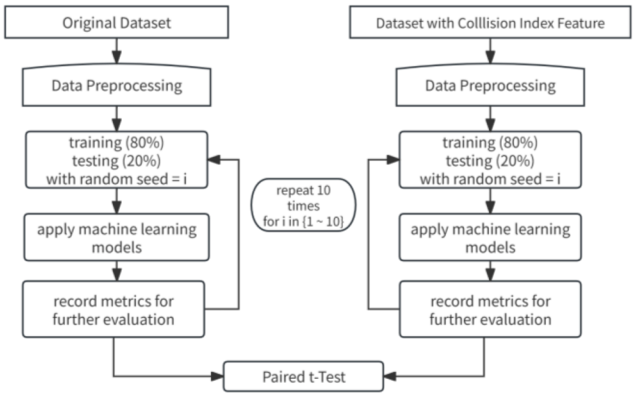

Fig 3. Workflow for each model (Picture credit: Original).

2.5 Evaluation

Confusion Matrix and Related Metrics. Model performance is evaluated using a confusion matrix, which compares predicted and true labels in terms of true positives, false positives, true negatives and false negatives. This paper uses F1 score which balances both Precision (the proportion of correct positive predictions), and Recall (the proportion of true positives correctly identified) [16]. Given the imbalance of the dataset, F1 score is chosen as the primary evaluation metric.

Paired t-Test. To evaluate whether the true mean F1 score of the models trained using the engineered dataset is greater than that of the original dataset, a one-sided paired t-

test was conducted [17]. Let X_i and Y_i denote paired F1 score from the engineered dataset and the original dataset, respectively. Define the differences as

$$D_i = X_i - Y_i \tag{2}$$

The null and alternative hypothesis are defined as

$$H_o: \mu_x = \mu_y \qquad H_a: \mu_x > \mu_y \tag{3}$$

A small p-value (e.g., $p < 0.05$) would lead to rejecting the null hypothesis and suggesting the feature provides a statistically significant improvement.

3 Results

Before interpreting the test results, it is essential to validate the assumptions for the paired t-test.

3.1 Assumption Validation

To ensure the validity of the paired t-test, several assumptions must be satisfied [17]. First, each pair (X_i, Y_i) is generated using different random seeds, ensuring independence. In each case, X_i and Y_i represents the F1 score generated from dataset after and before engineering, ensuring meaningful pairing. A sample size of 10 is selected per model. To assess normality, quantile-quantile (QQ) plots were generated for the differences D_i between two datasets. Fig. 4 illustrates the QQ-plots of 10 differences, with theoretical quantiles on the x-axis and ordered values on the y-axis. The blue dots, true F1 values, follows a straight pattern implied the differences are approximately normally distributed.

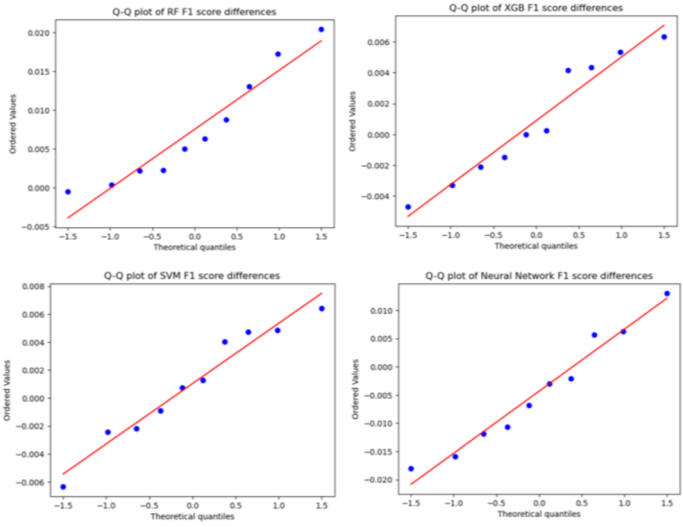

Fig. 4. QQ-plot for F1 score differences generated by four models (Picture credit: Original).

3.2 Results Summary

Among all four models evaluated, only RF witnessed a p-value smaller than 0.05. Therefore, at 0.05 significance level, there is sufficient evidence suggests that the true mean F1 score of the RF model trained on the dataset with the engineered feature is greater than that of the model trained on the original dataset. For XGB, SVM, and NN, the first paired t-test failed to reject the null hypothesis, so a second paired t-test was conducted with the required sample size for test power $\geqslant$0.95. However, even with increased sample size, there was no significant evidence for XGB and NN. SVM was excluded from the second paired t-test due to long training times with approximately 11 seconds per iteration and iterating 203 times on each dataset, which was not feasible within a reasonable runtime. Thus, the effect of the engineered feature on SVM remains inconclusive. In summary, only RF showed a statistically significant improvement with the engineered Collision Index. For other models, the feature yields no clear benefit or inconclusive results. Table 3 presents the results of the paired t-tests.

Table 3. Summary of paired t-test results.

Model	p-Value	Test Power with $n = 10$ (Cohen's d)	Required n for $power \geq 0.95$	Updated p-Value	Conclusion
RF	0.0048	-	-	-	Reject H_o
XGB	0.2459	0.0986	255	0.6703	Fail to reject H_0
SVM	0.2205	0.1117	203	-	Inconclusive
NN	0.8940	0.2253	75	0.2188	Fail to reject H_0

3.3 Limitation

There are several limitations that may have impacted the performance of the model as well as the effectiveness of the engineered collision index feature.

First, as shown in Table 1 and Fig. 1, the dataset is highly imbalanced, with a dominance of negative class in the entire dataset, and an overrepresentation of positive class in "Postal Code 212217". This skewed model learning and distributed disproportionate weight to "Postal Code 10238" and "Postal Code 21217". A more balanced dataset would redistribute the importance weight more fairly and improve the performance of the Collision Index. Additionally, the index is generated from open sources including police reports, city websites and census data, which might be outdated or inconsistent across the cities. Having access to more current and standardized data would enhance the reliability of the index and its contribution during training.

Second, during the hypothesis testing, the observed effect size (Cohen's d) was used to estimate the required sample size, as the true population standard deviation was unknown. This estimation assumes that the sample effect size remains constant as sample size increases, which may not be held in practise. As a result, the second paired t-test could be underpowered or overpowered, which might mislead the statistical conclusion [18].

Third, due to hardware limitations and long model training time, the sample size for some model (i.e. SVM) evaluation failed to meet the requirement to provide a significant and meaningful conclusion. Not all models were carefully analyzed and evaluated,

which lead to potential possibility that additional models may benefit from the engineered feature.

4 Conclusion

In this study, a new feature, Collision Index, was engineered to quantify the collision risk in different locations and aggregated into an insurance claim dataset to assess its effectiveness in improving model performance. Four machine learning models (RF, XGB, SVM and NN) were analyzed and evaluated using the paired t-test comparing the F1 score generated by the dataset with or without the engineered feature.

The results indicate that RF is the only model observed a statistically significant improvement in model performance at 0.05 significance level. For XGB and NN, the results showed no significant improvement was observed even after increasing the sample size to achieve stronger test power. The result for SVM remained inconclusive due to hardware constraints.

These findings suggest that the efficiency of adding engineered feature varies model-wise. Tree-based models like RF benefited from the new feature because it captures localized information about whether a client would file a claim. Although XGB is also tree-based, its gradient boosting mechanism focuses more on the error made by the previous trees, and the model becomes more sensitive to edgy cases and noises, which debased the benefit from the feature index. Models that fail to observe significant improvements may require further tuning, more balanced data or more precise estimation of localized information.

Several limitations that could distort the results, including the imbalanced dataset, the estimated nature of the collision index and variables for hypothesis testing, and computational constraints in model evaluation. Future studies can build upon these findings by refining the limitations and exploring more model structures or distinct datasets for generalizability and robustness of the feature index in a real-world context.

References

1. Hanafy, M., Ming, R.: Using machine learning models to compare various resampling methods in predicting insurance fraud. J. Theor. Appl. Inf. Technol. 99(12), 2819–2833 (2021)
2. Saikia, D.: Machine Learning Enhancements for Car Insurance Claim Prediction. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE. (2024)
3. Moonoo, D., Hosein, P.: Predicting Automobile Insurance Claim Rate Versus Through Severity and Frequency Predictions. 2024 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD). 1–6 (2024)
4. Jonkheijm, T.: Forecasting insurance claim amounts in the private automobile industry using machine learning algorithms. Master's thesis, Tilburg University (2023)
5. Mohammed, A., Kora, R.: A comprehensive review on ensemble deep learning: Opportunities and challenges. J. King Saud Univ. Comput. Inf. Sci. 35(2), 757–774 (2023)
6. James, G., Witten, D., Hastie, T., Tibshirani, R., Taylor, J.: Deep learning. In: An Introduction to Statistical Learning, 3rd edn., pp. 400–401. Springer, New York (2023)

7. Khan, A.A., Chaudhari, O., Chandra, R.: A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. Expert Syst. Appl. 244, 122778 (2024)
8. Sagnik1511: Car insurance data. Kaggle. https://www.kaggle.com/datasets/sagnik1511/car-insurance-data (last accessed 2025/06/26)
9. GeoNames: Postal Codes New York. https://www.geonames.org/postal-codes/US/NY/new-york.html (last accessed 2025/06/26)
10. UnitedStatesZipCodes: Zip codes by state. https://www.unitedstateszipcodes.org/ (last accessed 2025/06/26)
11. Mysarahmadbhat: NYC traffic accidents. Kaggle. https://www.kaggle.com/datasets/mysarahmadbhat/nyc-traffic-accidents (last accessed 2025/06/26)
12. Florida Department of Highway Safety and Motor Vehicles: Crash and citation reports & statistics. https://www.flhsmv.gov/resources/crash-citation-reports/ (last accessed 2025/06/26)
13. City of San Diego: Police collisions – Details. https://data.sandiego.gov/datasets/police-collisions-details/ (last accessed 2025/06/26)
14. Mchen72: Maryland vehicle crash data. Kaggle. https://www.kaggle.com/datasets/mchen72/maryland-vehicle-crash-data (last accessed 2025/06/26)
15. U.S. Census Bureau: City and county profiles: New York City, NY; Orange County & Orlando, FL; San Diego, CA; Baltimore, MD. https://data.census.gov/profile (last accessed 2025/06/26)
16. Sathyanarayanan, S., Tantri, B.R.: Confusion matrix-based performance evaluation metrics. Afr. J. Biomed. Res. 27(4S), 4023–4031 (2024)
17. Wilkerson, S.: Application of the paired t-test. XULAneXUS 5(1), Article 7 (2008)
18. Lakens, D.: Sample size justification. Collabra Psychol. 8(1), 33267 (2022)