**Machine Learning Models to Detect Fraud Auto Insurance Claims**

|  |  |  |
| --- | --- | --- |
| Harika Tumula  *Dept of Computer Science*  *University of Central Missouri*  [*Hxt16640@ucmo.edu*](mailto:Hxt16640@ucmo.edu) |  | Ravali Bhoothpuram  *Dept of Computer Science*  *University of Central Missouri*  [*Rxb23160@ucmo.edu*](mailto:Rxb23160@ucmo.edu) |

*Abstract*—Auto insurance entails an agreement between a client and an insurance provider company that provides financial support in the event of loss or accident of vehicle. Auto insurance fraud involves filling fake insurance claims or staged accidents with the intention of obtaining financial gain through false claims. Automobile fraud is one of the most serious problems that insurance companies are now facing. As a result, discovering this fraud is crucial in order to safeguard the financial interests of insurance companies. Majority of insurance companies rely on specialist knowledge to detect fraud, although expertise information is explainable and reusable, the straightforward method with which it is applied in practice frequently results in some level of lapse in judgment. To address this issue, a machine learning method which is measured in terms of various statistical parameters is implemented in this project. Real world data which consist of labels for claims as fraud or genuine is used to train numerous classification machine learning models such as Support Vector Machine, K-nearest neighbors, Decision Tree, Random Forest and XGBoost. Experimental results which consist of various metrics like accuracy, precision score, AUC, Roc Curve etc. shows that the XGBoost outperformed in terms of all the metrics for this problem.

*Keywords*—Fraud, XGBoost, Machine Learning, Automobile Insurance Fraud.

# 1. Introduction

Technology has advanced for the benefit of mankind but with this advanced technology, we also see misuse and scams happening around. As per insurance information institute, insurance scams cause around 29 billion damage to insurers annually and around 10 billion of damage is caused by people using fake IDs or information to get car insurance. As per a case study conducted by Maryland Auto Insurance in 2018 there were around 8500 vehicles set on fire intentionally to claim insurance fraudulently. There are different types of frauds that occur like false or inflated theft repair claim, intentional damage claim, falsifying the date or circumstances of an accident to get coverage, staged accidents etc. One of the most reported scams is insurance fraud. Insurance fraud incidents are common in day-to-day life and loss on fraudulent claims in the US reached to be around 34 billion dollars as per an insurance fraud survey conducted in 2019 by FRISS.

In most of the developed countries, insurance is a mandatory requirement on all assets like houses, cars, and medical. Once the fraud has been made, it takes a lot of human effort in order to recognize it and fix it. Nowadays we have Machine learning algorithms which usually decrease human efforts on various applications. Very few insurance companies around the world implement machine learning techniques to detect frauds. So, keeping in mind giving something useful to the society, we have tried to implement a system using machine learning that helps to detect automobile insurance frauds. For users, a better classified system can decrease the time for customers to get the cases solved and give more accurate quotes. As for the insurance companies, a better system helps them decrease the cost and time to examine auto insurance cases and most importantly find the true fraud cases to avoid unnecessary payments. As per insurance information institute, it is shown that only 3% of the fraud vehicle insurance cases have been detected and it would be a huge improvement for the auto insurance company. Hence, our system can benefit both the users to claim auto insurance and the car insurance companies to avoid payments to fraud cases to save money and time.

Frauds are unethical and lead to financial loss. It is important to detect and predict a legitimate insurance claim and disregard fraudulent claims which can help in reducing insurance premiums for genuine policy holders.[(https://www.nationwide.com/lc/resources/ auto-insurance/articles/insurance-fraud)](https://www.nationwide.com/lc/resources/auto-insurance/articles/insurance-fraud)

# 2. Motivation

# 2.1 Literature Survey

Many researchers have already proposed various machine learning approaches to tackle fraud detection problems, these studies have aided us in better understanding viable approaches and solutions for our project. (Bhowmik, 2011) conducted one such study, in which they primarily followed three phases. The first is to use the k-means clustering technique to discover clusters, which has the best performance for large-scale data. Second, they create a rule set using a decision tree technique, and then they visualize the results of their model. They also use a Bayesian network in conjunction with the decision tree to improve forecast accuracy. Various Metrics like confusion metrics, accuracy, Precision and recall are used to evaluate the model. (Nian et al., 2016) conducted another study in which they employed the unsupervised Spectral Ranking algorithm for anomaly, also known as SRA. They believe that SRA can be used as a supplement to the unsupervised support vector machine technique or as a better alternative. They used an available auto insurance claim dataset to conduct their research, leaving labels unsupervised while generating ranks. They demonstrated that their model overcomes the drawbacks of several fraud detection algorithms that are based on outliers by employing the SRA approach. They came to the essential conclusion that when building an efficient model, it is critical to use the proper measurements for similarity checks. We considered all prior studies on auto insurance policies and machine learning applications to obviate automobile insurance fraud in this project. We further used advanced techniques like SMOTE for handling imbalanced class classification dataset to improve the model accuracy score and dimensionality reduction algorithm, PCA (Principal Component Analysis) that helped in increasing model efficiency further.

# 3. Proposed Approach

Our approach includes comparison of various supervised machine learning models to check which algorithm performs efficiently in categorizing fraud insurance claims. To achieve this data is first evaluated for biasing issues before model creation because most fraud detection datasets are imbalanced, and this biasing is addressed using SMOTE (Synthetic Minority Over Sampling Technique). Using the python module scikit learn, this balanced dataset is then utilized to train several machine learning algorithms and various metrics like accuracy, F1-score are calculated. Further to improve accuracy, hyper tuning of parameters using random grid search is done to check for optimal parameters. The metrics of multiple models are compared using this parameter, and the model with overall better performance is stored in a pickle file.

# 4. Project Development Methodology

Remote pair programming method is used to develop this project. This pair programming is an agile development

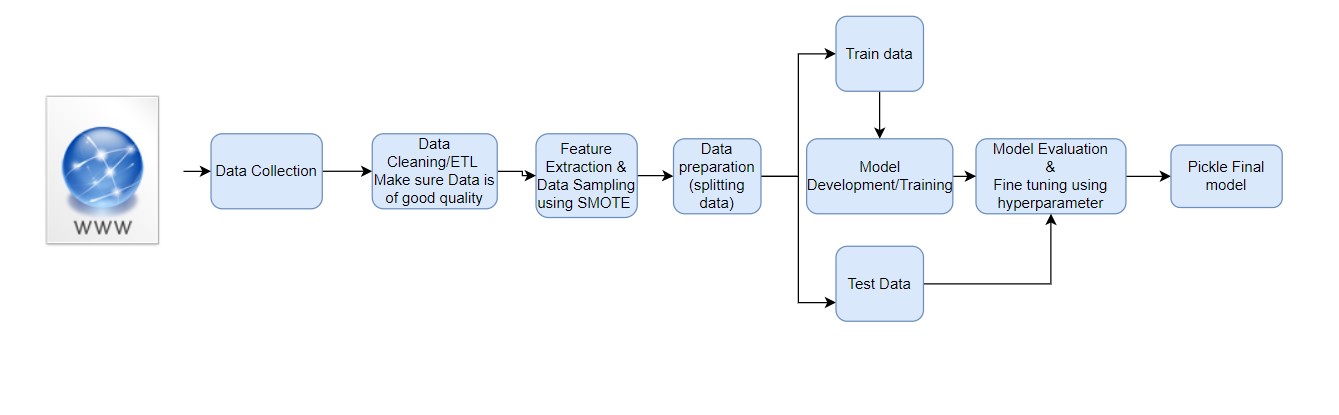


Figure 1. Architecture

methodology where developers can work in one workstation. There are two roles in this method: driver and navigator. The driver is who writes the code, and the navigator is to suggest and give feedback instantly which should be taken care of by the driver who implements and develops ideas to solve the tasks. If there are multiple developers, the role can be changed among the developers frequently or according to their need. This approach is best to establish team coordination and make progress in work effectively. The knowledge is shared among our team and time taken to code this project was low using this methodology.

We have established this method through zoom calls, screen sharing was conducted among the developer navigator pairs. The developer will share the screen while the navigator views it and gives suggestions or feedback, and the developer instantly applies the ideas that are shared by the navigator in code to improvise. Other than this pair programming we also followed the scrum method. We have assigned tasks and shared work equally. We have conducted meetings every week to exchange the status update and assign new tasks for the following week. This approach helped us in finishing our work and meet the deadlines assigned. The evidence is attached in an agile user stories document and submitted along with this report. All five in the team have worked and shared work equally which led us to successfully build a classification model and finish this project.

# 5. Data Engineering

## 5.1. Data Collection

The purpose of our project is to help insurance companies to detect fraud insurance claims to avoid unethical and unnecessary payments to the false claims.

In order to achieve this goal, we need to build machine learning models to have the ability to determine whether it is a false claim or not based on the input insurance claim data.

Hence, we need data about each claim in terms of information about the insurance claimer like income, age, education and claim history, as well as information about the car like car type, car usage purpose and car age. During our literature survey, we found this dataset from Kaggle including information of details about each claim.

This dataset contains 1000 vehicle claims from Ohio,

Illinois and Indiana in the period from 2015 January to March. This dataset contains 39 features as input parameters, the following section includes a detailed explanation about each variable.

Parameter List:

1. month as customer: duration of the insurance claimer being a customer
2. age: age of insurance claimer.
3. policy number: a unique key to identify each insurance policy.
4. policy bind date: policy begin date.
5. policy state: in which state the policy was made.
6. policy csl: policy combined single limit.
7. policy deductible: deductible amount for that policy.
8. policy annual premium: policy annual premium amount.
9. umbrella limit: what amount is the umbrella insurance set to in order to provide extra protection.
10. Insured zip: the zip code of the insurance disclaimer.
11. Insured sex: the gender of insurance disclaimer.
12. Insured education level: the education level of insurance disclaimer.
13. Insured occupation: the occupation of insurance disclaimer.
14. Insured hobbies: the hobbies of insurance disclaimer.
15. Insured relationship: the relationship status of insurance disclaimer with children or not.
16. Capital-gains: capital gain of insurance disclaimer.
17. Capital-loss: capital loss insurance disclaimer.18. Incident date: the date of incident happened.
18. Incident type: the type of incident.
19. Collision type: the type of vehicle Collision.
20. Incident severity: how severe is the incident.
21. Authorities contacted: what authority department reached out to.
22. Incident state: in which state the incident happened.
23. Incident city: in which city the incident happened.
24. Incident location: the detailed location where the incident happened.
25. Incident hour of the day: the hour when an incident happened in 24-hour style.
26. Number of vehicles involved: how many vehicles are involved in the accident.
27. Property damage: whether there is damage to property.
28. Body injuries: body injuries represent in integers from 0 to 2.
29. Witnesses: number of witnesses, integer from 0 to

3.

1. Police report available: whether there is police report.
2. Total claim amount: the total amount this incident claim for.
3. Injury claim: injury claim amount.
4. Vehicle claim: vehicle claim amount.
5. Auto make: the brand of vehicle.
6. Auto model: the model of vehicle.
7. Auto year: the production year of vehicle.
8. Fraud reported: ‘Y’ or ‘N’ to represent the actual case being fraud or not.

## 5.2. Data Pre-processing

In order to get our data ready for further analysis and model building, we first check for missing values. The result shows only one column has all null value, and we drop that column since it is meaningless for our project. Also, we check box plot for each feature for outlier value and we find outliers exist in feature ‘umbrella limit’ which is handled later in the data transformation section. As for duplication, there are no duplicate data entries in our dataset.

## 5.3. Random sampling

To ensure random sampling of data selection each time, we use train test split function to select training and testing data with random state equals 20.

## 5.4. Transformation: data relabeling, normalization

Furthermore, to ensure all data types are uniformed for analysis, we perform more data type checks. We find that for column ‘policy bind date’ and ‘incident date’, data is in type object. For easy implementation to access data, we convert those’ types to date datatype.

Moreover, we discover that there exist many columns using ‘?’ as column value including ‘collision type’, ‘property damage’, ‘police report available’. We replace ‘?’ using ‘unknown’ to guarantee the consistency of data.

Furthermore, we check for outliers in the dataset, and we find for some value of column ‘umbrella limit’ is negative. It would be meaningless to have a negative protection value limit, so we reset those negative values to 0.

During the exploration of our dataset, we also found several columns have too many unique values and it is not helpful for us to build a generalized model, so we decide to drop column ‘incident location’, ‘policy bind date’, ‘insured zip’, ‘incident date’, ‘authorities contacted’, ‘auto make’, ‘auto model’ to perform feature reduction.

Last, we transform non-numerical data to numerical data by encoding so all data fields are in numerical type in order to visualize and build machine learning models.

## 5.5. Statistics

To get a better understanding of the dataset, we use the ‘describe’ method to get a statistical report. For numerical data types, we find the mean, std, min and max value. For object data types, we find the unique value count and top value frequency. We also visualize the distribution of the target variable, and we know we have an imbalanced target variable. There are only 25% fraud claims with remaining 75% claims being honest claims. We handle imbalanced data later. We use count plot to visualize discrete variables.



Figure 4. Number of claims vs Fraud Reported

Some insights we get from statistical analysis:

1. Most fraud claims are distributed evenly from different brands of car, except very least fraud cases for Honda.

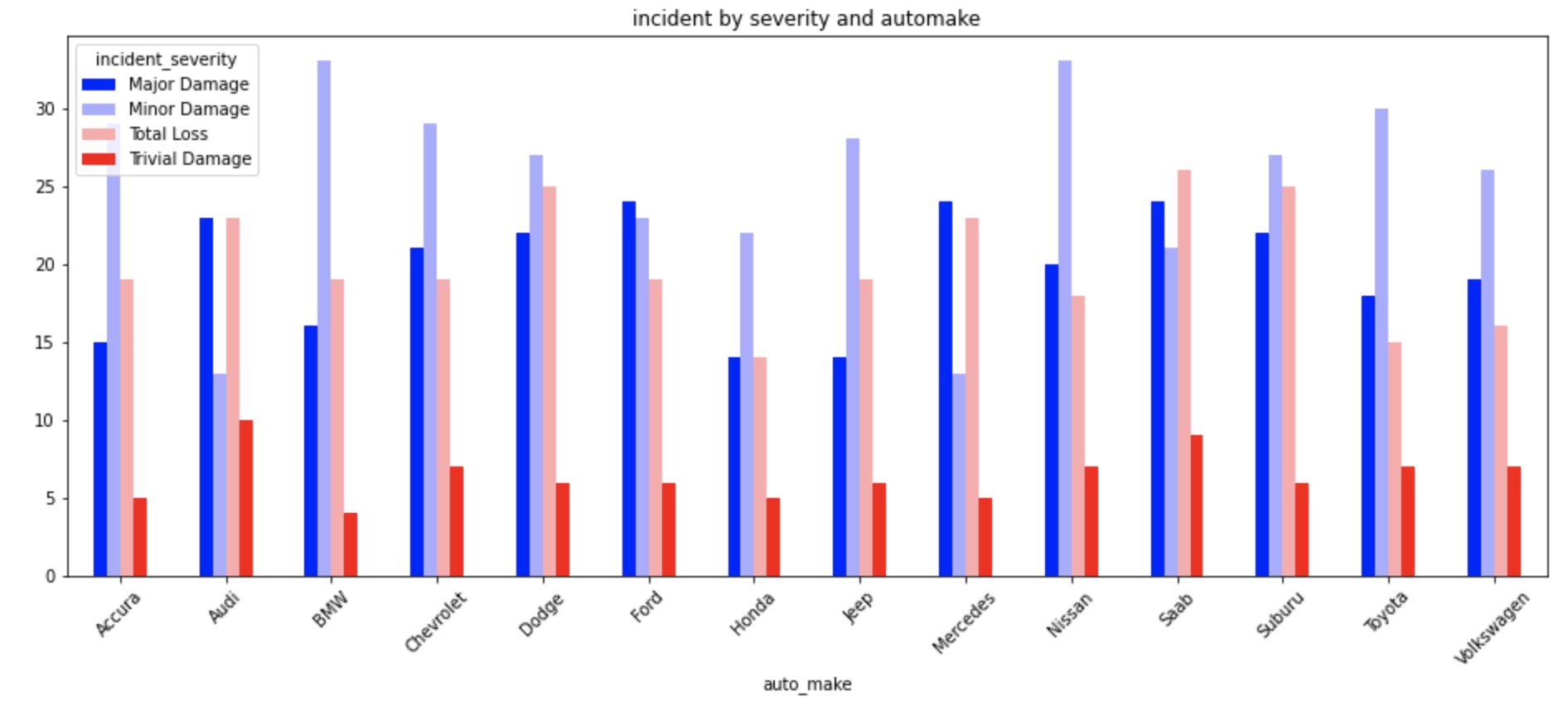


Figure 5. Incident Severity vs Auto Made

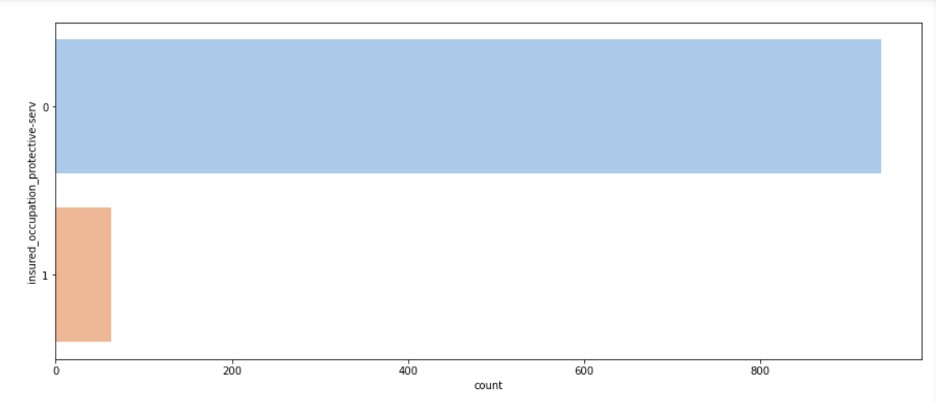


Figure 6. Insurance Occupation vs Fraud Count

1. For most accidents, multiple vehicles are involved due to collision, very rarely incidents are caused by theft or parked cars.
2. Feature ‘policy annual premium’ has normal distribution.

## 5.6. Data Exploration

To gain a deeper understanding of relations of each parameter with fraud claims, we conduct more visualizations. From the graphs, we can conclude there is no clear correlation between ‘month as customer’ and fraud claim amount. And for a claim amount around 60000, it is more likely to have fraud claims due to the impressive money gain. While for claim amounts below 20000, it is less likely to have fraud claims due to the limited money gain and claimants would think not worth forgery.

# 6. Model Development

To accurately forecast fraud insurance claims, which is a classification problem using the labeled data, we used supervised learning algorithms to see which one best suit for detecting automobile insurance fraud and assisting organizers in reducing losses.

6.1. Support Vector Machine:

Support Vector Machine (SVM) with less computation power produces significant accuracy. SVM can be used for both regression and classification problems. To separate two classes there are many hyperplanes present, but the main goal of SVM is to find the hyperplane that has maximum

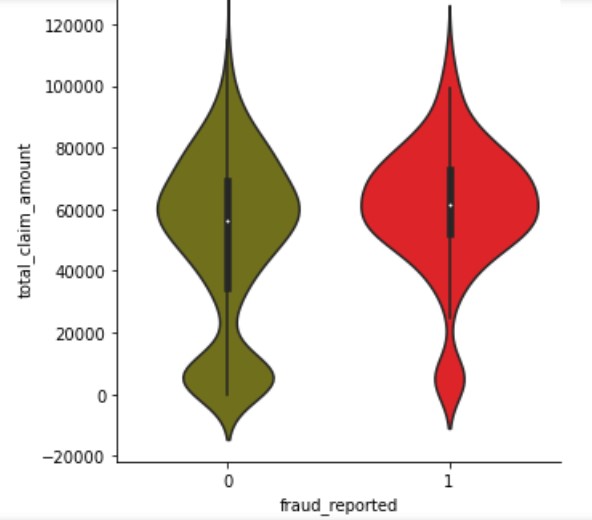
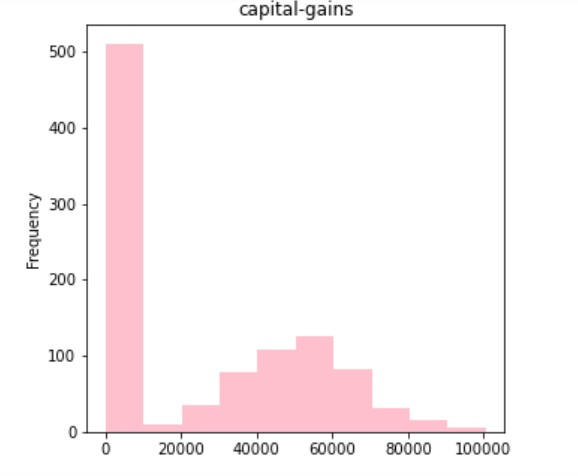
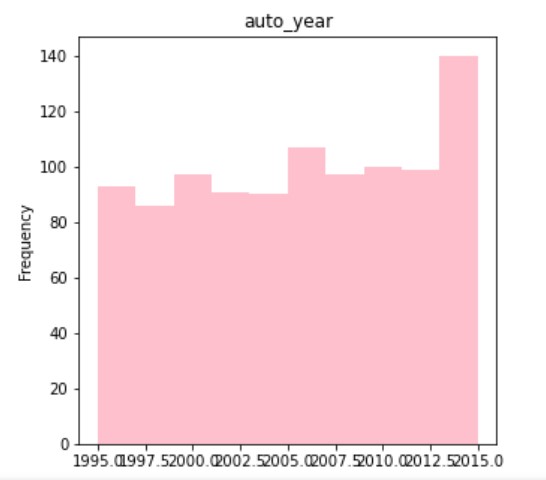


Figure 7. Capital Gain vs Fraud Claim Count

Figure 10. Total Claim Amount vs Fraud Claim Amount

margin. By maximizing the margin, the data points in future can be classified with more confidence.

Hyperplanes are dividing the two classes or training sets, using SVM choose the best classifier or hyperplane which is equidistant in both the groups.

*K*(*X,Y* ) = (*γ.XTY* + *r*)*d,γ >* 0

In this project, we first built SVM without changing any hyperparameters, i.e. We used the default kernel parameter of SVM, kernel=’Linear,’ and regularization parameter, C = 1. with default parameter and as dataset is biased towards No-fraud cases, specificity and Cohen Kappa obtained were below 50%, To deal with the Bias Dataset issue, we used the sampling approach SMOTE, in which new samples for the minority class are formed from existing samples using KNN, and for hyperparameters, we used random search grid cv technique to achieve the best values for our dataset. We used SVM for our project because:

* SVM is memory efficient.

Figure 8. Auto Year vs Fraud Claim Count

Figure 9. Month as Customer by Age by Fraud Count • Effective in higher dimensional space.

* Works well when there is a clear margin of separation between two classes.

6.2. K Nearest Neighbors:

The KNN stands for K nearest neighbors, this is a supervised machine learning algorithm which is used to solve classification and regression problems. Decide the k value to calculate the nearest neighbors that must be classified or predicted. This algorithm stores all possible cases and classifies new cases based on similarity measures. KNN is called a lazy algorithm since it doesn’t do any calculations during training time

instead, just stores until the dataset is finished.

To calculate the nearest neighbors, set parameter k to

the number of nearest neighbors, then calculate the distance between the query-instance and all the training samples.

Based on the kth minimum distance we must decide on the nearest neighbors.

There are several distance functions in KNN as shown in the figure below and we have used Euclidean distance for this project.

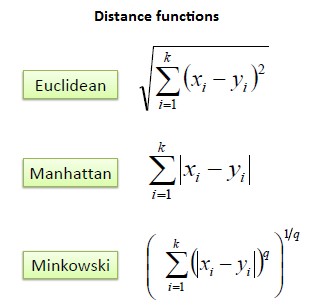


Figure 11. Distance Functions

Why KNN is chosen for our project:

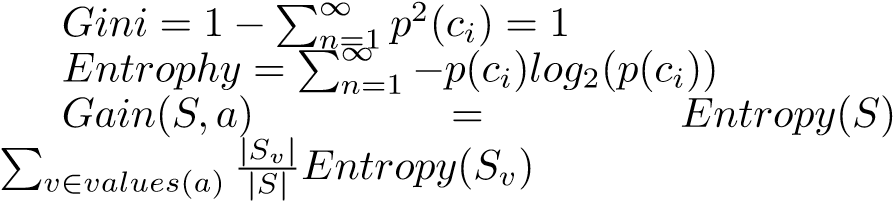
* The algorithm is versatile, it can be used on classification or regression problems.
* Simple and Easy to implement.
* No need to build the model, just tune the parameters.

6.3. Decision Tree:

Decision Tree algorithm is also supervised learning, this is good with huge datasets and takes less time. The Decision Tree algorithm is also used to solve regression and classification problems. The goal of this algorithm is to create a training model and predict the class by learning simple decision rules inferred from prior data.

To predict the class label, we should start from the root of the tree. Compare values of root attributes to record attributes, then we follow the branch and jump to the next node.

Decision tree is intuitive and has no assumptions or constraints, making it simple to understand and allowing reasons to see results.

 −

The decision tree is used in our project because:

* It helped us in identifying important features required for the classification.
* It is robust and works well with huge datasets and takes less time.

## 6.4. Random Forest

Random Forest is also a supervised learning algorithm, the forest it builds is an ensemble of decision trees trained with a bagging method. The idea behind the bagging method is a combination of learning models which increases the overall result.

The Random Forest was used in our project because:

* It helped to increase decision tree consistency by reducing overfitting.
* it ran efficiently on large datasets.
* It was robust to outliers.

6.5. XGBoost:

XGBoost is a machine learning algorithm which is efficient on tabular or structured data. This is an implementation of gradient boosted decision trees which are designed for performance and speed. It is an open-source library which aims to provide a “Scalable, Portable and Distributed gradient boosting library. This has recently started growing in popularity in machine learning for dealing with small, structured data and producing high accurate results.

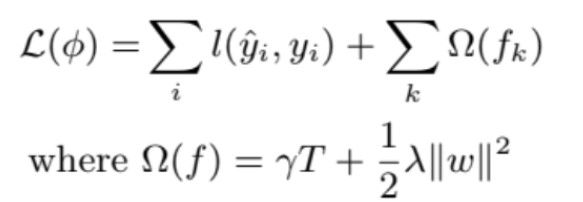


Figure 12. XGBoost Equation

We used this model because:

* This can handle missing values.
* Supports regularization and can cache.
* Allows us to do cross-validation at each step of the boosting phase.
* For its full potential it used parallel computing.

A major challenge during model development was finding the right training data set with fraudulent claims as frauds are less common compared to legit insurance claims. Training data with sufficient fraud claims was not easily available. To address this problem, we used one of the resampling methodology SMOTE (Synthetic Minority Oversampling technique) that helped in handling imbalanced class classification.

Another challenge was the number of features in the dataset i.e., 90 and we used PCA (Principal Component Analysis) to reduce the number of features to 72.

# 7. Evaluation and Comparison

The implementation of auto insurance fraud detection was done using five different classification algorithms. The performance metrics such as accuracy, precision, recall, f1-score, confusion matrix, sensitivity, specificity, Cohen Kappa score, ROC curve or Area under ROC curve(AUC), precision-recall curve, time taken by each algorithm to train the model are calculated for measuring and comparing the performance of the model. Our Dataset being unbalanced, more concentration to metrics like Cohen Kappa, Precision recall curve was given while selecting the best model.

## 7.1. Confusion Matrix

Confusion matrix is a table that is used to describe the performance of a classification table on a test data set whose true values are unknown. The true positives, true negatives, false positives, false negatives predicted and shown in 2\*2 matrix with actual values in one axis and predicted in another.

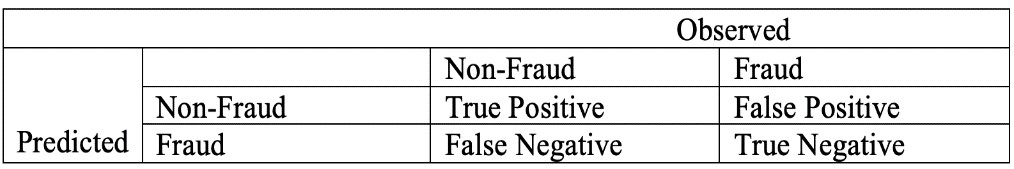


Figure 13. Confusion Matrix

## 7.2. Accuracy

The ratio of true results to the predicted values of total data is called Accuracy. Accuracy is calculated for both training data and test data for each model.



## 7.3. Precision

This is valid when we want our prediction to be sure. Precision is a ratio of true positives to the actual positives calculated.

## 7.4. Recall

This is good when we want to capture as many positives as possible. The ratio of true positives to the correct positive class predictions plus incorrect class predictions.



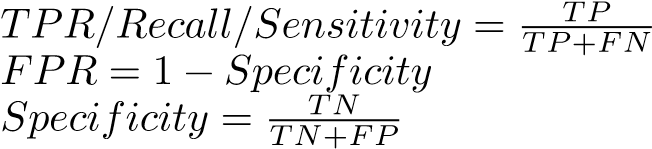
## 7.5. Cohen-Kappa Score

Cohen-Kappa is a metric used to measure the agreement between two raters. To access the performance of the classification model we can use this metric. The value of kappa can be less than 0 or negative. This measures the agreement between actual and predicted values.



## 7.6. ROC – AUC Curve

AUC stands for Area under the curve and ROC is receiver operating characteristics. This curve is the measurement for classification problems at various thresholds. ROC is a probability curve where AUC is the degree of separability. This explains the capability of the model in distinguishing between classes. If AUC is high, then it is predicted as expected. The trade-off between sensitivity and specificity is shown by the ROC curve. Sensitivity is TPR and specificity is 1- FPR. If the classifier gives the curves close to the top-left corner, then it indicates better performance.



## 7.7. F1 – Score

F1-score is used when we want good precision and recall. Harmonic mean of precision and recall is called f1score.



Below table summarizes the metrics of all models developed in this project:

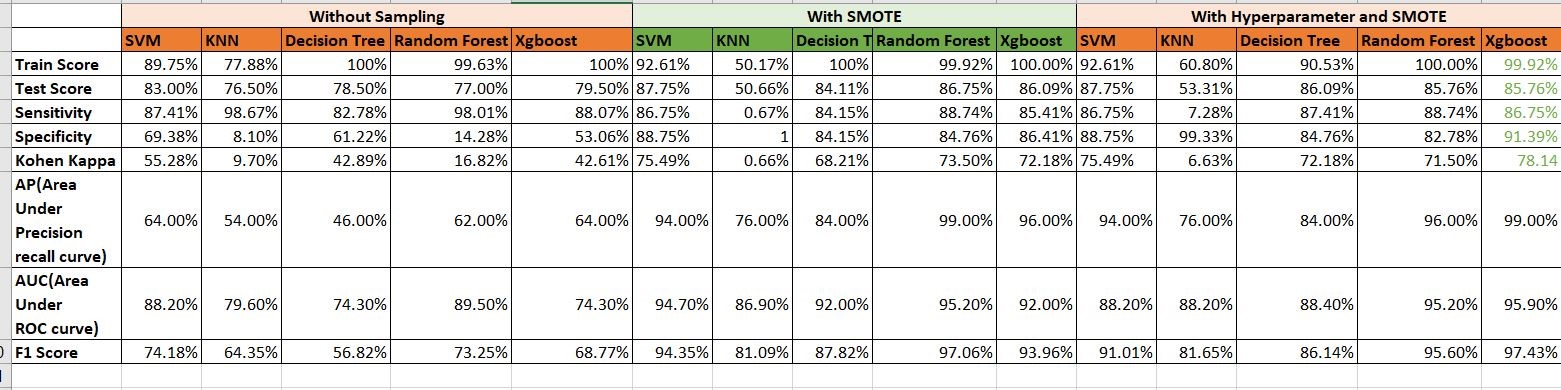


Figure 14. Metrics of All Models

## 7.8. Comparison of Model Performance

We tried five different machine learning models to detect the best accurate model for our project. First, we compared all the models without applying any tuning, the accuracy score of all the models obtained was mostly high which indicates low bias. However, we cannot simply rely on accuracy as our dataset was biased which was also evidence from specificity and Cohen Kappa score of models. So the main step here was to remove dataset bias by using some oversampling technique, hence we used SMOTE which is an oversampling technique in which samples for the minority class are formed from existing samples using KNN, typically with K=5. After SMOTE dataset was balanced having equal number of both the classes, using this dataset we ran all the model again and compared various metrics, this time accuracy was increased as compared to first case , Also we can see improvement in Cohen Kappa which tells that accuracy is not by chance but it is actual accuracy of your model. Then we also concentrated on F1 score of models and precision score as our main aim was to detect fraud cases and precision is one metric which tells us how many fraud cases were correctly detected. In this comparison we finally chose SVM, Random Forest and XGboost as all of them were performing better. Lastly, we performed hypertuning by using randomized grid search cv to obtain best parameters for our models. After hypertuning we compared models and checked for Cohen Kappa, F1 score and precision recall curve to select the best model in detecting fraud cases.

In the next step we used First we compared all the models without any treatment then to select the most accurate model, we have calculated performance metrics for each model and compared the metrics among them then came up with the final model with high performance. Initially we applied all the models without any tuning in the dataset, the dataset was imbalanced, and we observed the specificity is very low. Also, precision and recall curve for minority class is less. Cohen kappa score for imbalanced data is below 50%. So, used synthetic minority over sampling technique (SMOTE) to handle the imbalanced dataset. Using the standard scaler method scaled down all the input values. Then after applying SMOTE again applied all five machine learning models then Cohen kappa score has been increased. We have dropped some features like C39 which has a greater number of null values and is not useful for analysis.

After SMOTE we have done hyper tuning and applied it to all models. So, performance metrics for all five models without sampling, with SMOTE and with hyper tuning has been done and noted. Considering all these scenarios one best high-performance model will be chosen for detecting insurance frauds.

## 7.9. Results of the XGBoost Model

The classification report, confusion matrix, Cohen kappa score and the precision-recall curve of the XGBoost algorithm are shown below.

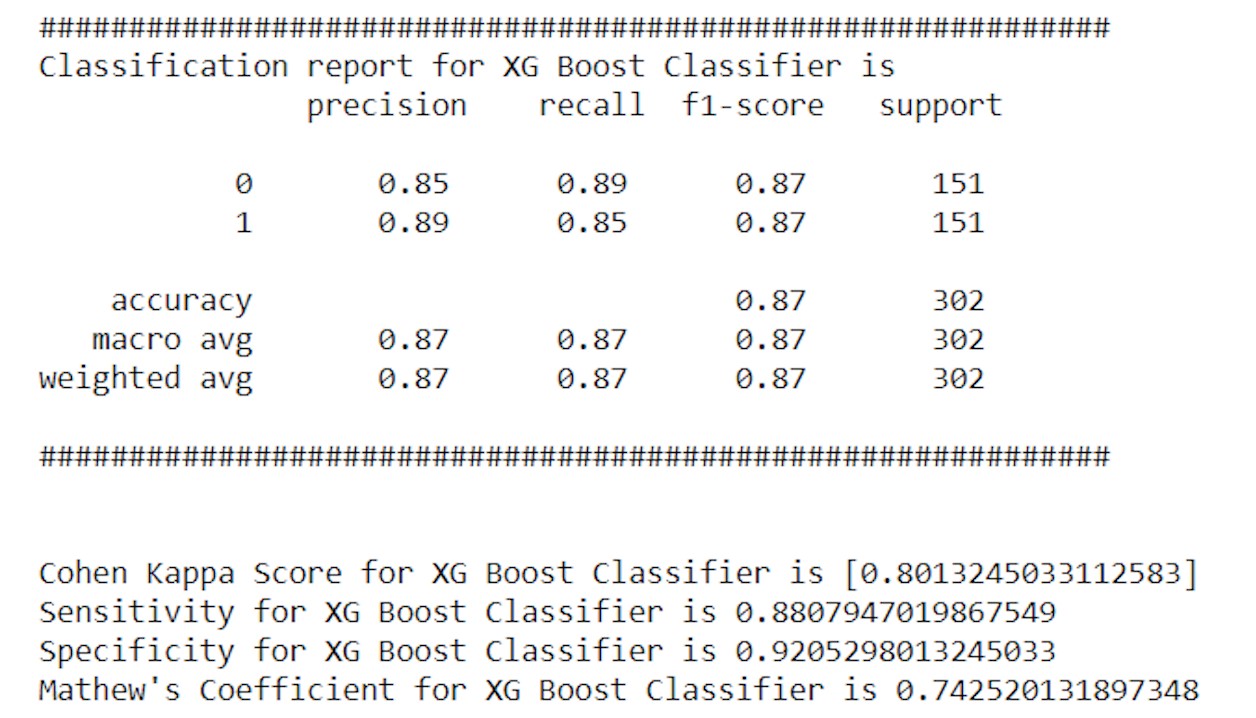


Figure 15. XGBoost Model Result

Sensitivity obtained is 88.07% which indicates True Positive rate or recall score, meaning how many correct results or positive cases are recalled.

Specificity obtained is 92.05% which indicates how specific the algorithm was to detect false cases.

Precision score for both the classes of fraud detection is between 85-89%, However precision score of class 1 which is 89% indicates how many fraud cases are predicted correctly out of total fraud cases.

Cohen Kappa Score of 80.13% which indicates that obtained accuracy of 99% is not by chance.

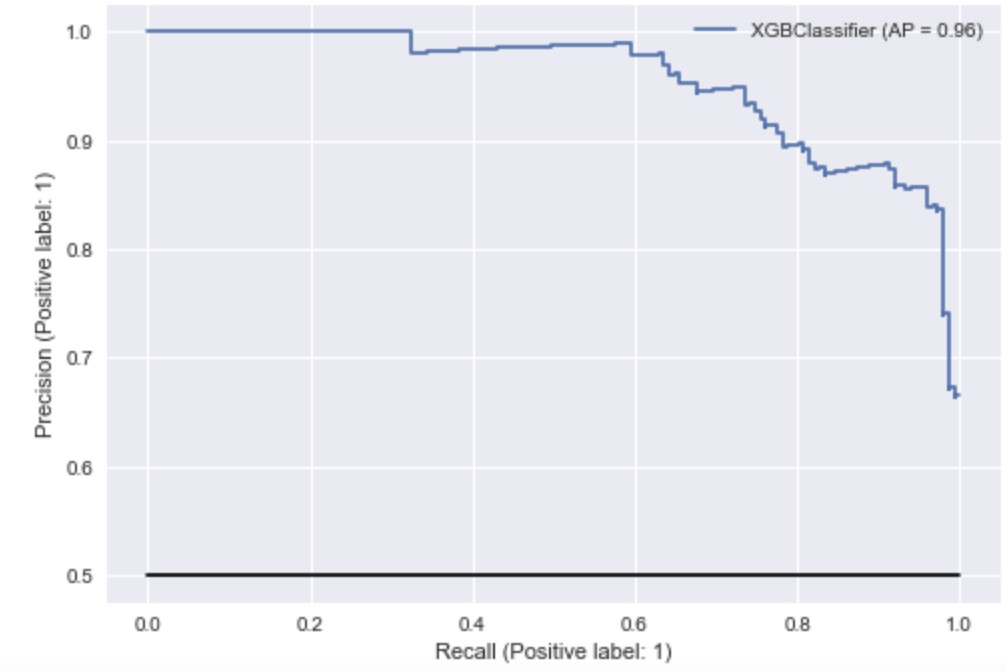


Figure 16. Precision by Recall

The consolidated ROC curve of all the models is as shown below. From the ROC curve, we can see that KNN auc value is the lowest for every run and hence least performer on our dataset.

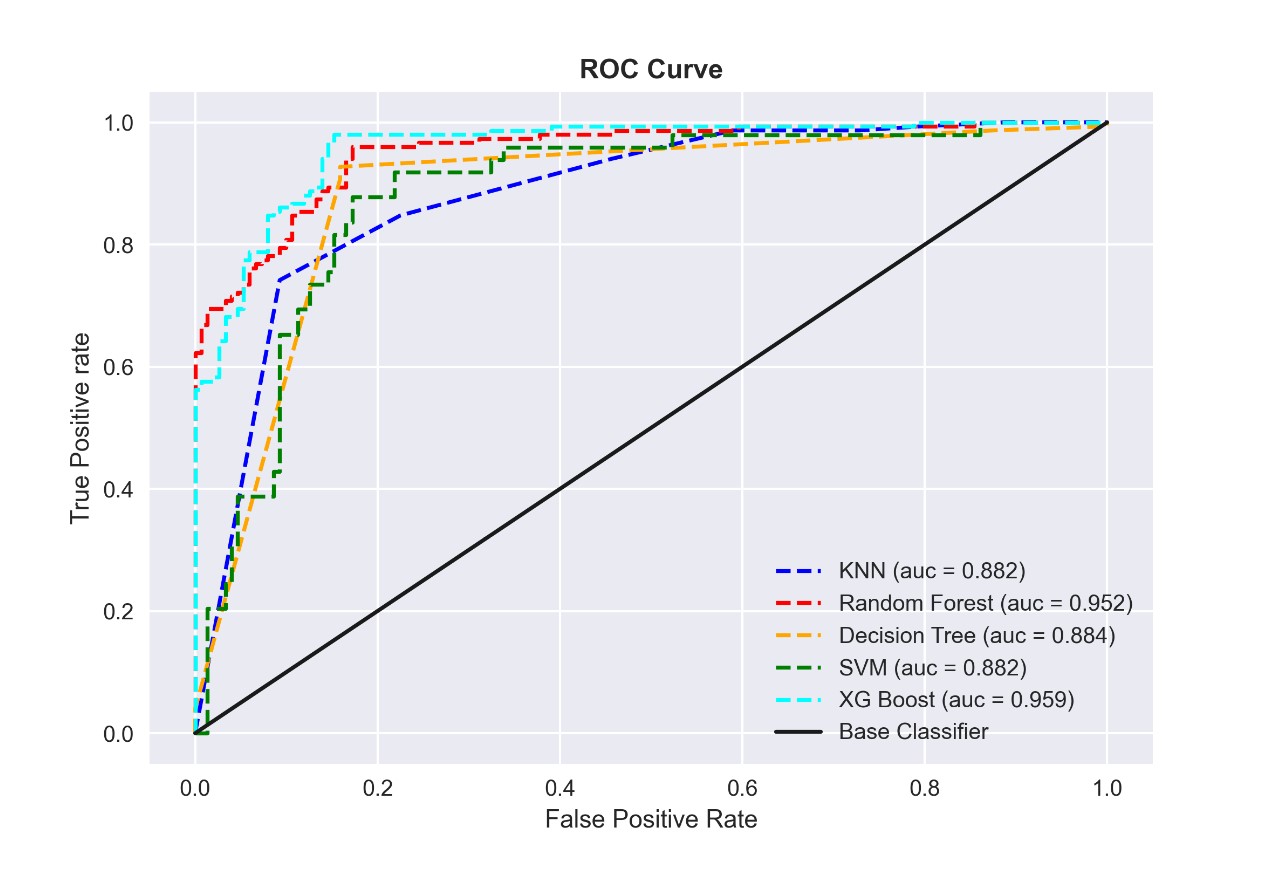


Figure 17. ROC curve

# 8. Conclusion

To detect the insurance frauds and cut losses for organizers the machine learning algorithms should be capable of detecting the frauds accurately. We built a machine learning model by preprocessing the raw datasets and transformed the data which is best to detect the frauds and made our dataset available to the public in GitHub. By this user can easily get access to our dataset. Companies can use models to find out the auto insurance frauds. Users can try our machine learning models built without doing much of the feature engineering which we had to do initially.

We compared various ML models developed using metrics like kappa score, precision recall curve and F1-score.

We can conclude based on comparison that XGBoost performed better than other models. Training accuracy of 99% and testing accuracy of 86% is achieved which is less when compared to random forest algorithm but for unbalanced dataset cases precision recall curve,F1 and Kappa Score plays an important role and XGBoost performed well in terms of all these metrics. In the classification report we can see the F1 score for both Fraud and Non-Fraud cases is above 80% which indicates that the model performed better in predicting both non-fraud and fraudulent cases. By looking at the confusion matrix we can say that models have a larger rate of false alerts than frauds that go undetected. In our instance, it is preferable to detect more frauds than to allow fraud cases to go undetected. As a result, we can conclude that our approach was successful in detecting fraud claims.

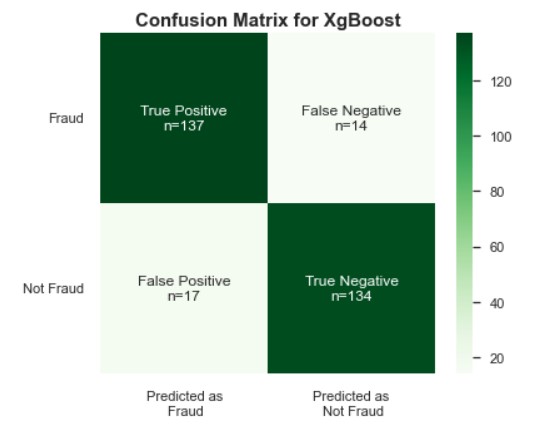


Figure 18. Confusion Matrix for XgBoost

# 9. Future Scope

* Dataset has around 1000 rows and this dataset is of one year auto insurance data, we will collect more data which is of 5 years data which would improve the model’s performance on classification.
* The machine learning model developed can be used to label unlabeled data in future.
* We would plan to work on live data in future which can help the organizers to cut losses.
* Shift the entire process of training and testing to the cloud to make it easier to integrate on several applications.

Acknowledgments

The authors we would like to thank...

# References

[1] Khatri, S., Arora, A., and Agrawal, A. (2020). Supervised machine learning algorithms for credit card fraud detection: a comparison. IEEE, 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 680–683.

[2] Dua, D., & Graff, C. (2019). UCI machine learning repository. Retrieved September12, 2019, from http://archive.ics.uci.edu/ml.

[3] Arenz, O., Zhong, M., and Neumann, G. (2020). Trust-Region variational inference with Gaussian mixture models. Journal of Machine Learning Research, 21(163), 1-60.

[4] Van Engelen, J. E., and Hoos, H. H. (2020). A survey on semi-supervised learning.Machine Learning, 109(2), 373–440.

# Appendix

## 1. Visualizations

Few patterns observed in our dataset using exploratory data analysis (EDA):

* More auto insurance claims are observed to be fraudulent by people who have served as exec-managers
* People with an education level of JD are involved in more fraud transactions.
* Comparatively, people with lesser education are claiming more fraud claims.
* Another interesting observation is that people with hobbies such as chess are claiming more fraud claims.
* Gender is not impacting any fraudulent claims.
* Auto makers of ’Ford’, ’Mercedes’, ’Chevrolet’ and ’Audi’ have the highest Fraud claim.
* Age has no impact on fraudulent claims. A group of people across age from 20-40 are relatively new insurance customers

## 2. Significance to real world

* The dataset generated from the Kaggle is preprocessed and transformed data is accessible to the public where they can easily download and apply models to detect the insurance frauds.
* We applied four different machine learning models and found the best accurate model to detect the insurance claims.
* In future companies can directly use this model and detect the frauds.

## 3. Report

The effort on the report is in the team, we worked together to frame this report and we referred to many documents and references are mentioned in the ‘References’ section in this project report.

## 4. Key Learnings

* Working on different models in machine learning helped us to correlate between time and accuracy requirements on each model practically.
* Applying all the models that we learned in class helped us to learn in depth by practically using those models in our project.
* Choosing the right kernel in SVM, because depending on the problem each kernel type either increases or decreases the accuracy.
* Handling an imbalanced dataset improves overall model performance, and the accuracy obtained is not by chance once the dataset is balanced.

## 5. Prospects of winning competition

In our literature survey we found that most of the research papers haven’t used XGBoost for insurance fraud detection, we included XGBoost in our model list and it achieved the highest accuracy with the least execution time.

## 6. Velocity

XGBoost can parallel process and cache optimization, which will be reliable for real time data, since we are planning to apply this on live data this would be the best. It also applies regularization techniques internally to avoid overfitting by not causing any impact on classification.

## 7. Innovation

We haven’t directly worked on the dataset that is available, we did preprocess and removed features that are not needed and did label encoding on some features. We haven’t directly worked on the dataset that is available, we did preprocess and removed features that are not needed and did label encoding on some features.

## 8. Evaluation of Performance

All the models are compared using different performance metrics like accuracy, precision, recall, Roc curve, precision-recall curve and kappa score.

Along with the efficiency, the execution time is also an important metric, which was considered along with accuracy comparison. The final model was chosen with high accuracy with the least execution time which is ideal for real time executions.

## 9. Teamwork

We are a team of two, collaborating and sharing the work equally to complete the project more effectively and efficiently. Each of our opinion and feedback taken into consideration during the entire process. We have assigned the works and timelines given and arranged meeting timings weekly once to discuss the status updates and further steps.

## 10. Technical Difficulties

Exploratory data analysis was another challenge because there were many features (¿90) and we had to explore and analyze the features that may/may not impact our model. The features identified should help to increase the efficiency rather than increasing the dimensionality leading to overfitting.

Finding the right data set with fraudulent claims was a challenge as frauds are less common as compared to legit insurance claims. Training data with sufficient fraud claims was not available easily. This is called ‘imbalanced class classification’. All the datasets which we found were biased from auto insurance companies had no fraudulent data.

## 11. Resolution of technical challenges

Handling imbalance data can have a significant impact on the performance of classification systems. The prediction will be biased in favor of the dataset’s majority class. To address this problem, we used one of the re-sampling methodologies SMOTE (Synthetic Minority Over-Sampling technique) that helped in handling imbalanced class classification

We used PCA (Principal Component Analysis) to reduce the number of features.

## 12. Practiced Pair Programming

We did pair programming to work collaboratives and meet the deadlines of assigned work and to establish team coordination to produce working code rapidly. We have explained pair programming in detail under the “Project Management Methodology” section in this report.

## 13. Practice Agile/Scrum

Scrum meetings are held weekly and document the work every week. We have submitted the agile document along with this project report.

## 14. Used Grammarly

After successfully finishing all the tasks and report writing we have Grammarly checked to correct all the grammatical errors and tested for plagiarism check to make sure we don’t fall under plagiarism or unknowingly take

credit for others work. The report for all these checks has been submitted to canvas along with this report.

## 15. Used Latex

Latex is used for preparing the whole document with high quality typesetting by following the IEEE format

## 16. Creative Presentation Techniques

We have used Microsoft PowerPoint for creating slides for project presentations.

## 17. Literature Survey

We conducted a thorough literature review of all methodologies and techniques used to solve this problem to date. We looked at almost ten to twelve academic articles to get a better understanding of the problem and how machine learning techniques are implemented. We developed our approach to solve this problem after a thorough review of all articles and came up with a model that provided maximum accuracy and, most importantly, a decent kappa score, which is required for an imbalanced dataset. More information about the literary survey can be found in the Literature Survey section of the publication.