Insurance Fraud Detection Using Machine Learning

Veer Pal Singh
Data Scientist
16/12/2021

Outline

- ✓Executive Summary
- ✓Introduction
- ✓Methodology
- ✓Results
- ✓Conclusion
- ✓Appendix

Executive Summary

- ✓ Summary of methodologies
  - ✓ Extract Transform Load(ETL)
  - ✓ Data Cleaning
  - ✓ Feature Engineering
  - ✓ Model Definition
  - ✓ Model Training
  - ✓ Model Evolution
  - ✓ Model Deployment
- ✓ Summary of all results
  - ✓ Exploratory data analysis
  - ✓ Anomaly Detection
  - ✓ Result of all Classification Algorithms
  - ✓ Voting Classifier Results
  - ✓ XGB Classifier-Grid Search Result
  - ✓ Model Performance
  - ✓ Confusion Matrix Outcomes
  - ✓ Accuracy after Dimension Reduction

Introduction

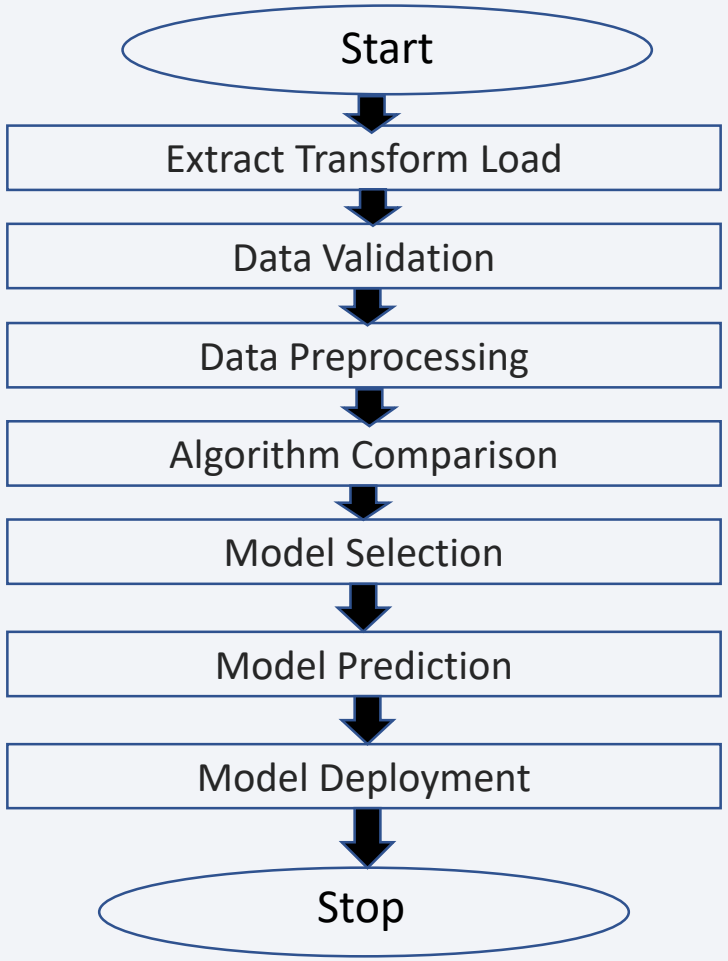
Project background and context

- ✓ In 2019, fraudulent insurance claims costed insurance companies in the United Kingdom £1.2 billion alone. Today, we see a rise in insurance fraud particularly in the property, automotive and healthcare sectors.
- ✓ With Accenture naming insurance as the industry most susceptible to future disruption, insurance companies need to adopt digital innovations urgently to reduce instances of fraudulent claims and better prepare for future threats.

Problems that need to be solved?

- ✓ How can AI, and machine learning more specifically, help your organization detect insurance fraud more effectively?.

Methodology



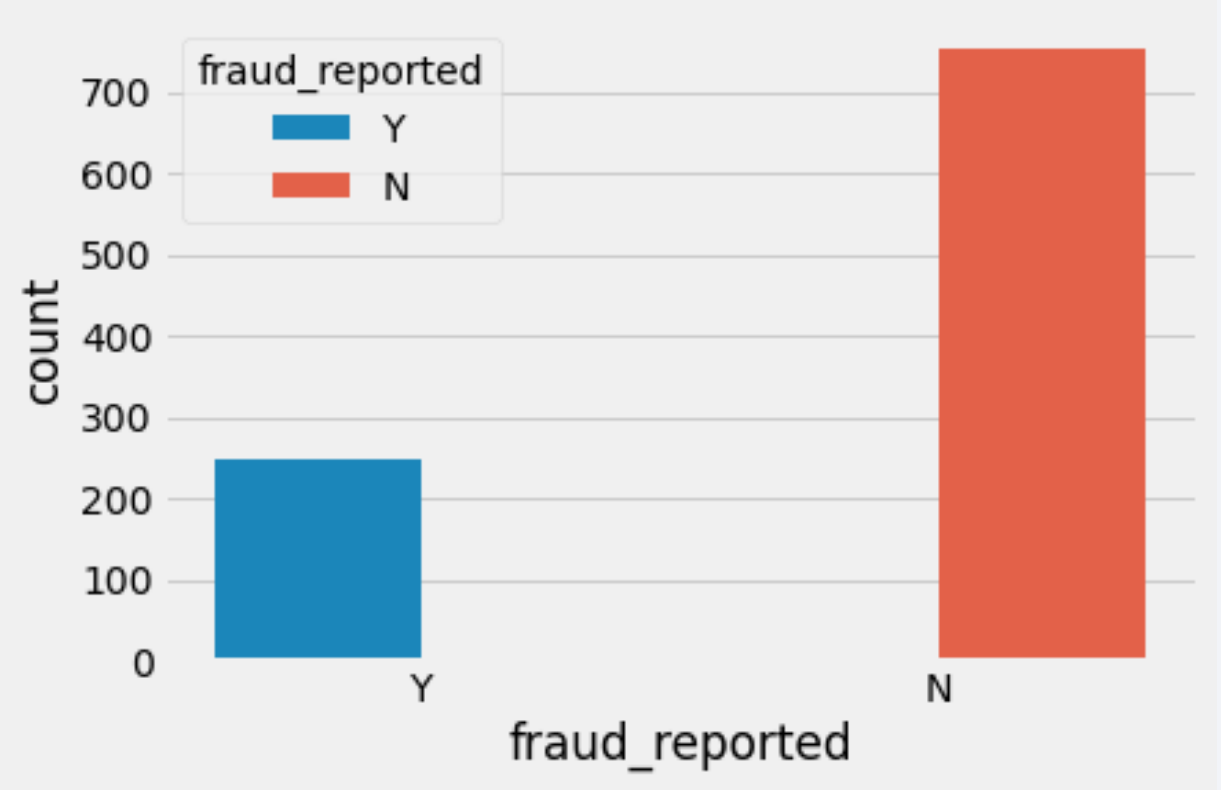
- ✓ Collected Data “insurance_claims.csv“ From Kaggle dataset “
- ✓ Classification is performed using the XGBoost (eXtreme Gradient Boosting) algorithm.
- ✓ Make prediction for the test data and evaluate
- ✓ The XGB model provide improved performance @82.5%

Insights drawn from EDA Analysis

Insights drawn from EDA Analysis

Exploratory data analysis

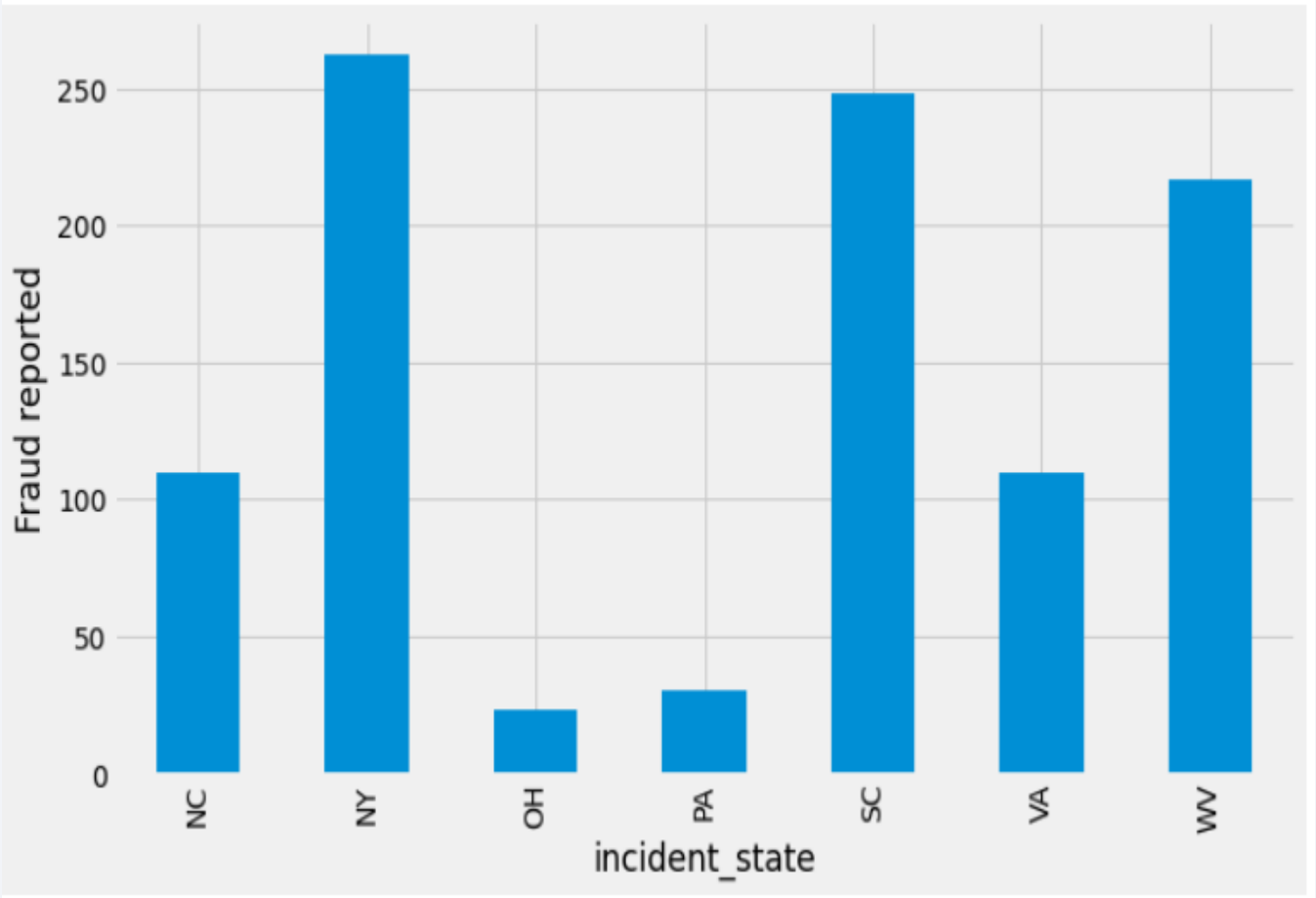
Exploratory data analysis was conducted starting with the dependent variable, Fraud reported. There were 247 frauds and 753 non-frauds. 24.7% of the data were frauds while 75.3% were non-fraudulent claims.



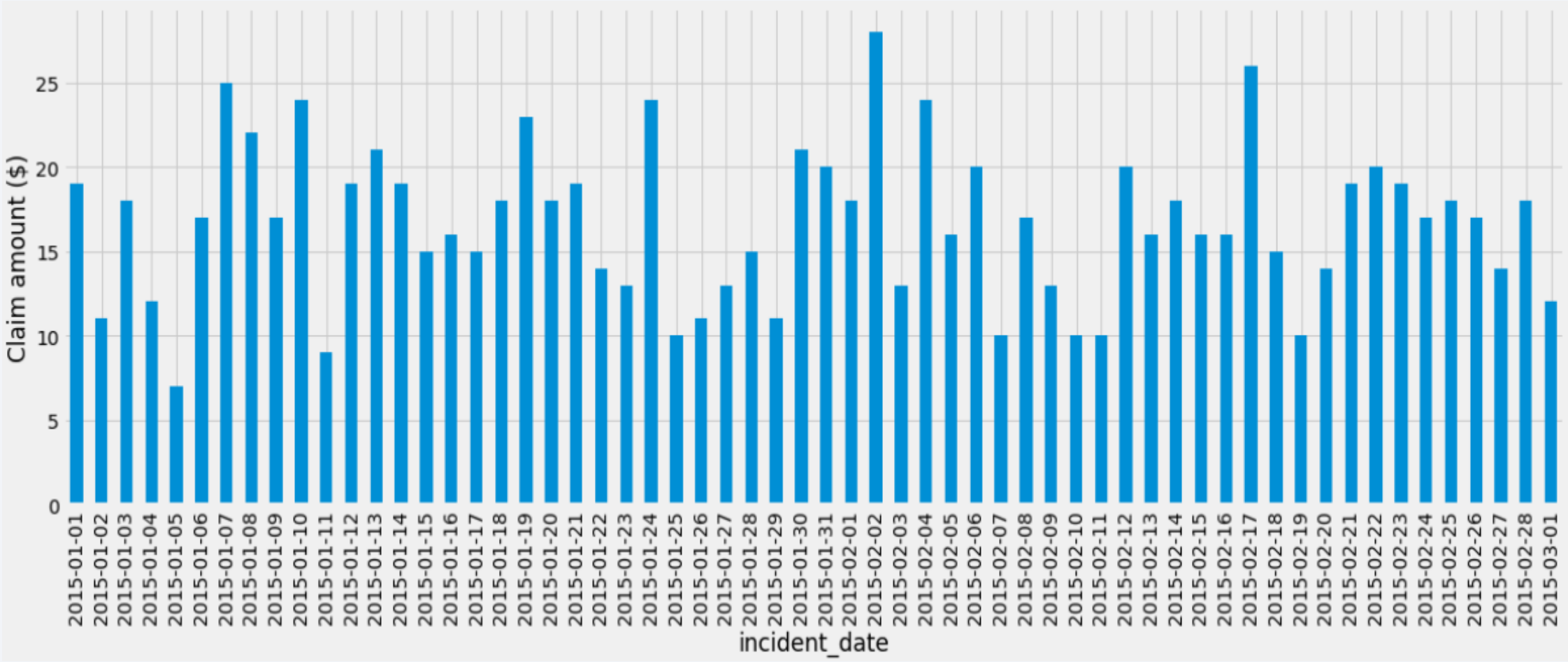
Insights drawn from EDA Analysis

State wise Claim Reported

NY	262
SC	248
WV	217
VA	110
NC	110
PA	30
OH	23

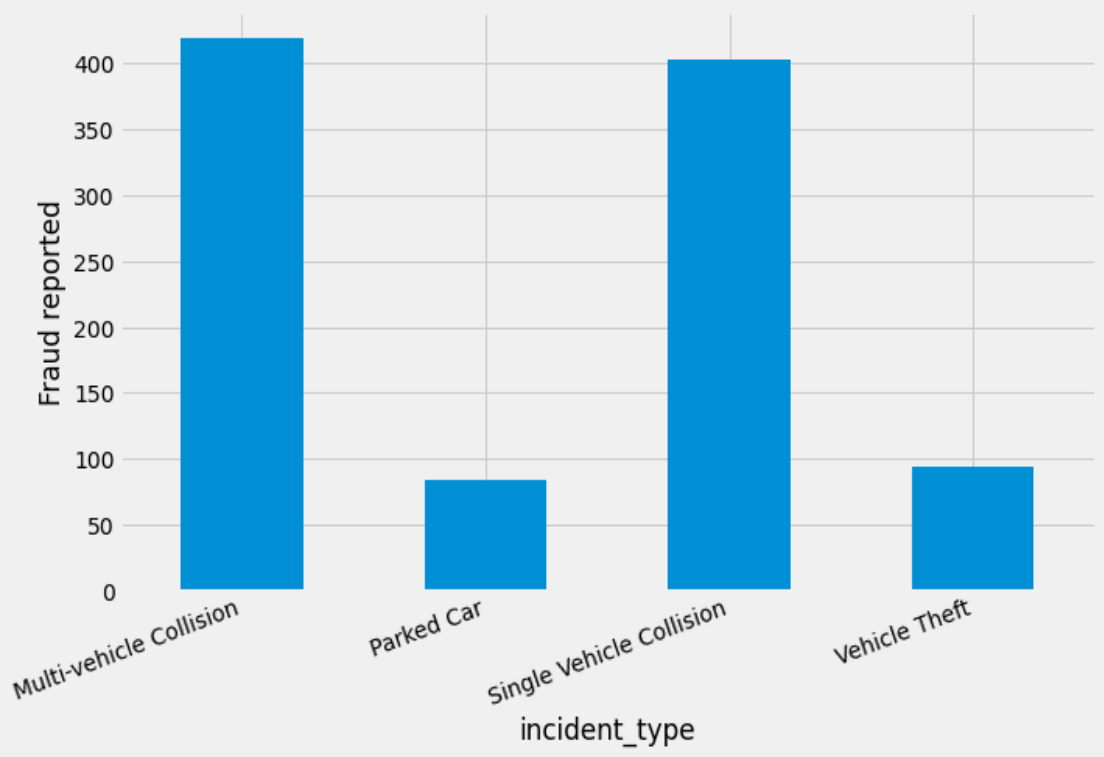


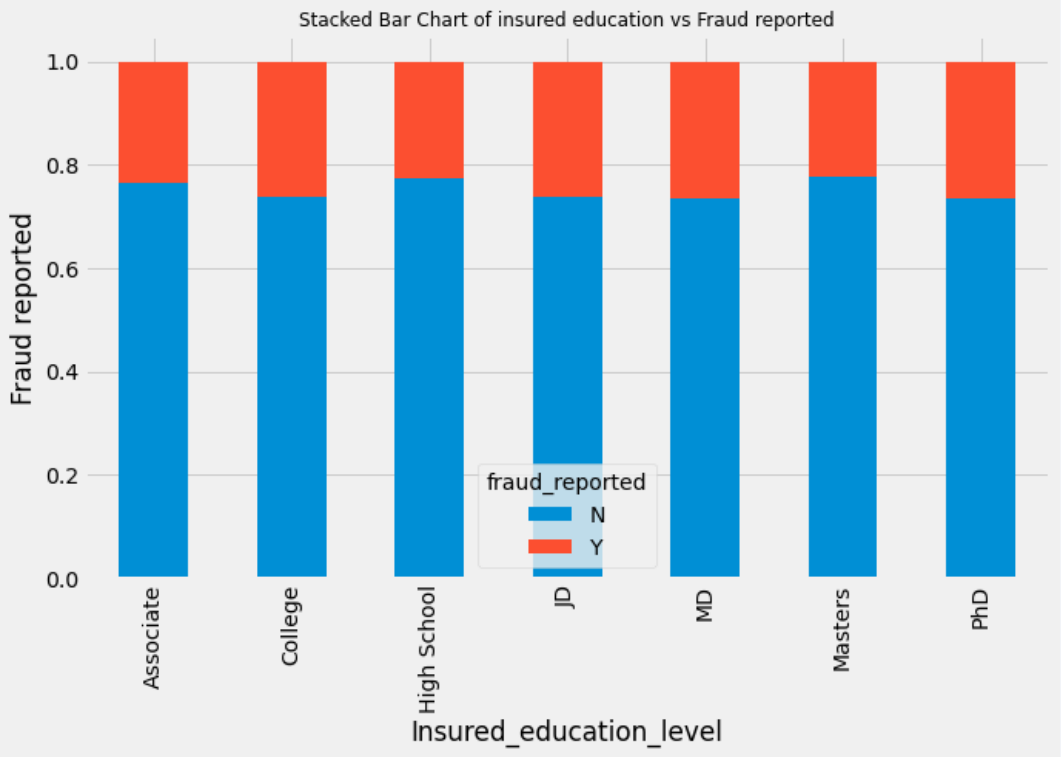
Insights drawn from EDA Analysis



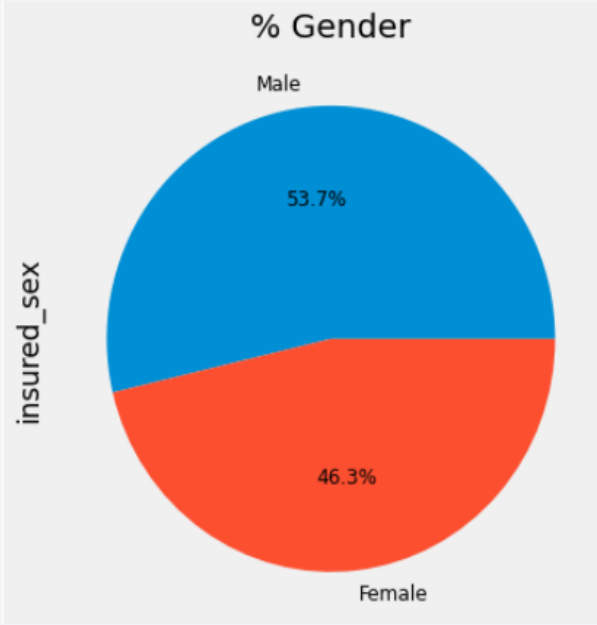
Date wise insurance claim reported

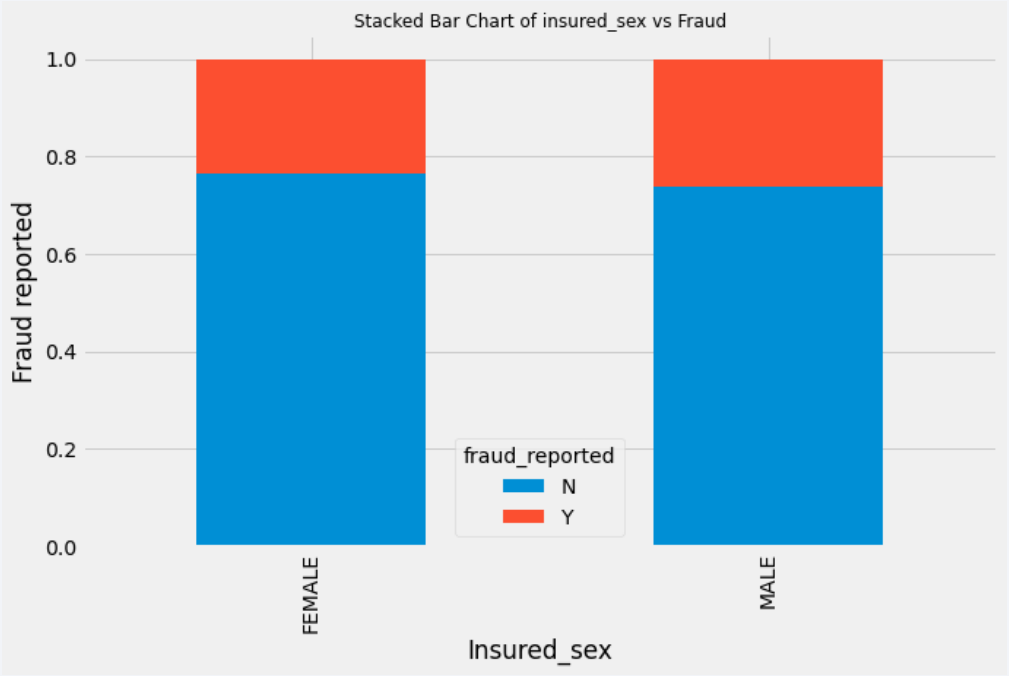
Insights drawn from EDA Analysis





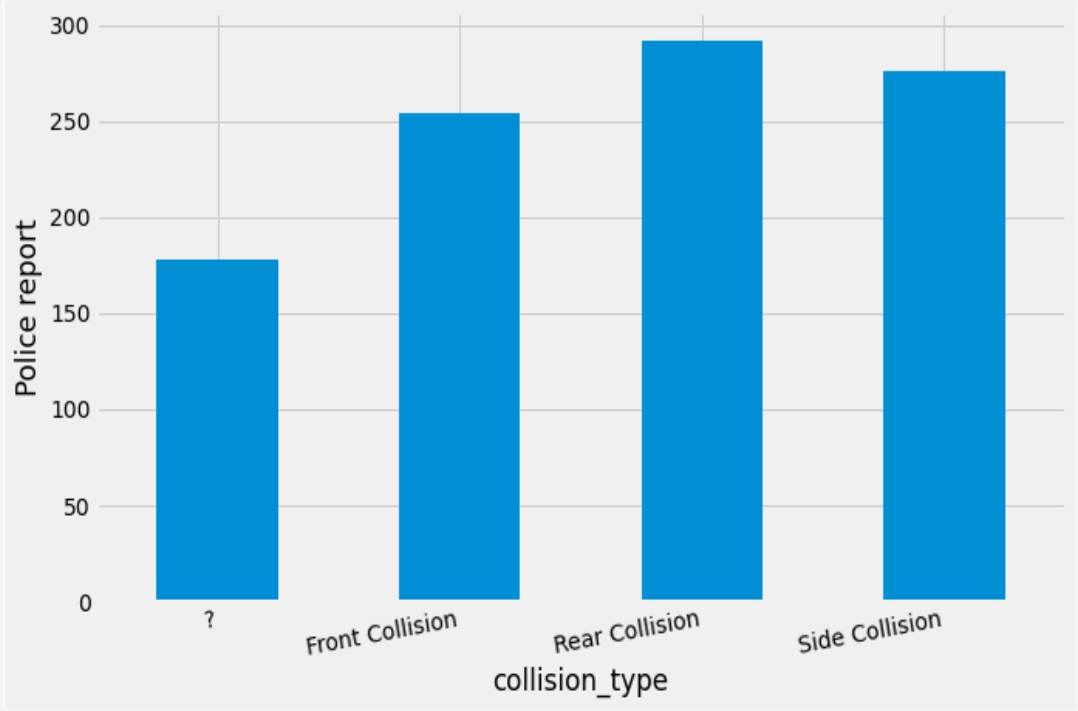
Insights drawn from EDA Analysis

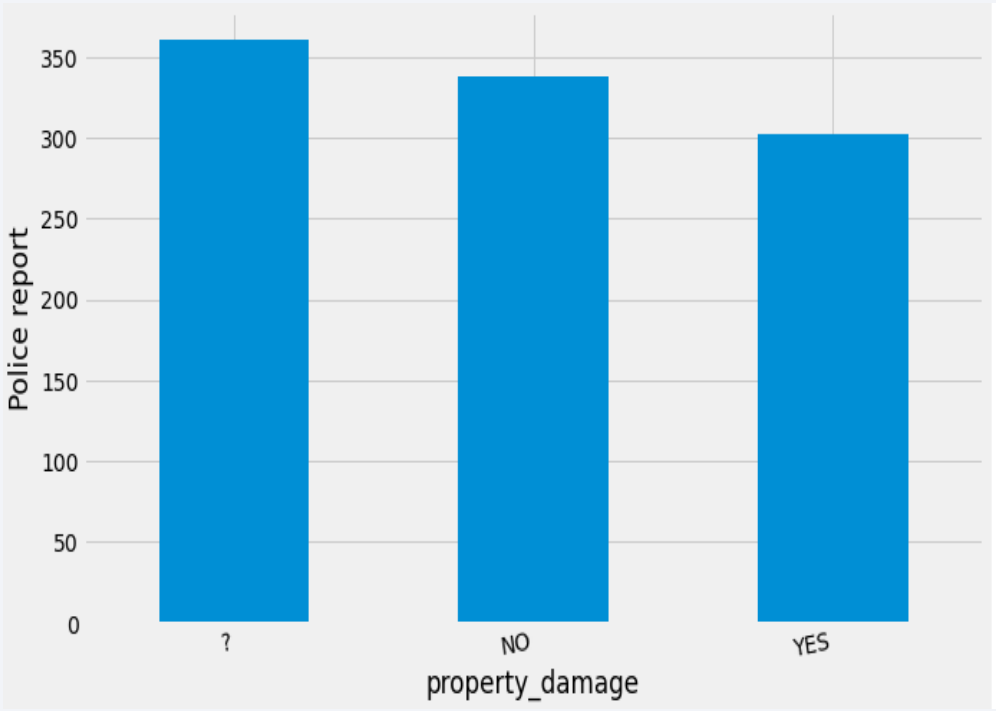




Gender wise Claim

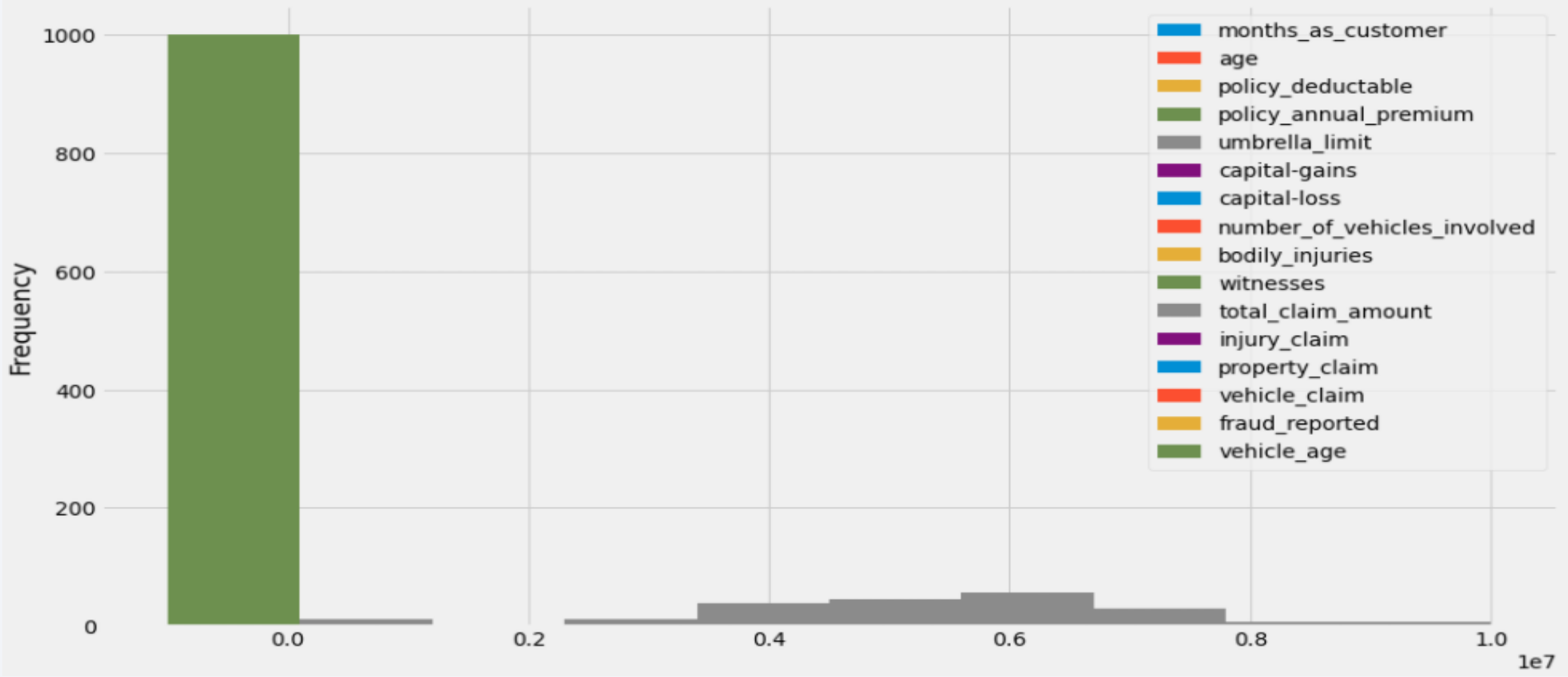
Insights drawn from EDA Analysis





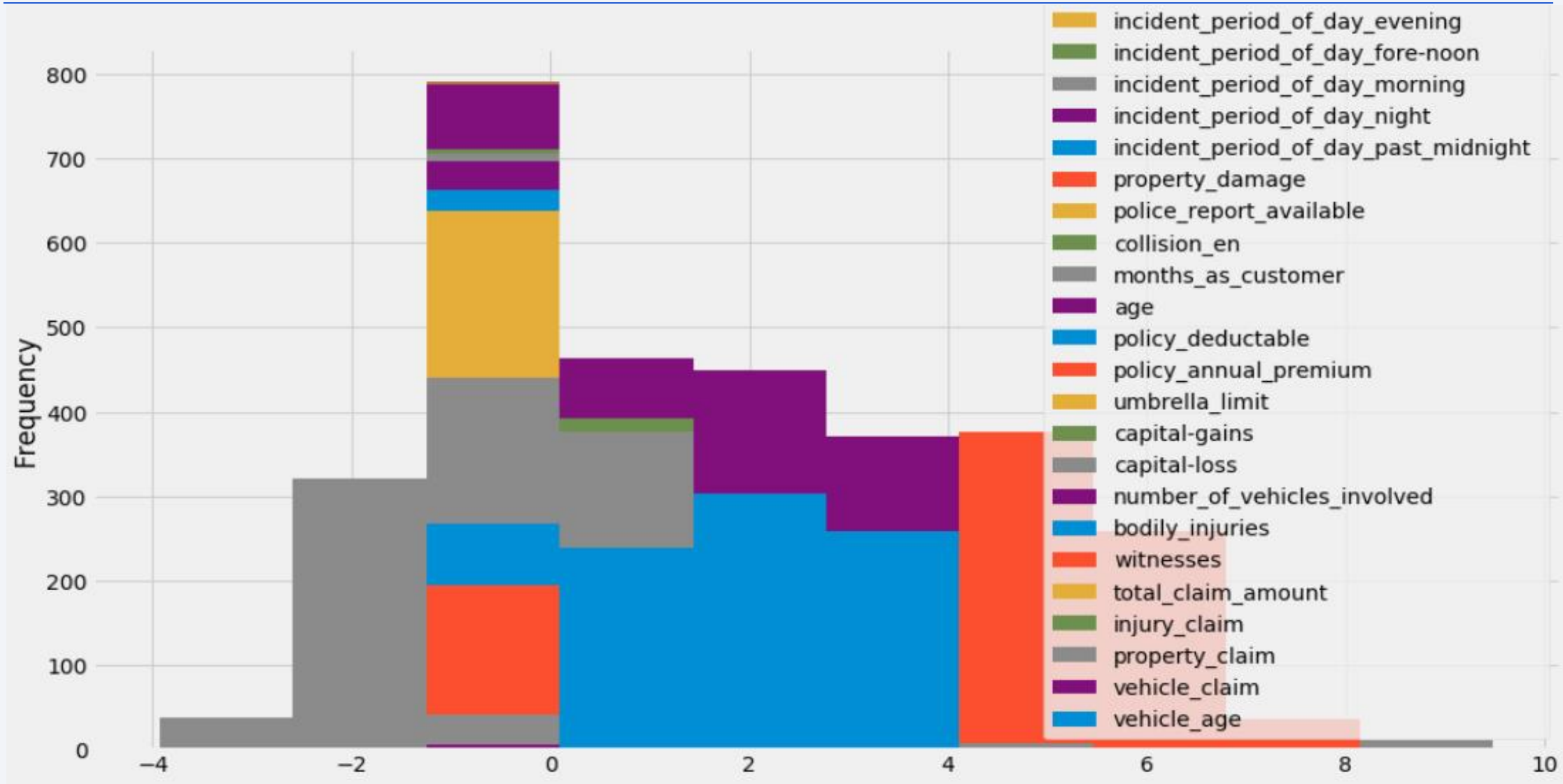
Collision type and total damage report

Anomaly Detection

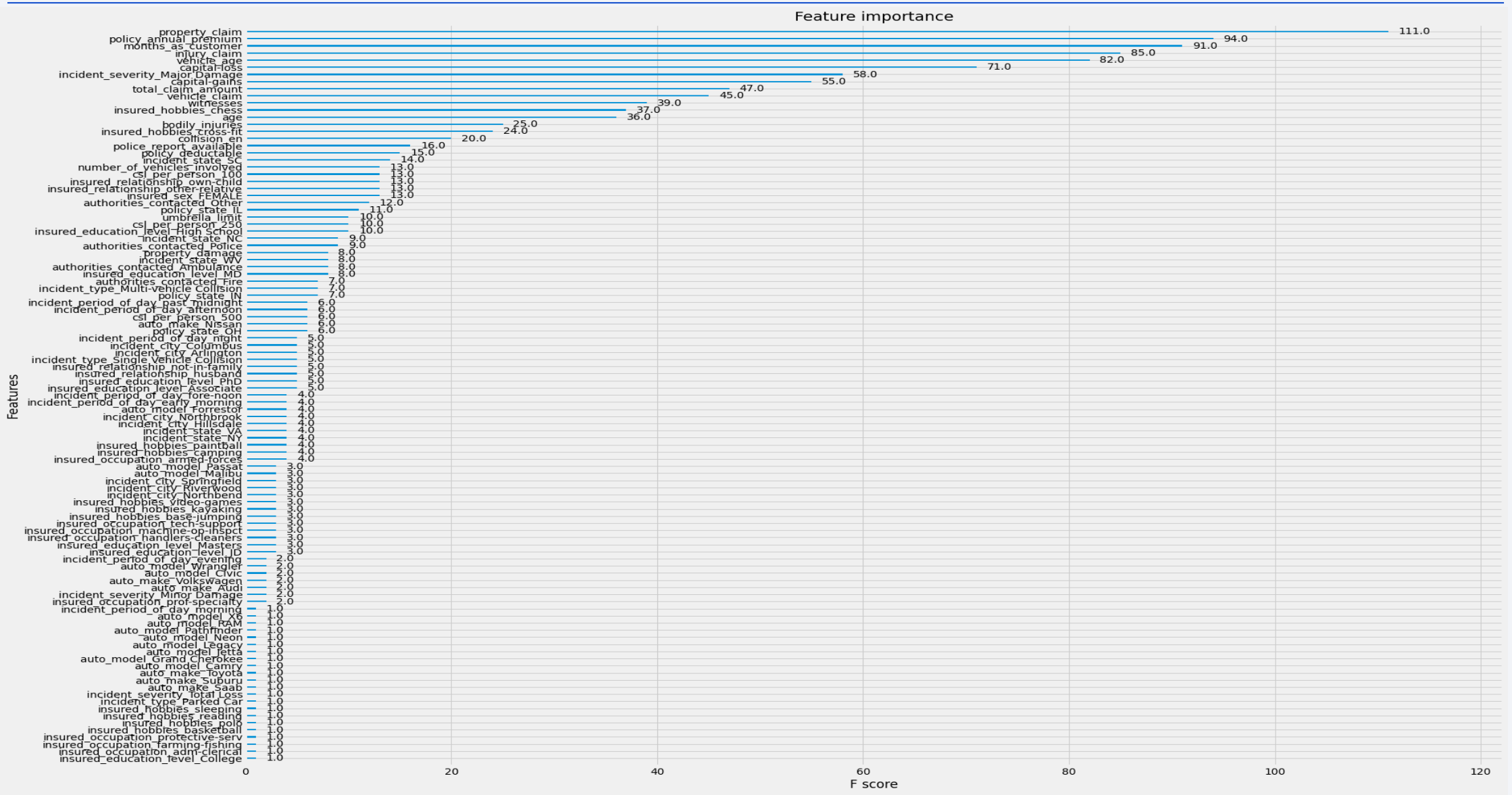


The green bar standing tall and away from all, signifies anomalies in either of policy_annual_premium, witnesses or vehicle_age. Let's draw box-and-whisker plot on each to check the presence of outliers

Histogram Plot on scaled data to check Anolalies

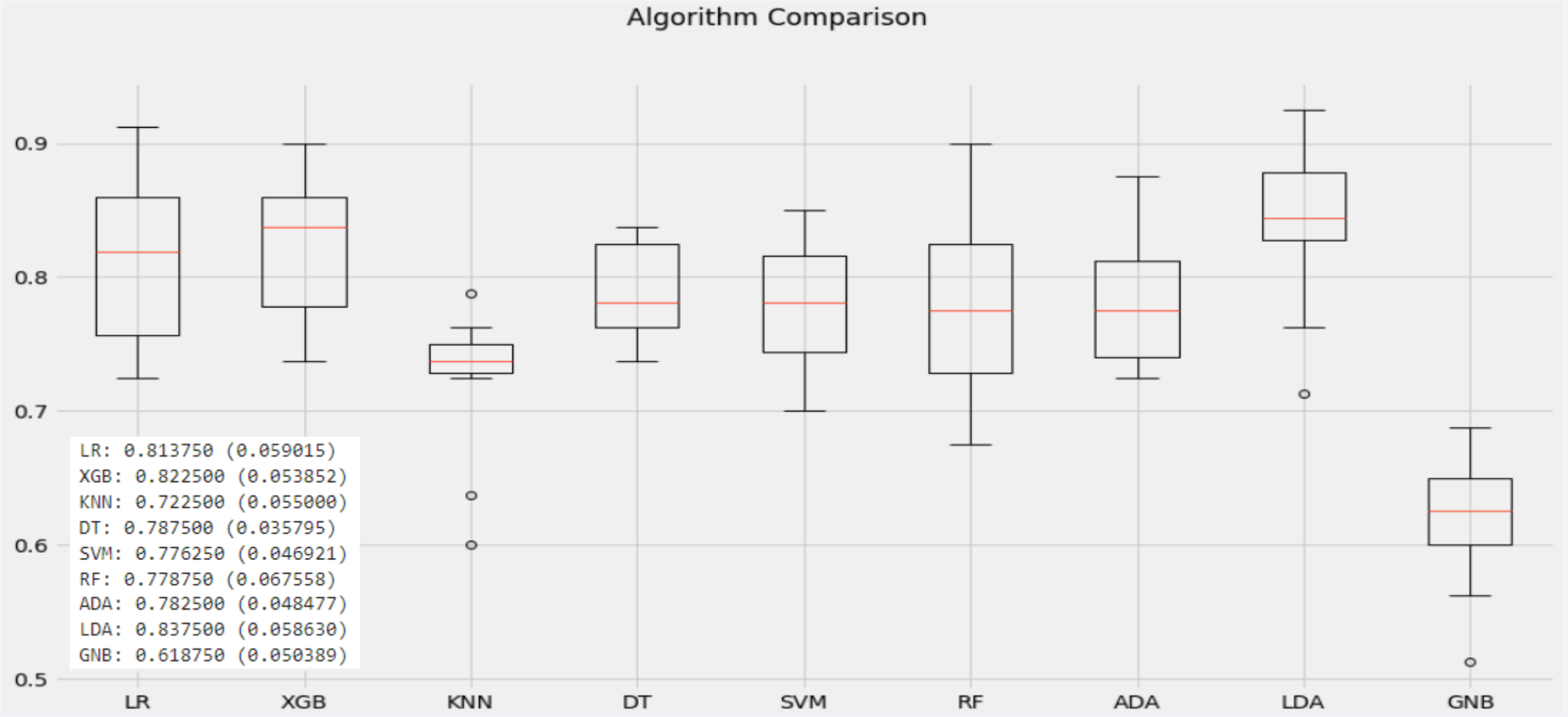


Fitting the model on DataFrame to identify the Feature Names

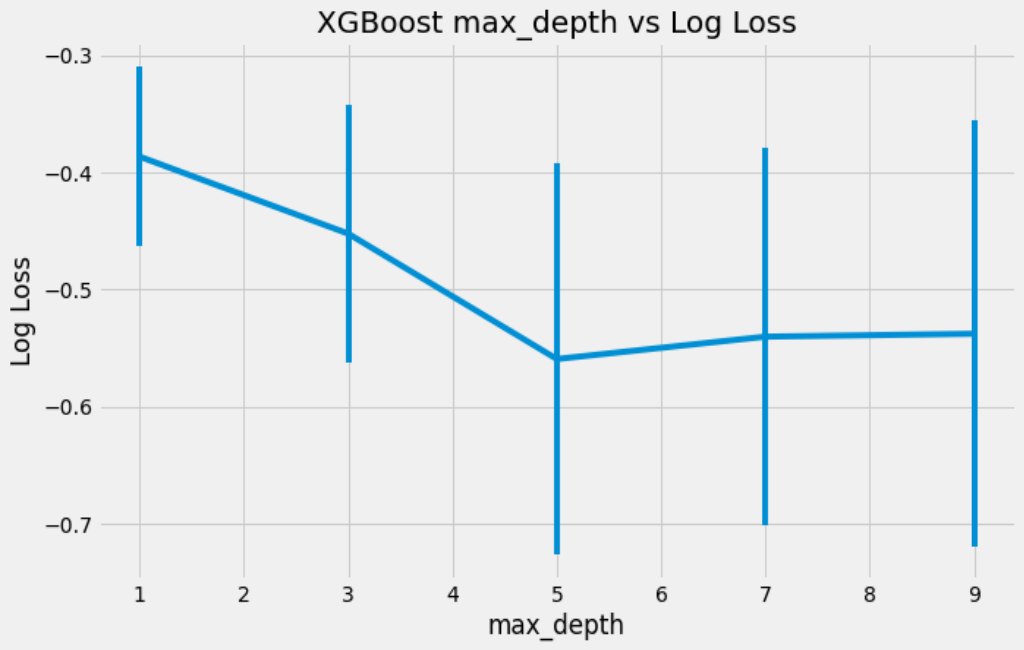



Predictive
Analysis

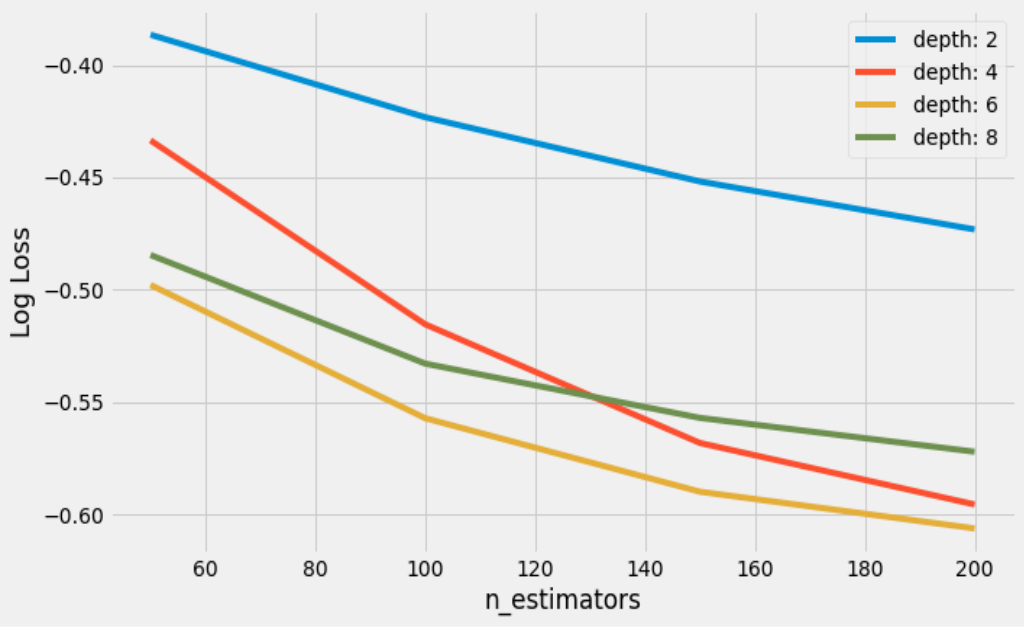
Model Selection



Model Fitting



We can see the log loss for each max_depth. The optimal configuration is max_depth=1 resulting in a log loss of 0.3865.



We see here that, the best result was achieved with a n_estimators=50 and max_depth=2

Model Prediction for Test Data

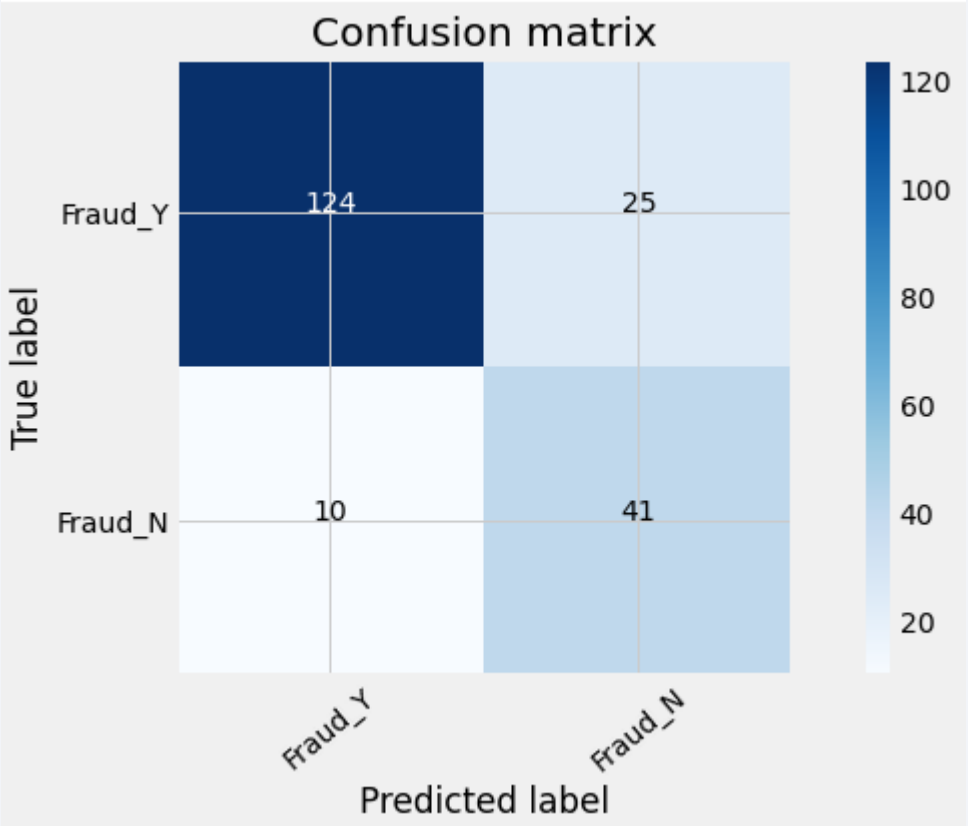
The XGB model provides improved performance @ 82.5% with fitted model (which means 1 in every 18 fraud reported are incorrect).

```
Accuracy:  82.5
Cohen Kappa: 0.58
Recall:  80.39

 Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.83      0.88       149
           1       0.62      0.80      0.70        51

    accuracy                           0.82       200
   macro avg       0.77      0.82      0.79       200
weighted avg       0.85      0.82      0.83       200

0.836
```

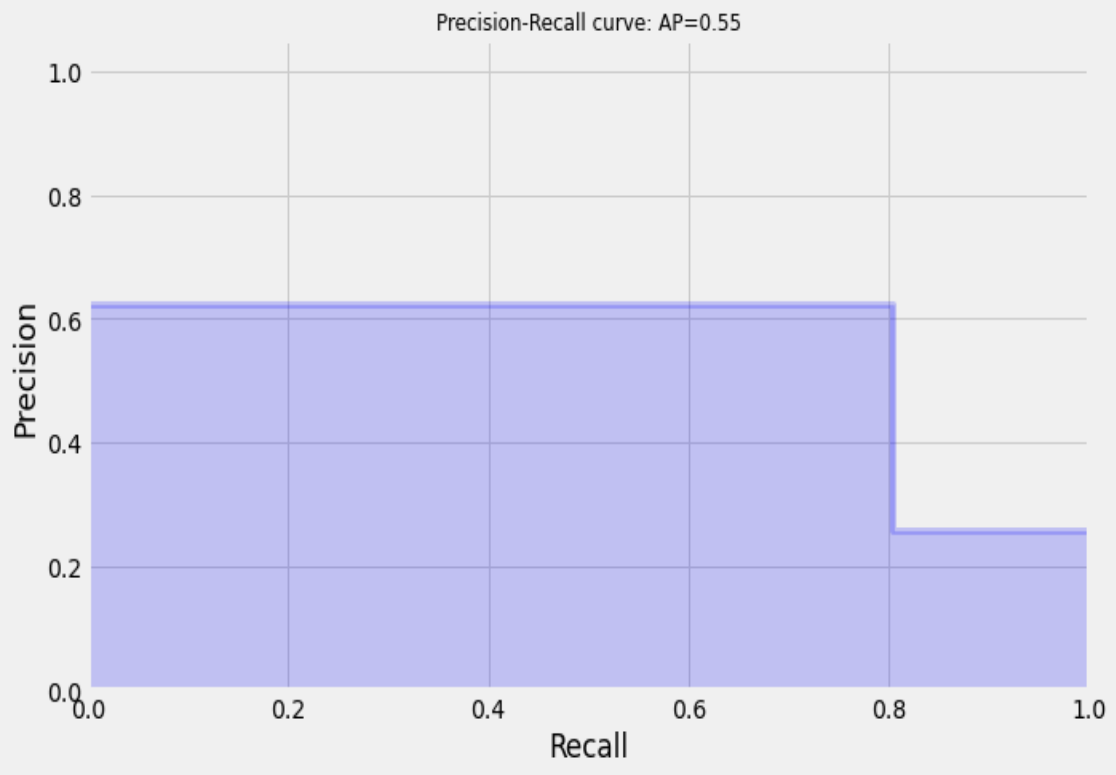
Confusion Matrix

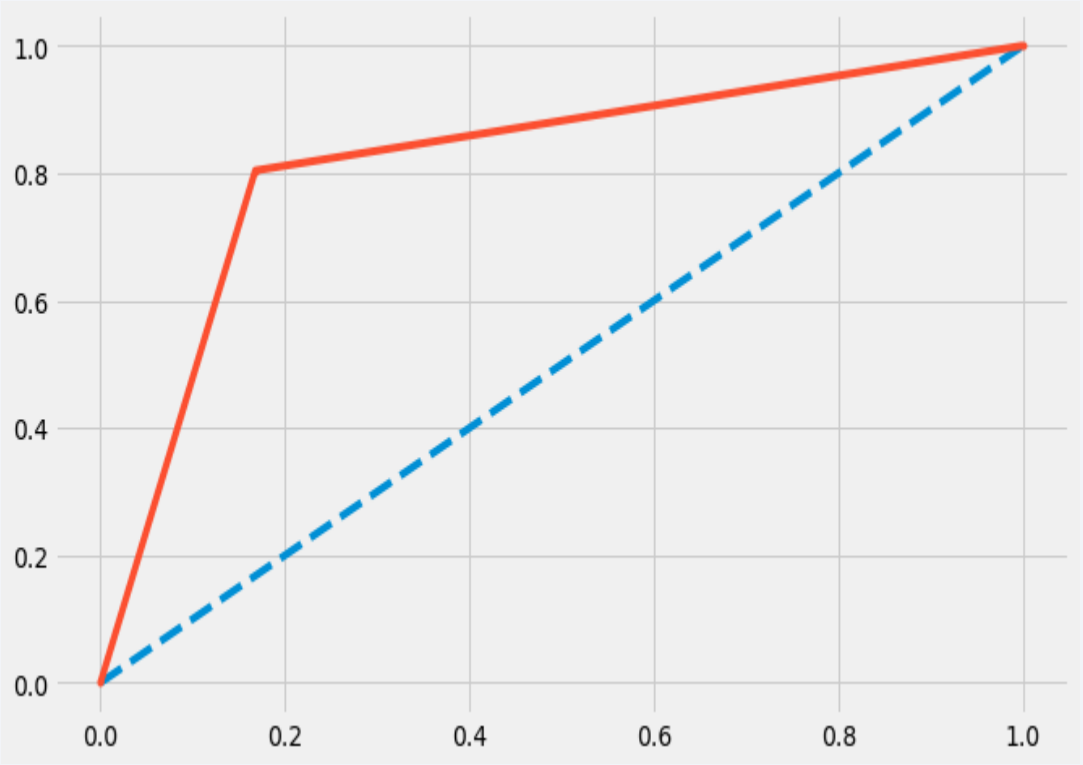
- ✓ 124 transactions were classified as valid that were actually valid
- ✓ 10 transactions were classified as fraud that were actually valid (*type 1 error*)
- ✓ 25 transactions were classified as valid that were fraud (*type 2 error*)
- ✓ 41 transactions were classified as fraud that were actually fraud.
- ✓ Err = {(25+10) / (124+10+25+41)}*100 = 17.5%

Algorithm misclassified 17.5% fraudulent transactions.



Model Performance





The plot of the ROC Curve confirms the AUC interpretation of a skillful model for most probability thresholds.

Conclusions

- ✓LR and LDA are good enough for both Feature Selection as well as Model Selection
- ✓From Voting Classifier, Logistic Regression is best with the Accuracy Score 83%
- ✓XGB Model provide the improved performance of 82.5% with fitted model

Appendix

- ✓GitHub Repository URL:
  https://github.com/veer2701/github_projects/blob/main/Advanced%20Data%20Science%20with%20IBM%20Specialization/INSURANCE%20FRAUD%20DETECTION%20USING%20MACHINE%20LEARNING-OK.ipynb
- ✓Kaggle Dataset: “insurance_claims.csv”
- ✓Instructors:

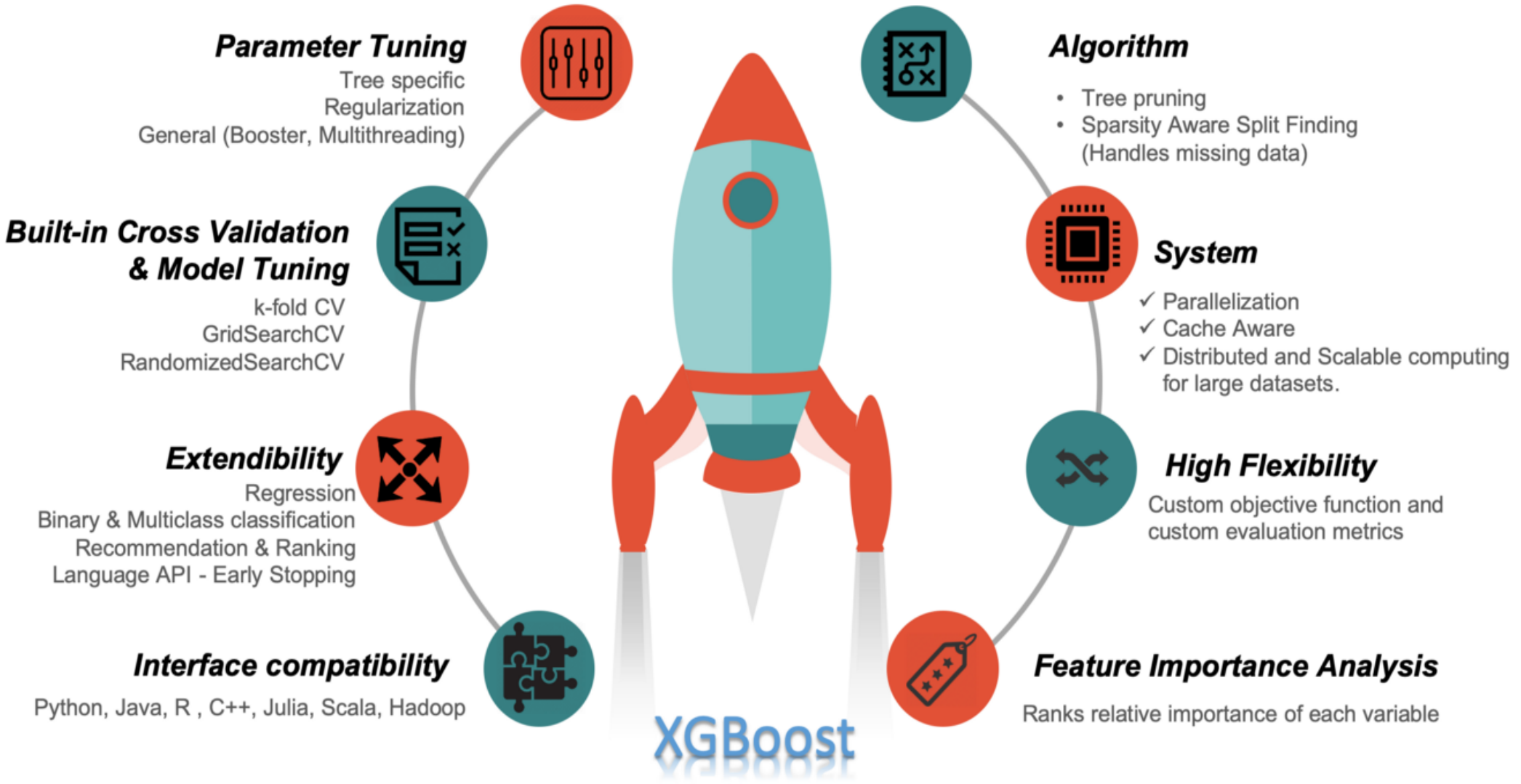
  Romeo Kienzler, Niketan Pansare, Max Pumperla,

  Nikolay Manchev, Tom Hanlon, Ilja Rasin

Appendix

- **References**
- XGBClassifier
- Hyper-parameter Tuning
- SMOTE
- Getting started with XGBoost
- GridSearchCV & Optimisation:
- **Book:** Python Machine Learning by Sebastian Raschka and Vahid Mirjalili
- **Book:** An introduction to variable and feature selection by Isabelle Guyon

Appendix


Parameter Tuning
Tree specific
Regularization
General (Booster, Multithreading)
Built-in Cross Validation & Model Tuning
k-fold CV
GridSearchCV
RandomizedSearchCV
Extendibility
Regression
Binary & Multiclass classification
Recommendation & Ranking
Language API - Early Stopping
Interface compatibility
Python, Java, R , C++, Julia, Scala, Hadoop
Algorithm
• Tree pruning
• Sparsity Aware Split Finding (Handles missing data)
System
✓ Parallelization
✓ Cache Aware
✓ Distributed and Scalable computing for large datasets.
High Flexibility
Custom objective function and custom evaluation metrics
Feature Importance Analysis
Ranks relative importance of each variable
XGBoost


Thank You