Final Project Report

[Manish Patel April- 2024]

Vehicle Insurance Claim Fraud Detection

Problem Statement :

Auto Insurance fraud is a major issue in the auto insurance industry. It is essential to detect and prevent fraud to ensure the industry's integrity and fair treatment of policyholders. In the United States alone, it is estimated that around 8,898 cars were intentionally set on fire in 2020. Car insurance scams cost companies around $29 billion every year, causing significant damage to the industry.

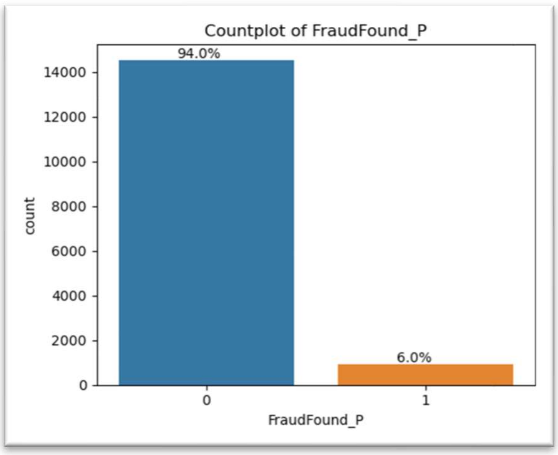
Apparently, it's a real database of an American insurance company. Vehicle insurance fraud involves conspiring to make false or fabricated claims involving property damage or personal injuries following an accident. Our objective is to classify claims as fraudulent or genuine.

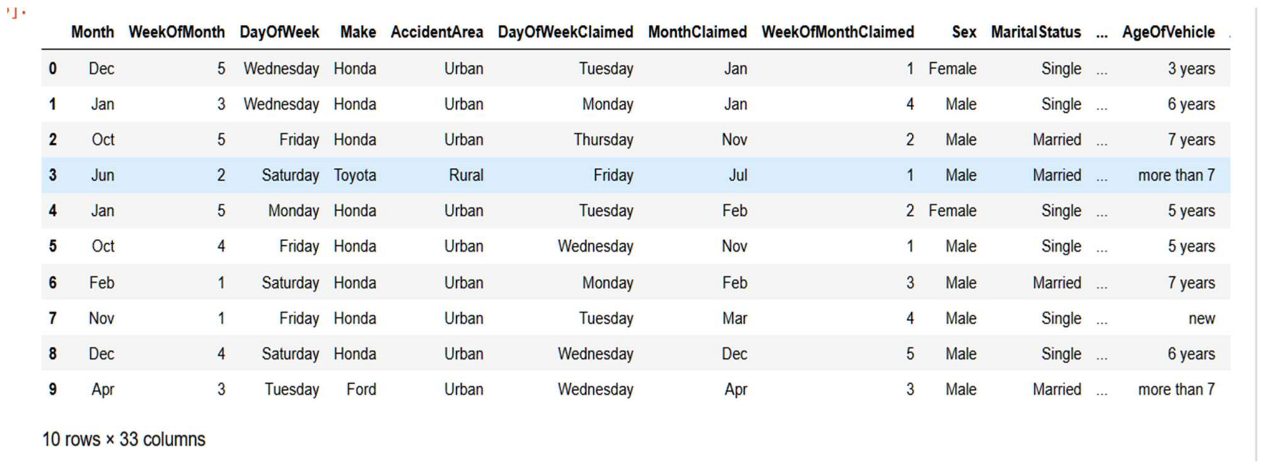
Some common examples include staged accidents where fraudsters purposely “arrange” for accidents to occur; the use of phantom passengers where people who were not even at the scene of the accident claim to have suffered harmful injury, and make false personal injury claims where personal injuries are significantly overstated.

Data Wrangling :

The dataset we used 'the auto insurance fraud' is available on Kaggle [Dataset.csv], which contains 15420 insurance claims with 33 features for each claim, with a mix of both fraudulent and legitimate claims. The data contains various features such as the insured amount, age of the driver, incident month incident location, policyholder information, and more. The 'FraudFound_P ' column was used for the target variable, as it indicates whether the claim was fraudulent or legitimate with "0" and "1" respectively.

Initially, the domain of each feature was established to classify if it was a categorical or a numerical variable. The data set has 24 categorical and 9 numerical variables and it has no missing values, we also found three features have "0" values in 320 rows technically it's Nan which is 2%, therefore we removed it, Also 'AgeOfPolicyHolder' and 'Age' variable found same so drop one of them, Since three two variable having 'none' string as its value, so in the further investigation found that it makes sense to replace it with "0" in two variable called 'NumberOfSuppliments' & 'PastNumberofClaims' While 'Days_Policy_Accident' having 55 'none' values so it was removed. Also, we found that our target variable has 94.13% of '0' values and 5.86% of '1' values so we have an unbalanced dataset. A feature called AddressChange_Claim has a value of 'no change' so that also needs to change to '0'.





	Month	WeekOfMonth	DayOfWeek	Make	AccidentArea	DayOfWeekClaimed	MonthClaimed	WeekOfMonthClaimed	Sex	MaritalStatus	...	AgeOfVehicle
0	Dec	5	Wednesday	Honda	Urban	Tuesday	Jan	1	Female	Single	...	3 years
1	Jan	3	Wednesday	Honda	Urban	Monday	Jan	4	Male	Single	...	6 years
2	Oct	5	Friday	Honda	Urban	Thursday	Nov	2	Male	Married	...	7 years
3	Jun	2	Saturday	Toyota	Rural	Friday	Jul	1	Male	Married	...	more than 7
4	Jan	5	Monday	Honda	Urban	Tuesday	Feb	2	Female	Single	...	5 years
5	Oct	4	Friday	Honda	Urban	Wednesday	Nov	1	Male	Single	...	5 years
6	Feb	1	Saturday	Honda	Urban	Monday	Feb	3	Male	Married	...	7 years
7	Nov	1	Friday	Honda	Urban	Tuesday	Mar	4	Male	Single	...	new
8	Dec	4	Saturday	Honda	Urban	Wednesday	Dec	5	Male	Single	...	6 years
9	Apr	3	Tuesday	Ford	Urban	Wednesday	Apr	3	Male	Married	...	more than 7

10 rows × 33 columns

The data set has 8 features that need to convert from object to integer The 8 features are:

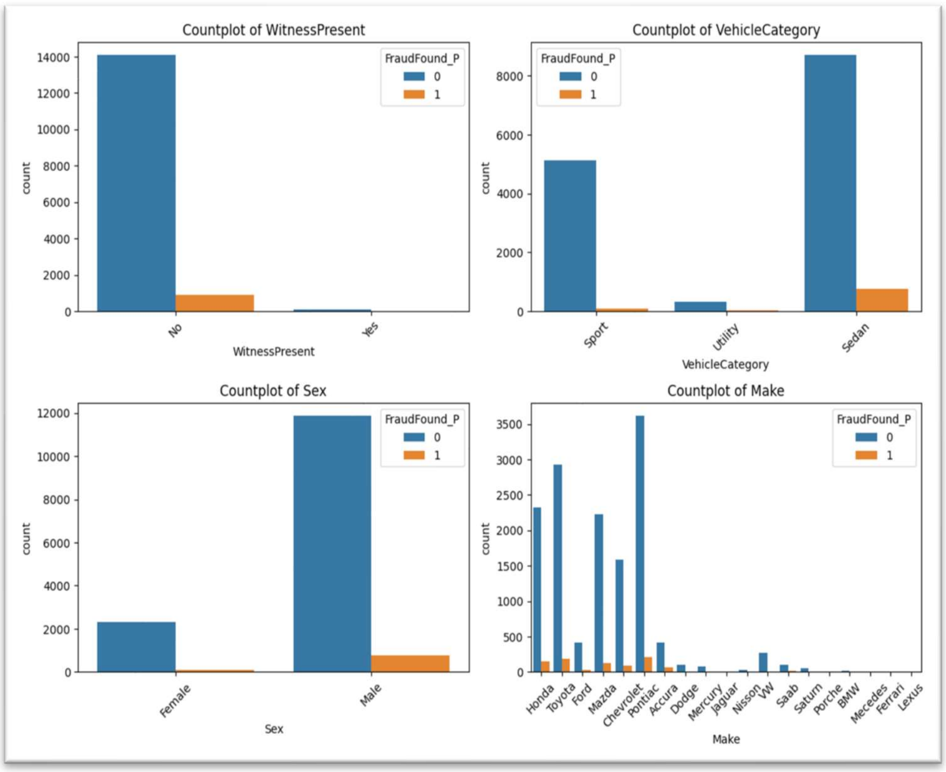
- ‘NumberOfCars',
- 'NumberOfSuppliments',
- 'PastNumberOfClaims',
- 'Days_Policy_Claim',
- 'Days_Policy_Accident',
- 'VehiclePrice',
- 'AgeOfVehicle',
- 'AddressChange_Claim'.

After doing the wrangling process size of the data set is 32 features and 15045 rows.

Exploratory Data Analysis:-

In EDA the findings are to be observed guardedly and carefully so that proper interpretation can be made. The trends in the spread of data and correlation between variables give good insights for making suitable changes in the data parameters.
The dataset has 15 numerical, and 17 categorical variables.
The graph below displays the count plot for 'WitnessPresent', 'VehicleCategory', 'Sex', and 'Make'.
(0 – Fraud not found , 1 – Fraud found)

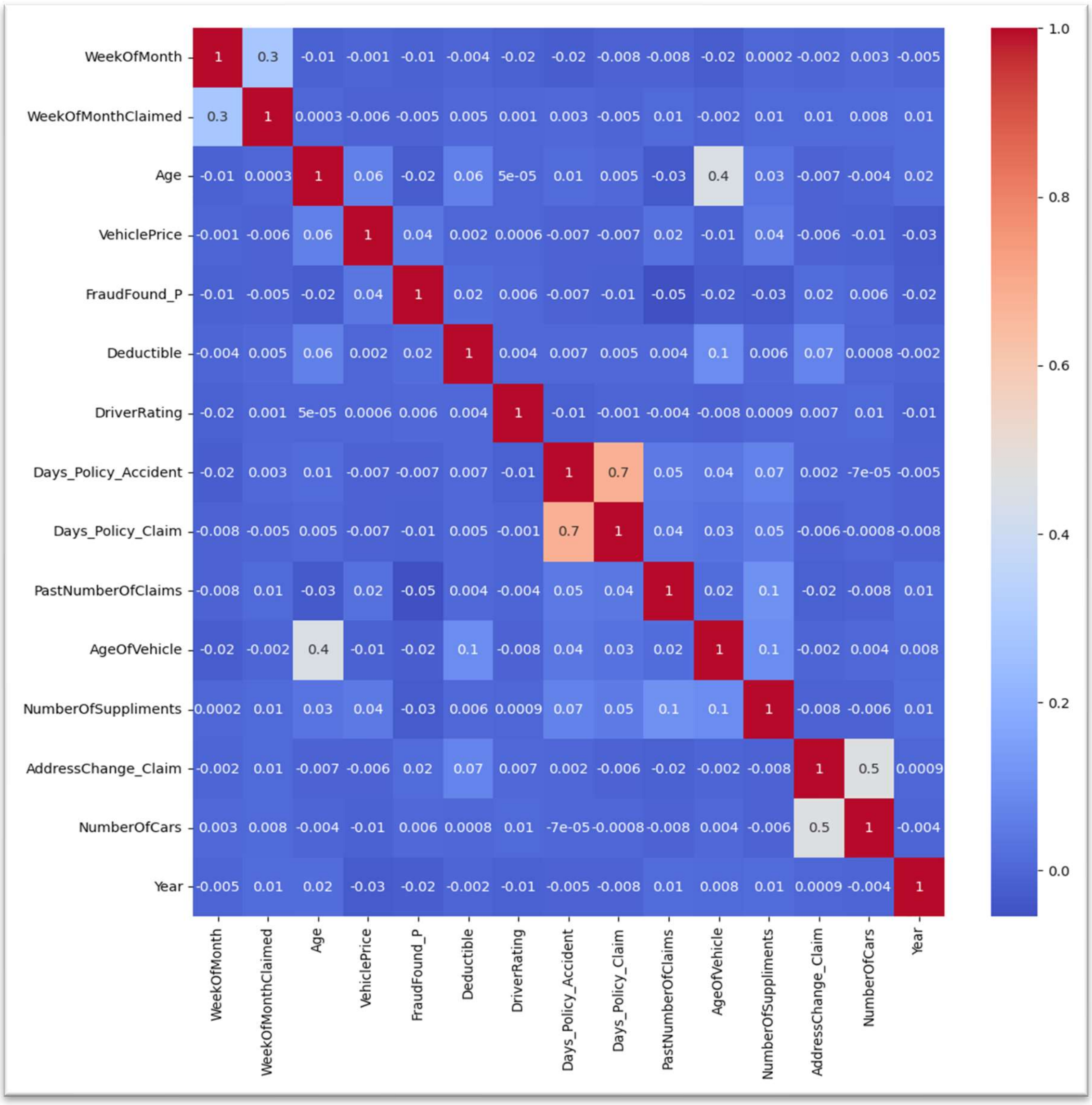
- The WitnessPresent graph indicates that in most claims, witnesses were not present.
- In the VehicleCategory plot, it is evident that sedan is the category with the highest number of fraudulent claims.
- Regarding Sex, the graph shows that men make up the majority of the claimants and that men who make false claims are more common.
- Lastly, among 19 car brands, it is observed that only 5 of them have more claims while the others have negligible claims.

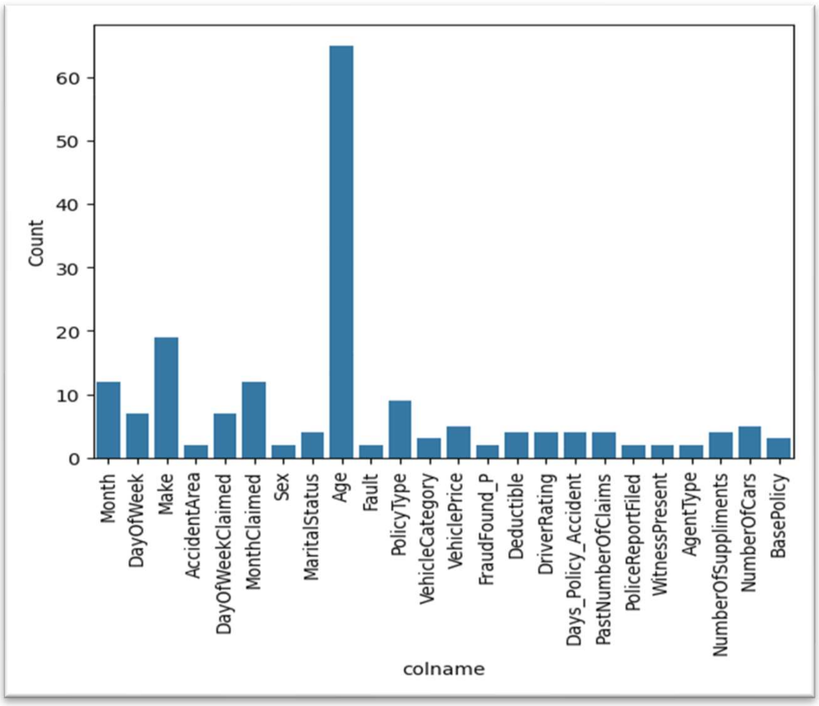


Feature Selection and Transformations

We utilized a heatmap to identify which features are correlated with each other. This helped us select relevant features and minimize redundant data. For numerical features, we set the range of the test from 0 to 1 to determine positive relationships. Based on the heatmap for numerical features, we discovered that certain features were highly correlated. Consequently, we removed one of these features from the data frame.

‘AddressChange_Claim and ‘NumberOfCars’
‘WeekOfMonth’ and ‘WeekOfMonthClaimed’
‘AgeOfVehicle’ and ‘Age’
Days_Policy_Accident and ‘Days_Policy_Claim’



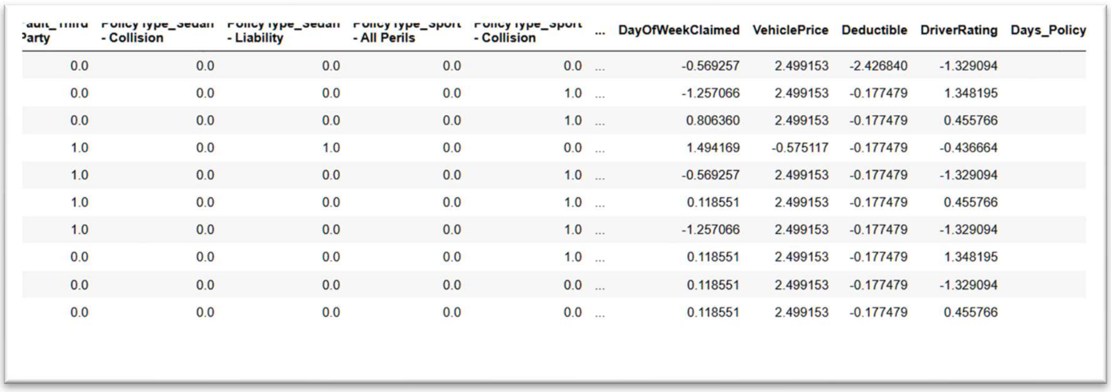


The above figure indicates that the ‘Age’ variable possesses a wide range of unique values. To simplify the analysis and potentially reveal patterns more effectively, it’s recommended to categorize these values into 10 distinct groups, or “bins.” These bins will be collectively referred to as ‘Age_bins’. This binning process involves grouping continuous data into intervals, which can then be analyzed as categorical data which will be named 'Age_bins'.

For the other variables mentioned—‘Month’, ‘MonthClaimed’, ‘DayOfWeekClaimed’, ‘DayofWeek’, and ‘Make’—since they each have fewer than 20 unique values, it’s suggested to treat them as discrete variables. Discrete variables are those that can only take on a finite number of values. In this context, treating these variables as discrete means that each unique value will be considered a separate category within the variable. This allows for a more straightforward analysis of the data, as each category can be counted and analyzed individually.

Scaler and Normalization

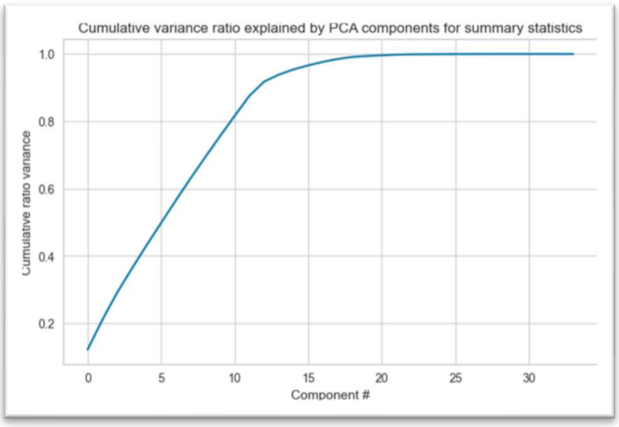
To prepare the data for analysis, two common techniques were employed - one hot encoding and standard scaling. One hot encoding is a method used to convert categorical variables into numerical format so they can be easily understood by machine learning algorithms. Standard scaling, on the other hand, is a normalization technique that rescales the data to have a mean of zero and a standard deviation of one. After applying both techniques, the data was transformed into a more uniform and machine-readable format. The resulting table can be viewed below.



ault_Third Party	Policy Type_Sedan - Collision	Policy Type_Sedan - Liability	Policy Type_Sport - All Perils	Policy Type_Sport - Collision	...	DayOfWeekClaimed	VehiclePrice	Deductible	DriverRating	Days_Policy
0.0	0.0	0.0	0.0	0.0	...	-0.569257	2.499153	-2.426840	-1.329094	
0.0	0.0	0.0	0.0	1.0	...	-1.257066	2.499153	-0.177479	1.348195	
0.0	0.0	0.0	0.0	1.0	...	0.806360	2.499153	-0.177479	0.455766	
1.0	0.0	1.0	0.0	0.0	...	1.494169	-0.575117	-0.177479	-0.436664	
1.0	0.0	0.0	0.0	1.0	...	-0.569257	2.499153	-0.177479	-1.329094	
1.0	0.0	0.0	0.0	1.0	...	0.118551	2.499153	-0.177479	0.455766	
1.0	0.0	0.0	0.0	1.0	...	-1.257066	2.499153	-0.177479	-1.329094	
0.0	0.0	0.0	0.0	1.0	...	0.118551	2.499153	-0.177479	1.348195	
0.0	0.0	0.0	0.0	0.0	...	0.118551	2.499153	-0.177479	-1.329094	
0.0	0.0	0.0	0.0	0.0	...	0.118551	2.499153	-0.177479	0.455766	

PCA method for feature reduction

Our data set underwent a process called principal component analysis, which is used to reduce the dimensionality of data sets. This technique involves transforming a large set of variables into a smaller one that still contains most of the information in the original set. By doing so, we were able to simplify the data and identify the underlying patterns and relationships among the variables. This in turn helped us to better understand the data and draw more meaningful insights from it. The below graph shows the first thirteen components seem to account for over 95% of the variance. To continue with our process, we have taken into account 13 components that are relevant for our project.



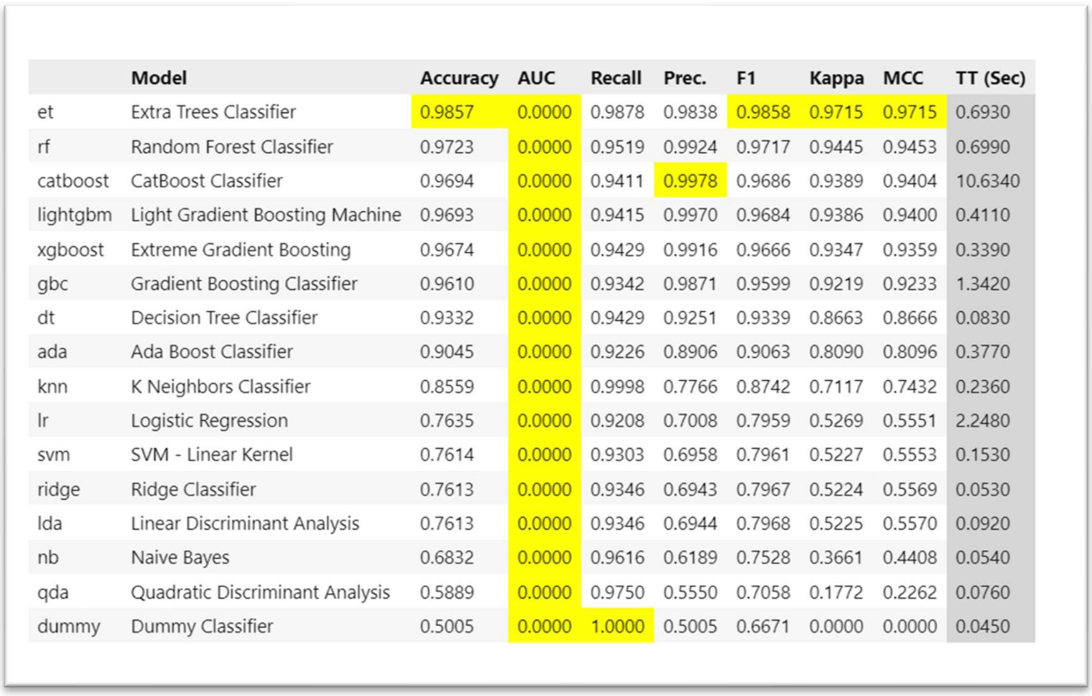
Sampling and Evaluation Settings

Due to the fact that our data set is unbalanced, we have implemented a technique called over-sampling to address the issue of underrepresented classes. Specifically, we have utilized a method called ADASYN (Adaptive Synthetic Sampling) to generate synthetic samples of the minority class in order to balance out the overall distribution of the data. By doing so, we aim to improve the performance of our classification algorithm, particularly with regard to accurately identifying and predicting outcomes for the underrepresented classes in the data set.

We created the training and testing datasets and decided to split the training data into 80% and the testing data into 20%. This gives enough data to the models for each to be trained effectively and make accurate predictions.

Model Creation:-

With the help of the Pycaret library, we can analyze and evaluate the effectiveness of various models (As seen in the table below) on our dataset. This allows us to choose the best model that fits our needs. By leveraging the power of Pycaret, we can optimize our data analysis process and improve the accuracy of our results.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.9857	0.0000	0.9878	0.9838	0.9858	0.9715	0.9715	0.6930
rf	Random Forest Classifier	0.9723	0.0000	0.9519	0.9924	0.9717	0.9445	0.9453	0.6990
catboost	CatBoost Classifier	0.9694	0.0000	0.9411	0.9978	0.9686	0.9389	0.9404	10.6340
lightgbm	Light Gradient Boosting Machine	0.9693	0.0000	0.9415	0.9970	0.9684	0.9386	0.9400	0.4110
xgboost	Extreme Gradient Boosting	0.9674	0.0000	0.9429	0.9916	0.9666	0.9347	0.9359	0.3390
gbc	Gradient Boosting Classifier	0.9610	0.0000	0.9342	0.9871	0.9599	0.9219	0.9233	1.3420
dt	Decision Tree Classifier	0.9332	0.0000	0.9429	0.9251	0.9339	0.8663	0.8666	0.0830
ada	Ada Boost Classifier	0.9045	0.0000	0.9226	0.8906	0.9063	0.8090	0.8096	0.3770
knn	K Neighbors Classifier	0.8559	0.0000	0.9998	0.7766	0.8742	0.7117	0.7432	0.2360
lr	Logistic Regression	0.7635	0.0000	0.9208	0.7008	0.7959	0.5269	0.5551	2.2480
svm	SVM - Linear Kernel	0.7614	0.0000	0.9303	0.6958	0.7961	0.5227	0.5553	0.1530
ridge	Ridge Classifier	0.7613	0.0000	0.9346	0.6943	0.7967	0.5224	0.5569	0.0530
lda	Linear Discriminant Analysis	0.7613	0.0000	0.9346	0.6944	0.7968	0.5225	0.5570	0.0920
nb	Naive Bayes	0.6832	0.0000	0.9616	0.6189	0.7528	0.3661	0.4408	0.0540
qda	Quadratic Discriminant Analysis	0.5889	0.0000	0.9750	0.5550	0.7058	0.1772	0.2262	0.0760
dummy	Dummy Classifier	0.5005	0.0000	1.0000	0.5005	0.6671	0.0000	0.0000	0.0450

Model Selection

I tested 4 different machine learning classification models:

- Extra trees classifier(ETC)
- RandomForest (RF)
- k-nearest neighbors (KNN)
- support vector machine (SVM)

During the development of models, I placed a more focus on the precision metric. This involved analyzing a vast amount of data and employing advanced algorithms to identify patterns and trends that would enable the model to make highly accurate predictions.

To assess the effectiveness of a system in identifying fraudulent claims, we can compare the number of correctly identified fraudulent claims with the total number of fraudulent claims. This comparison helps us determine the accuracy rate and precision of the system. A higher accuracy rate and precision indicate that the system is more effective at identifying fraudulent claims. Therefore, it is essential to carefully measure the accuracy and precision of the system to ensure that it can effectively detect fraudulent claims.

During our model section, we chose to first assess the Decision Tree classifier since it has been proven to effectively predict the class of never-seen data using rule-based approaches. However, during evaluations, SVM model computed low accuracy scores around 79% and held low precision and recall scores in determining fraud cases in particular.

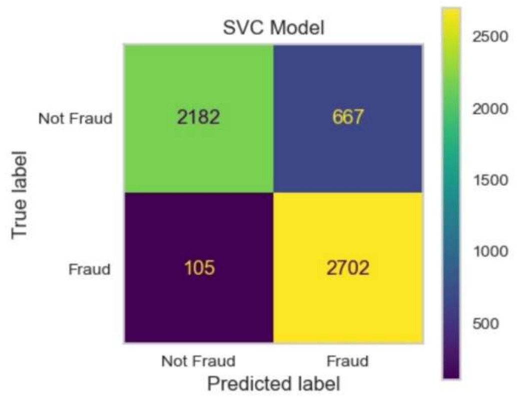
We decided to then evaluate how the k-nearest neighbor model would perform on this data. As we set the k neighbors to 5 as a base for observation, we proceeded to run the model with different values of k to improve performance metrics. We noticed how when k was 2 or lower, the predictions showed more errors in detecting fraud and more cases went unseen by the model. Yet, when k was increased to 6 or higher the model began to underfit more, in turn increasing the errors. As k-nearest neighbors also seemed to be slow, we decided to look further at other models.

We then decided to pursue other models that utilized the Random Forest classifier and Extra trees classifier. However, after evaluation, Random Forest computed accuracy scores and precision around 94% and 91%. According to the evaluation metrics, it has been observed that the Extra Trees Classifier algorithm performs with a high degree of accuracy, with an accuracy rate of approximately 96%. Additionally, the precision score of this model is also commendable, being around 95%.ETC model performed the best accuracy, precision, f1-score, and recall. Take a look at the table below, which displays the Accuracy score and precision of our models.

	Algorithm	Accuracy	Precision
3	ETC	0.963225	0.950746
2	RF	0.940594	0.915294
1	KN	0.864215	0.785175
0	SVC	0.791018	0.726134

Based on my analysis, I have found that the Extra Trees classifier and Random Forest classifier model have performed the best in comparison to the other models that were tested. However, the support vector machine model performed the worst and did not produce accurate results. To further improve the models, I conducted hyperparameter tuning for all four models using Grid Search Cross-Validation. This process helped me to fine-tune the models and improve their accuracy and precision. In order to provide a comprehensive overview of the performance of each model, I have also created a matrix for all models. This matrix provides detailed information on different performance metrics, such as accuracy and precision, for each model.

```
For   SVC
```



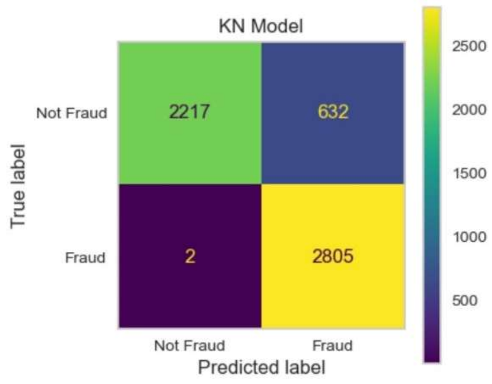
```
--------------------------------
```

```
              precision    recall  f1-score   support

           0       0.95      0.77      0.85      2849
           1       0.80      0.96      0.88      2807

    accuracy                           0.86      5656
   macro avg       0.88      0.86      0.86      5656
weighted avg       0.88      0.86      0.86      5656
```

```
------
For  KN
```

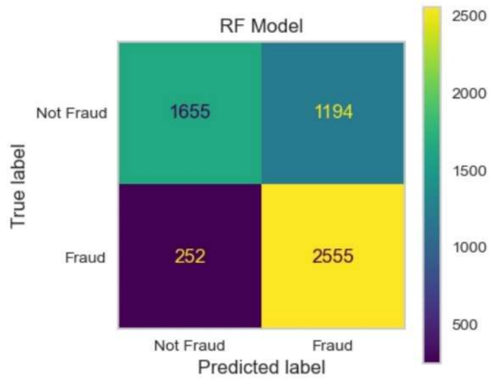


```
              precision    recall  f1-score   support

           0       1.00      0.78      0.87      2849
           1       0.82      1.00      0.90      2807

    accuracy                           0.89      5656
   macro avg       0.91      0.89      0.89      5656
weighted avg       0.91      0.89      0.89      5656
```

```
----------------------------------
------
For  RF
```

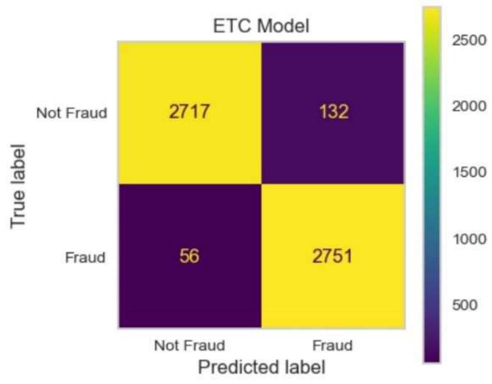


```
              precision    recall  f1-score   support

           0       0.98      0.91      0.94      2849
           1       0.92      0.98      0.95      2807

    accuracy                           0.94      5656
   macro avg       0.95      0.94      0.94      5656
weighted avg       0.95      0.94      0.94      5656
```

```
----------------------------------
------
For  ETC
```



```
              precision    recall  f1-score   support

           0       0.98      0.96      0.97      2849
           1       0.96      0.98      0.97      2807

    accuracy                           0.97      5656
   macro avg       0.97      0.97      0.97      5656
weighted avg       0.97      0.97      0.97      5656
```

```
----------------------------------
```

After a comprehensive analysis of the four candidate models, the Extra Trees Classifier emerged as the superior performer. This conclusion was drawn from a multi-faceted evaluation focusing on key performance metrics: recall, precision, and accuracy.

The Extra Trees Classifier not only achieved the highest accuracy score compared to its counterparts but also demonstrated exceptional performance in recall. Recall, or the true positive rate, is particularly crucial in the context of fraud detection as it measures the model’s ability to correctly identify all actual fraud cases. A higher recall indicates fewer false negatives, meaning fewer instances of fraud go undetected.

Precision, which assesses the model’s ability to identify only the true cases of fraud as such, was also a consideration. While precision is important to minimize false positives, in fraud detection scenarios, the cost of missing an actual fraud can be much higher than the cost of investigating a false alert. Therefore, the balance tipped in favor of recall.

The Extra Trees Classifier’s robustness stems from its ensemble approach, where it constructs a multitude of randomized decision trees and averages their predictions. This method not only improves predictive accuracy but also controls over-fitting, making the model more generalizable to unseen data.

Given these compelling attributes, the decision to adopt the Extra Trees Classifier for our fraud detection system was clear. Its superior recall metric assures us that it will be an effective tool in identifying fraudulent activities, thereby safeguarding the integrity of our operations. The model’s high accuracy ensures that we can rely on its predictions and its precision guarantees that the number of false positives remains manageable.

In summary, the Extra Trees Classifier aligns perfectly with our objectives for a robust fraud detection system, offering a blend of high accuracy, excellent recall, and satisfactory precision. It stands as the optimal choice for our needs, promising to enhance our fraud detection capabilities significantly.

Reference :

[1] Auto Insurance Claims Data: Kaggel

[2] Dealing with Imbalanced Datasets: kaggel