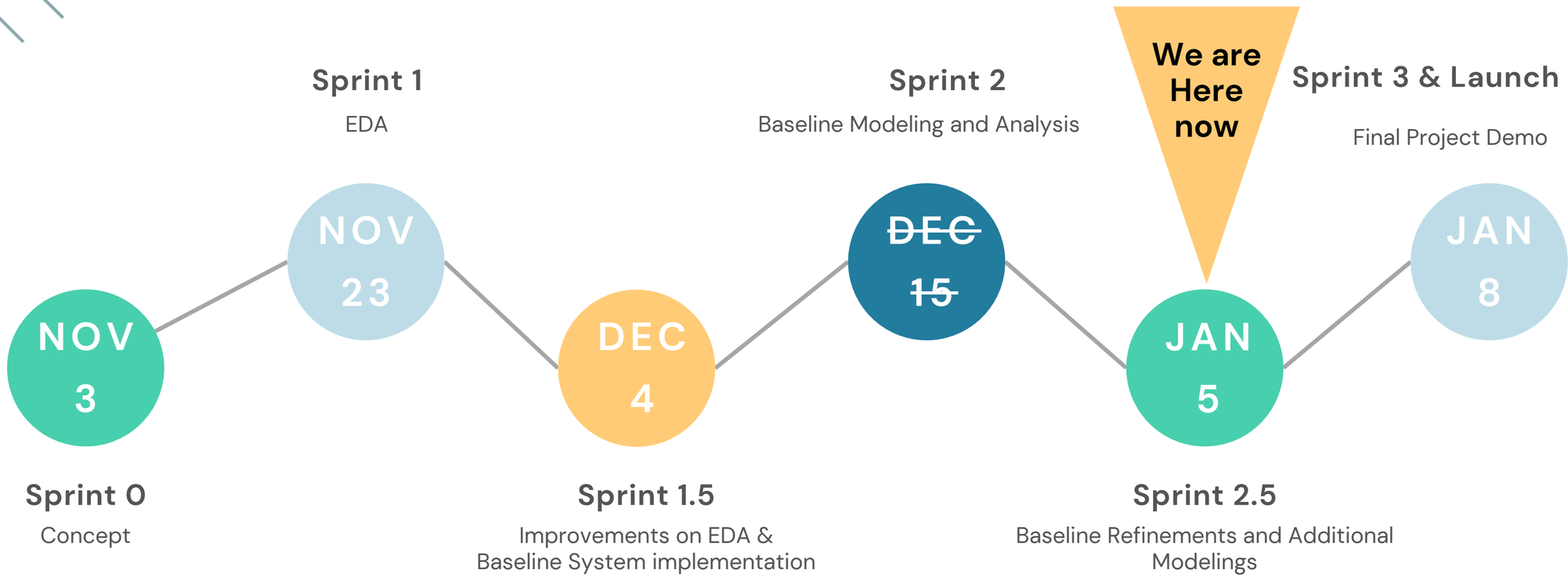
VEHICLE INSURANCE CLAIM FRAUD DETECTION: MODELING

Hannah Lal
January 2024

PROJECT INTRODUCTION

Detecting and preventing these deceptive activities is crucial for minimizing financial losses for issuers as well as safeguarding policyholders. Given its significant impact, insurance fraud detection has emerged as a prominent research area in data science and machine learning. This capstone project aims to concentrate on identifying and preventing fraudulent or misleading insurance claims using the dataset available at Kaggle. The objective is to develop algorithms and models that can automatically detect suspicious activities through historical data analysis.

PROJECT TIMELINE



FEATURE OVERVIEW

KAGGLE

AGE_OF_VEHICLE_OWENER AGE_OF_VEHICLE

MAKE_1 ('PONTIAC', 'TOYOTA', 'HONDA') VEHICLE_PRICE

MAKE_2 ('CHEVROLET', 'MAZDA') MARRIED

SEX BASE_POLICY_COLLISION BASE_POLICY_LIABILITY

POLICY_TYPE_1 ('SEDAN - COLLISION', 'SEDAN - LIABILITY', 'SEDAN - ALL PERILS')

POLICY_TYPE_2 ('SPORT - COLLISION', 'UTILITY - ALL PERILS')

FAULT VEHICLE_CATEGORY_SEDAN

VEHICLE_CATEGORY_SEDAN DEDUCTIBLE

DRIVARE RATING AGE OF POLICY HOLDER POLICE REPORT FILED

WITNESS PRESENT NUMBER OF PAST CLAIMS AGENT TYPE

NUMBER OF CARS NUMBER OF SUPPLIMENTS

ADDRESS CHANGE CLAIM DAYS POLICY CLAIM DAYS POLICY ACCIDENT

FEATURE SET

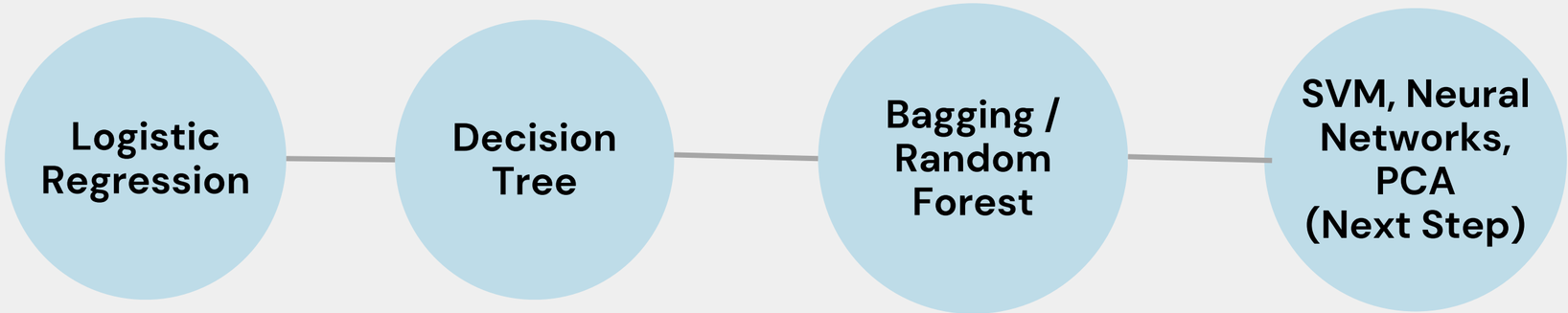
FRAUDFOUND_P (FAULT)

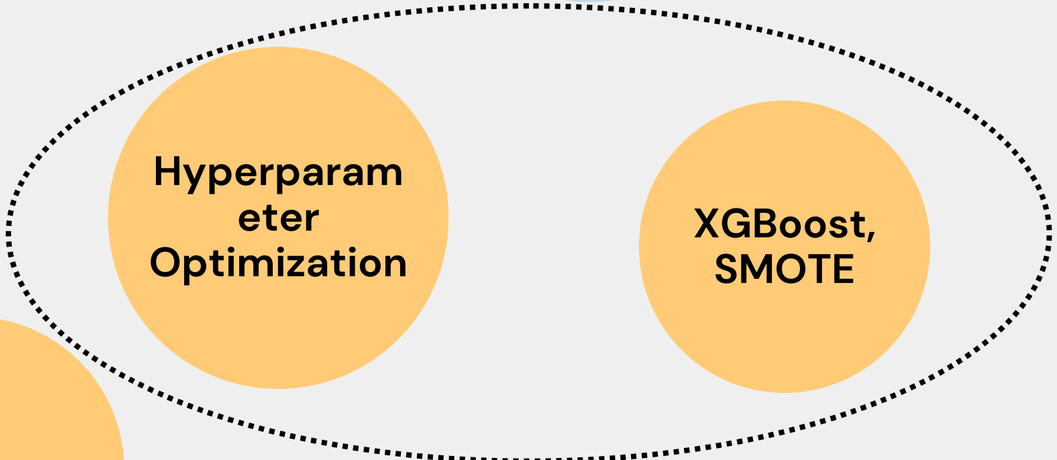
TARGET VERIABLE

https://blogs.oracle.com/machinelearning/post/a-two-step-process-for-detecting-fraud-using-oracle-machine-learning

MODELING:

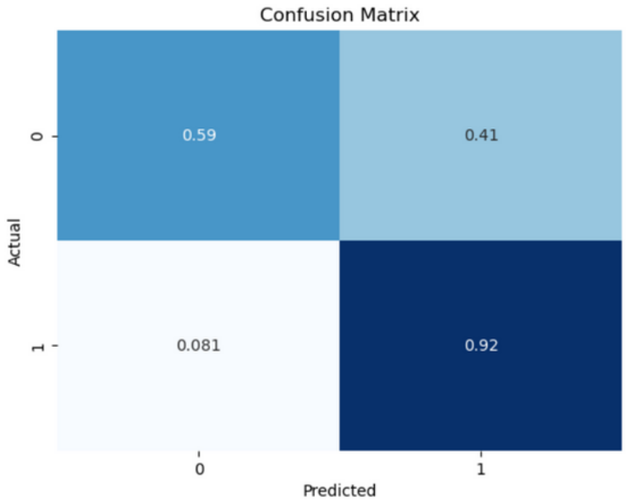
- Logistic Regression
- Logistic Regression with SMOTE
- Logistic Regression with GridSearchCV and CrossValidation
- Logistic Regression with XGBoost
- Logistic Regression Measurement Metrics
- Logistic Regression Feature Importance
- Decision Tree with GraidSearchCV
- Bagging/Random Forests with GridSearchCV
- Decision Tree Measurement Metrics
- Decision Tree Feature Importance
- Best-Performing Model

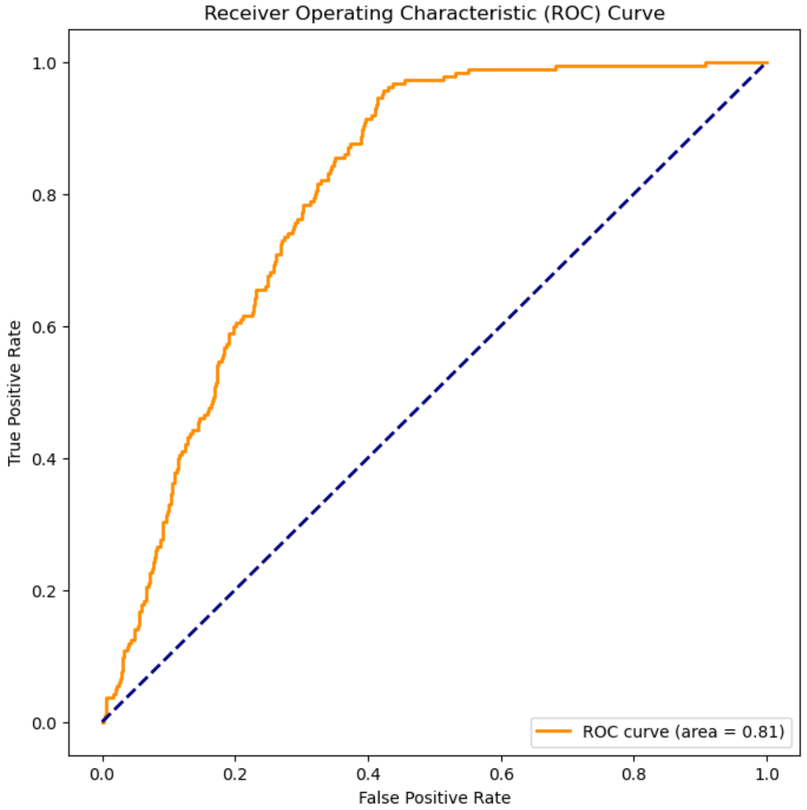




LOGISTIC REGRESSION

Accuracy for train: 0.61151195784353 46
Accuracy for test: 0.6092736705577172





ROC AUC Score: 0.8059666427379432

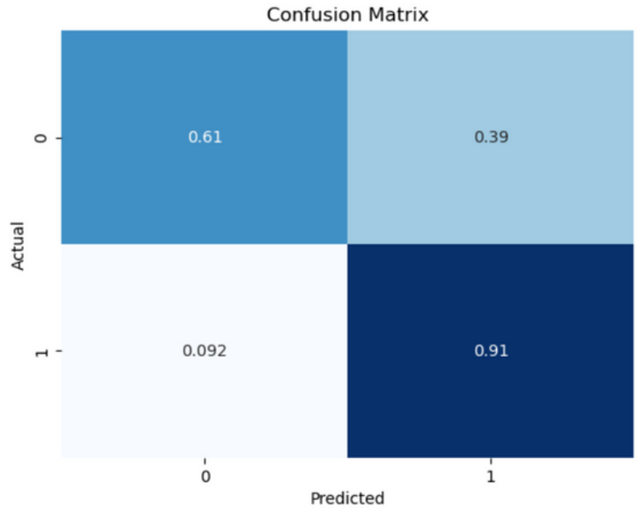
```
Precision: 0.125
Recall: 0.918918918918919
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.59      0.74      2899
           1       0.12      0.92      0.22       185

    accuracy                           0.61      3084
   macro avg       0.56      0.75      0.48      3084
weighted avg       0.94      0.61      0.71      3084

ROC AUC Score: 0.8059666427379432
```

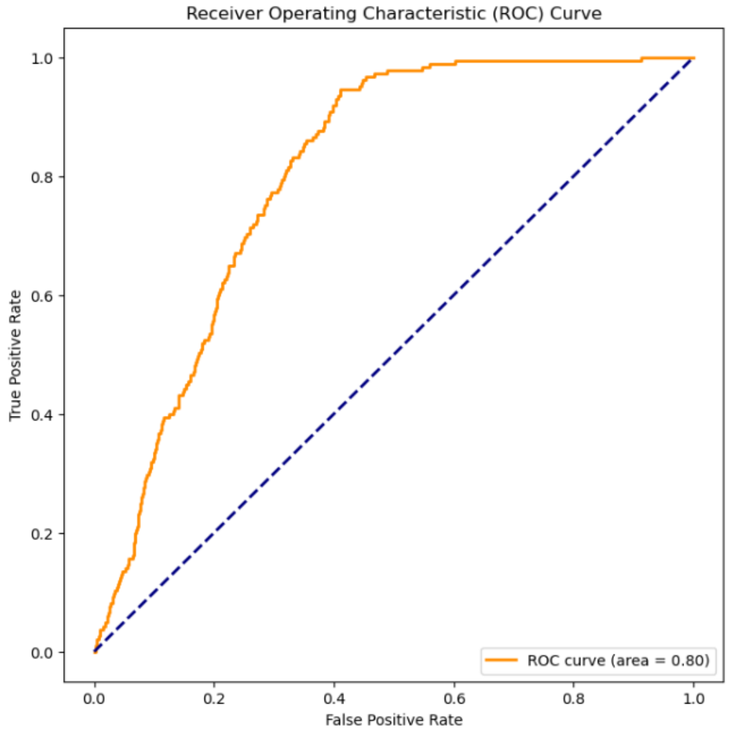
Precision and Recall:

- Percision for class 1: When the model predicts fraud (class 1), it is correct only 12.5% of the time
- Recall for class 1: The model captures 91.9% of the actual instances of fraud

LOGISTIC REGRESSION WITH SMOTE (TO ADDRESS IMBALANCE OF DATA)

Accuracy for train: 0.6248885285772193
Accuracy for test: 0.6235408560311284





ROC AUC Score: 0.8048907824692578

```
Classification Report LogReg with Smote:
              precision    recall  f1-score   support

           0       0.99      0.61      0.75      2899
           1       0.13      0.91      0.22       185

    accuracy                           0.62      3084
   macro avg       0.56      0.76      0.49      3084
weighted avg       0.94      0.62      0.72      3084

ROC AUC Score: 0.8048907824692578
Precision: 0.12804878048780488
Recall: 0.9081081081081082
```

LOGISTIC REGRESSION WITH GRIDSEARCHCV AND CROSSVALIDATION (STRATIFIEDKFOLD)

Best Parameters: {'C': 0.01, 'penalty': 'l1', 'solver': 'liblinear'}

Best Found Model

Cross-validated ROC AUC scores:

	param_C	param_penalty	param_solver	mean_test_score
0	0.001	l1	liblinear	0.871199
1	0.001	l1	saga	0.871199
2	0.001	l2	liblinear	0.907832
3	0.001	l2	saga	0.876669
4	0.01	l1	liblinear	0.910535
5	0.01	l1	saga	0.910535
6	0.01	l2	liblinear	0.895643
7	0.01	l2	saga	0.887516
8	0.1	l1	liblinear	0.890228
9	0.1	l1	saga	0.890228
10	0.1	l2	liblinear	0.883453
11	0.1	l2	saga	0.883453
12	1	l1	liblinear	0.883462
13	1	l1	saga	0.883462
14	1	l2	liblinear	0.883462
15	1	l2	saga	0.883462
16	10	l1	liblinear	0.883462
17	10	l1	saga	0.883462
18	10	l2	liblinear	0.883462
19	10	l2	saga	0.883462
20	100	l1	liblinear	0.883462
21	100	l1	saga	0.883462
22	100	l2	liblinear	0.883462
23	100	l2	saga	0.883462

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.58	0.74	2899
1	0.13	0.94	0.22	185
accuracy			0.61	3084
macro avg	0.56	0.76	0.48	3084
weighted avg	0.94	0.61	0.70	3084

ROC AUC Score: 0.7983852773090441

We see enhanced performance for class 1 recall measure.
This is the best model found so far!

LOGISTIC REGRESSION WITH GRIDSEARCHCV & XGBOOST

Best Parameters: {'colsample_bytree': 1.0, 'learning_rate': 0.2, 'max_depth': 9, 'n_estimators': 200, 'subsample': 0.9}

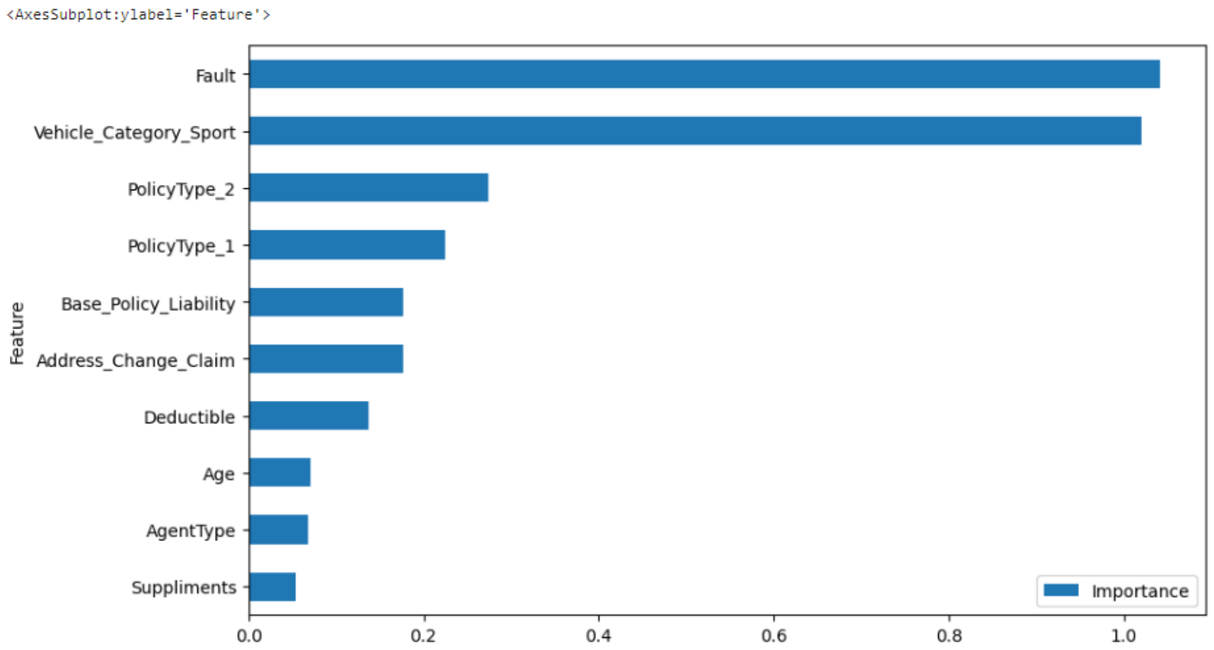
Classification Report:

	precision	recall	f1-score	support
0	0.94	0.99	0.96	2899
1	0.22	0.05	0.09	185
accuracy			0.93	3084
macro avg	0.58	0.52	0.53	3084
weighted avg	0.90	0.93	0.91	3084

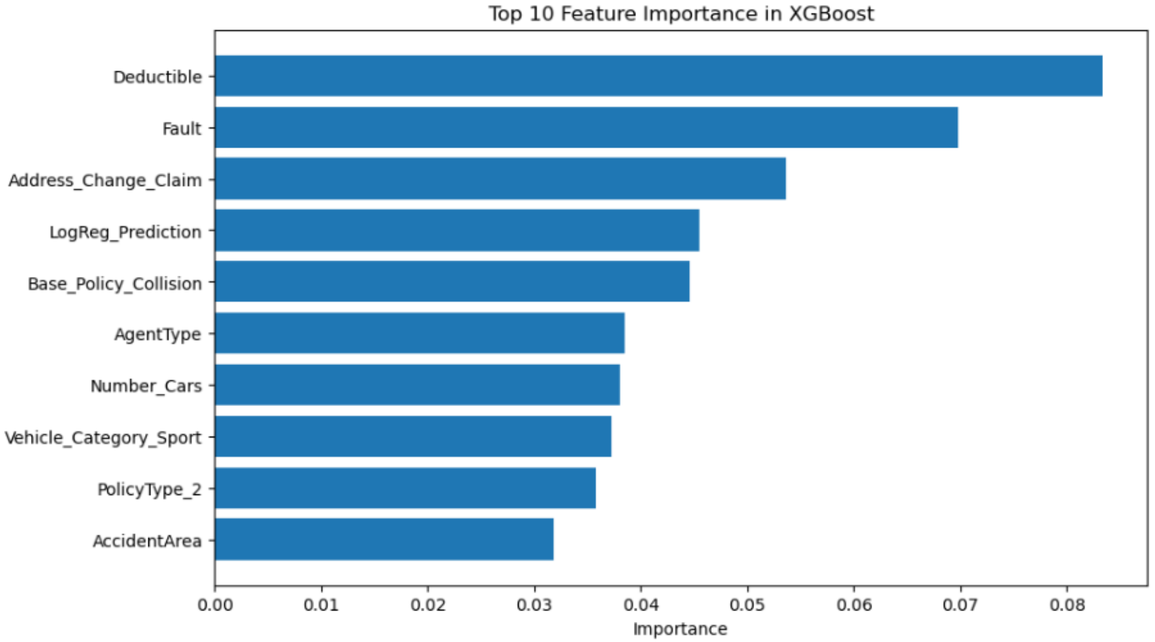
We see awfully low performance for class 1 recall measure.
This is the worst model found so far!

LOGISTIC REGRESSION FEATURE IMPORTANCE

Best Model Feature Importances:

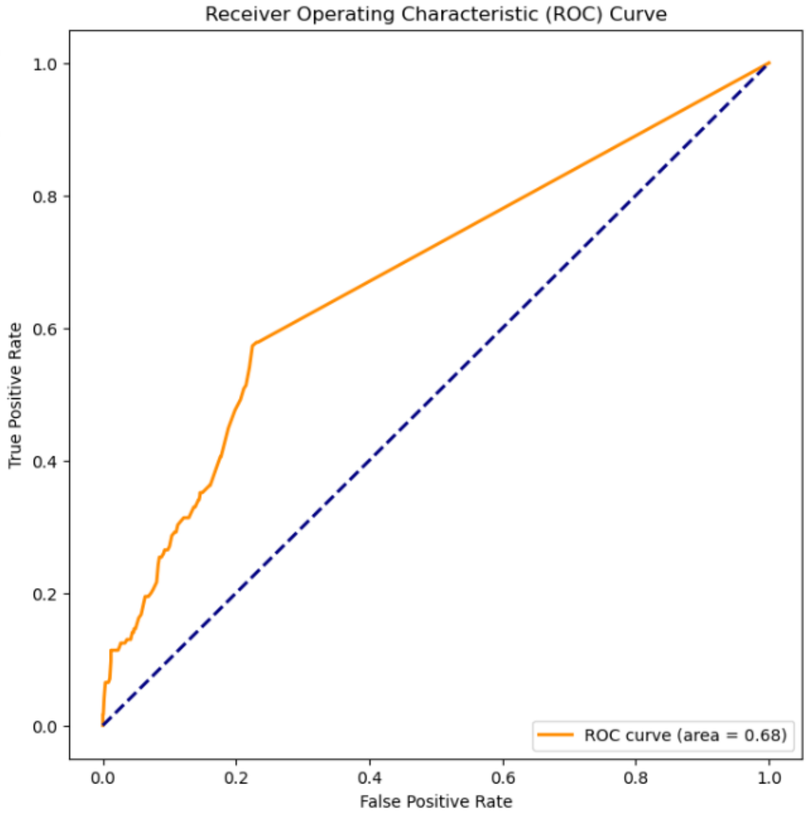


Worst Model Feature Importances:



The worst model is still able to recognize some important features as found in the best model, yet it introduces some otherwise not important features among the important ones.

DECISION TREE WITH GRAIDSEARCHCV



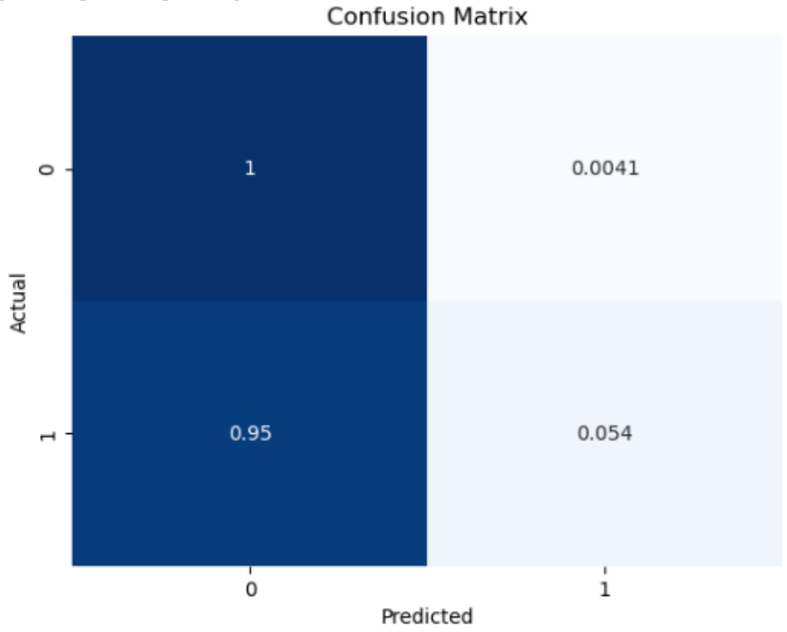
ROC AUC Score: 0.6750678239467477

```
Best Hyperparameters:
{'ccp_alpha': 0.0001, 'max_depth': 30, 'min_samples_leaf': 10}
Classification Report for Decision Tree:
              precision    recall  f1-score   support

           0       0.96      0.79      0.86      2899
           1       0.13      0.51      0.21       185

    accuracy                           0.77      3084
   macro avg       0.55      0.65      0.54      3084
weighted avg       0.91      0.77      0.83      3084

ROC AUC Score: 0.6750678239467477
Precision: 0.13231197771587744
Recall: 0.5135135135135135
```

- For class 1, the recall measure is now considerably decreased (from around 94% to 51%).
- ROC AUC score suggests we can no longer distinguish the two classes as well as we used to in the best found model..

BAGGING/RANDOM FORESTS WITH GRIDSEARCHCV

Fitting 5 folds for each of 180 candidates, totalling 900 fits
Best score: 0.9408998783948114
Best parameters: {'ccp_alpha': 0.0001, 'max_depth': 50, 'n_estimators': 50}



Classification Report:

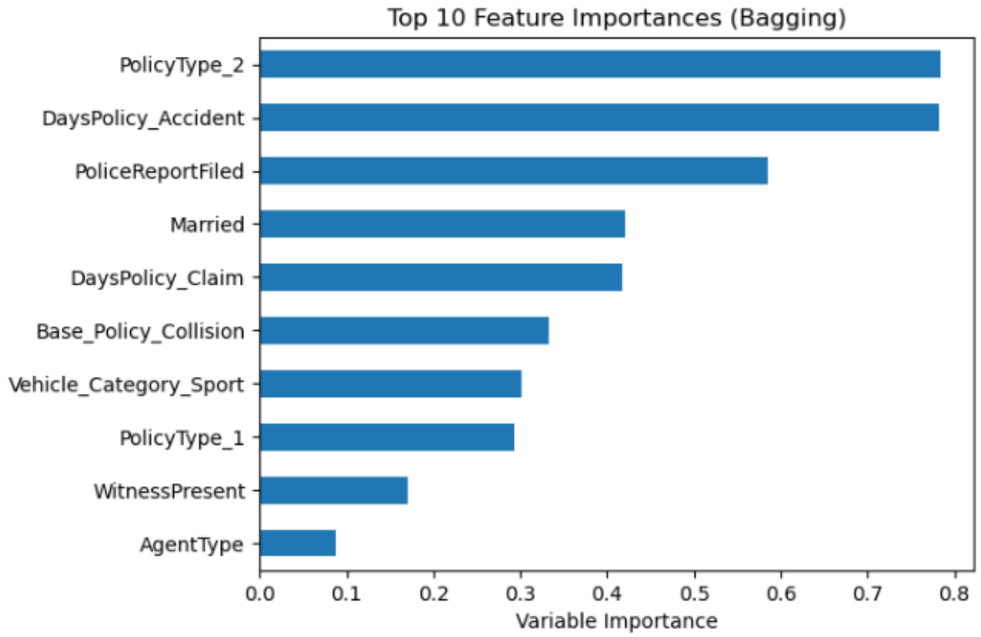
	precision	recall	f1-score	support
0	0.94	1.00	0.97	2899
1	0.45	0.05	0.10	185
accuracy			0.94	3084
macro avg	0.70	0.52	0.53	3084
weighted avg	0.91	0.94	0.92	3084

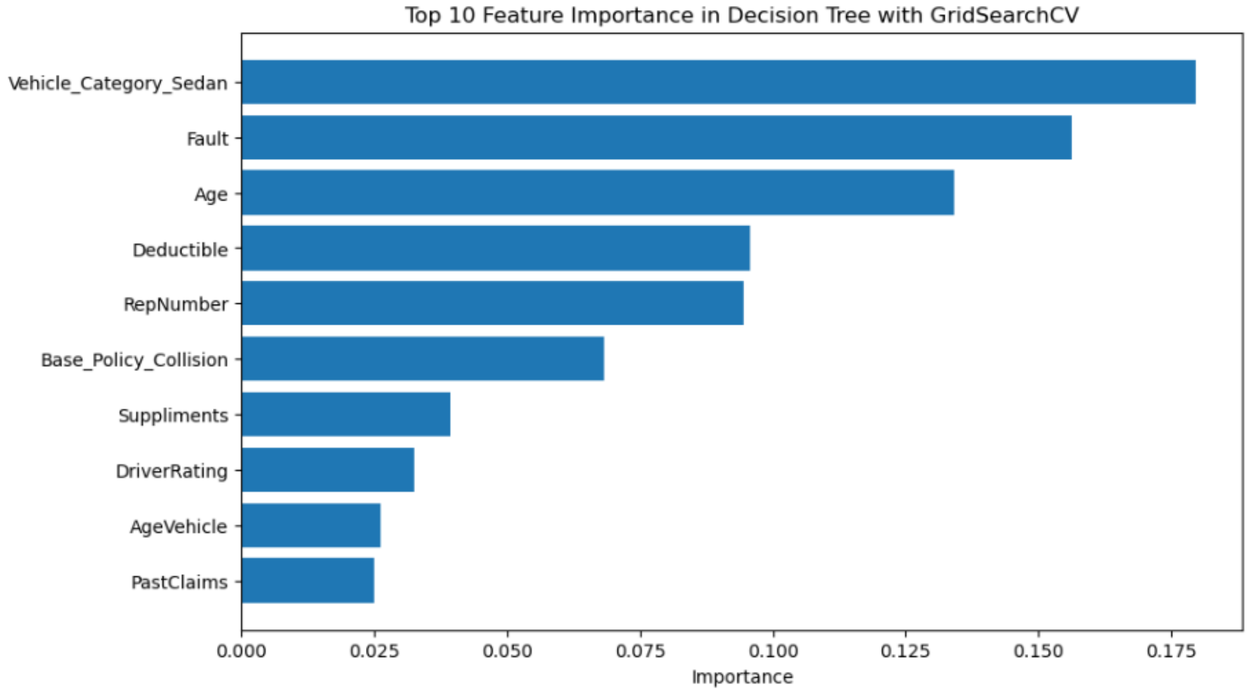
ROC AUC Score: 0.7961869423752832

The model is not able to correctly classify fraudulent data. Even though the precision is quite high as opposed to previous models, the recall measure is not at all satisfying.

DECISION TREE FEATURE IMPORTANCE

<Figure size 600x400 with 0 Axes>



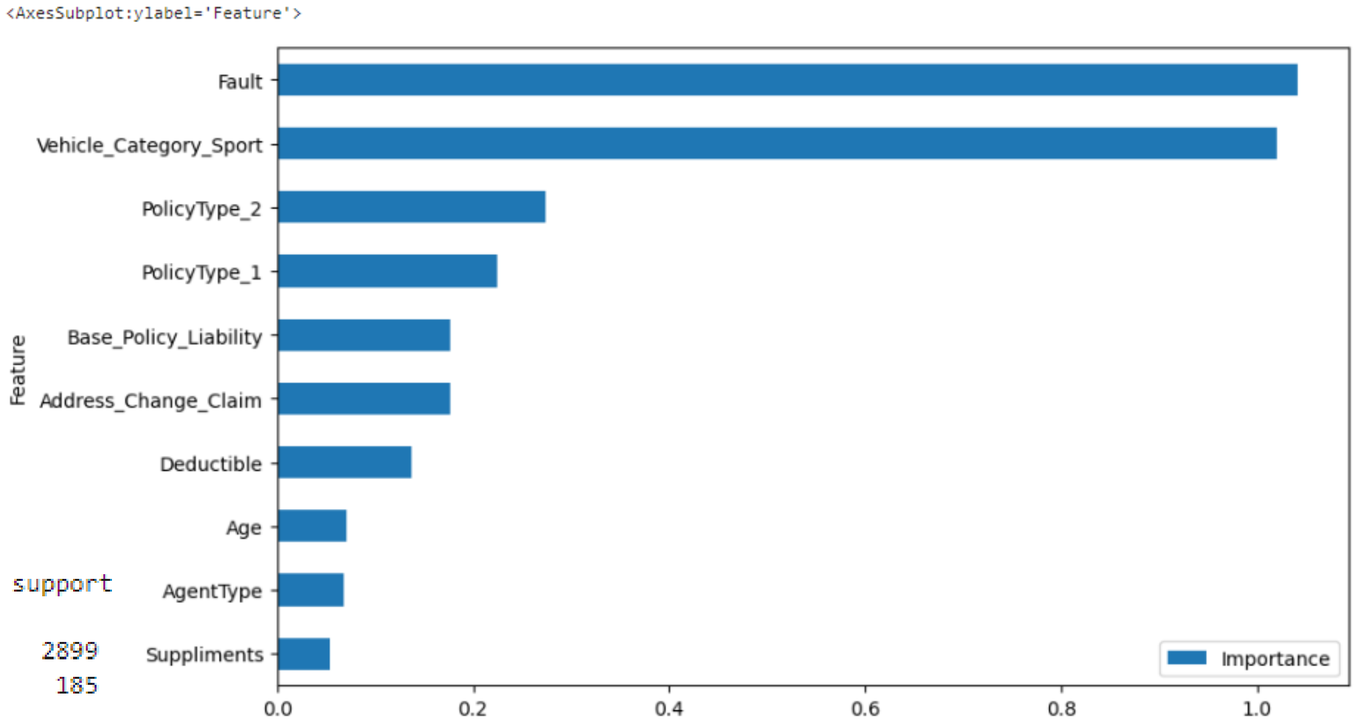


Random Forests may not perform well on certain types of data. If the dataset is small, highly imbalanced, or contains noisy features, a single Decision Tree might capture the patterns more effectively. Random Forests excel in situations where there is a large and diverse dataset.

Random Forests are generally computationally more expensive than a single Decision Tree. If the computational resources are limited, GridSearchCV might not be able to explore the hyperparameter space sufficiently.

BEST-PERFORMING MODEL

Logistic Regression with GridSearchCV and CrossValidation (StratifiedKFold)




```
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.58      0.74      2899
           1       0.13      0.94      0.22       185

    accuracy                           0.61      3084
   macro avg       0.56      0.76      0.48      3084
weighted avg       0.94      0.61      0.70      3084

ROC AUC Score: 0.7983852773090441
```

THANK YOU