Fraudulent Claim Detection

Rupam Mallick
Muthumariappan B

Problem Statement

- Global Insure faces significant financial losses due to fraudulent claims.
- Current fraud detection relies on manual inspections, leading to:
    - Time-consuming processes
    - Late detection (after payouts)
    - Inefficiency

Objective

Build a predictive model to classify insurance claims as fraudulent or legitimate using:

- Historical claim data (claim amounts, types)
- Customer profiles
- Other relevant features

Key Questions to Address:

1. Pattern Analysis – How can we detect fraud indicators in historical claims?
2. Predictive Features – Which factors best predict fraud?
3. Fraud Likelihood – Can we score new claims for fraud risk before approval?
4. Actionable Insights – How can the model improve fraud detection?

Data Preparation

Objective: Load the dataset and get a basic understanding of its structure and content.

Actions:

1. Import necessary libraries (pandas, numpy, seaborn, matplotlib).
2. Load the insurance_claims.csv dataset into a pandas DataFrame.
3. Display the first few rows of the DataFrame (df.head()) to preview the data.
4. Check the dimensions of the dataset (df.shape).
5. Inspect the data types of each column (df.dtypes).

Data Cleaning

Objective: Handle missing values, redundant columns/values, and incorrect data types to ensure data quality.

Actions:

- Handle Null Values
- Dropped column with complete null values.
- Identify and Handle Redundant Values and Columns
- Fix Data Types

Train Validation Split

Objective: Divide the data into training and validation sets to train and evaluate the model effectively.

Actions:

1. Import train_test_split from sklearn.model_selection.
2. Defined feature variables (X) by dropping the target column (fraud_reported).
3. Defined the target variable (y).
4. Split the data into 70% training and 30% validation sets using train_test_split.

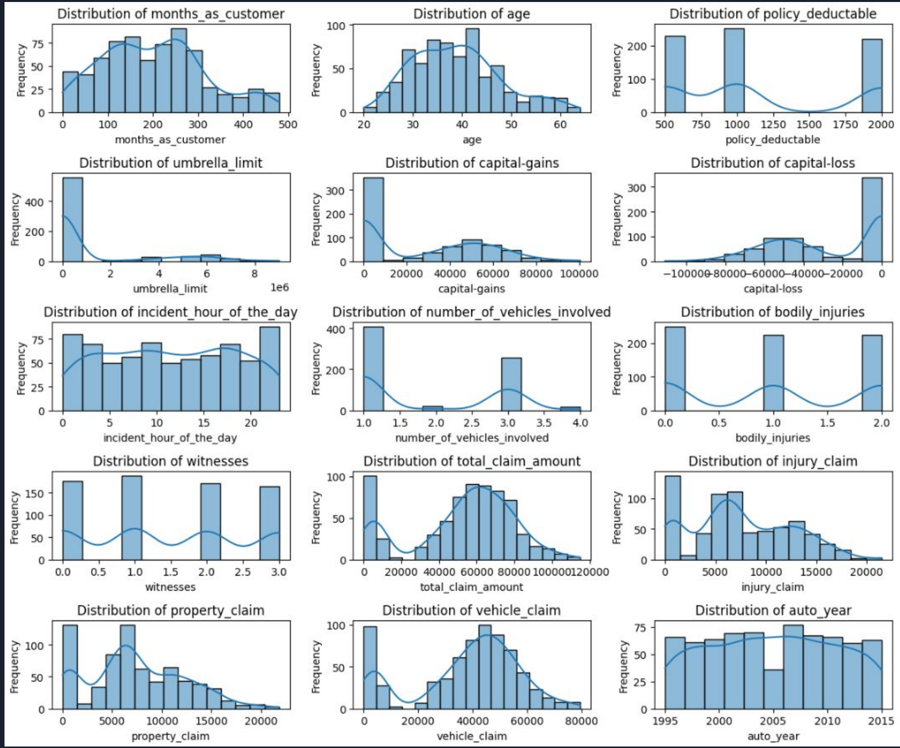
EDA on Training Data

Objective: Explore and visualize the training data to understand feature distributions, relationships, and identify patterns related to fraudulent claims.

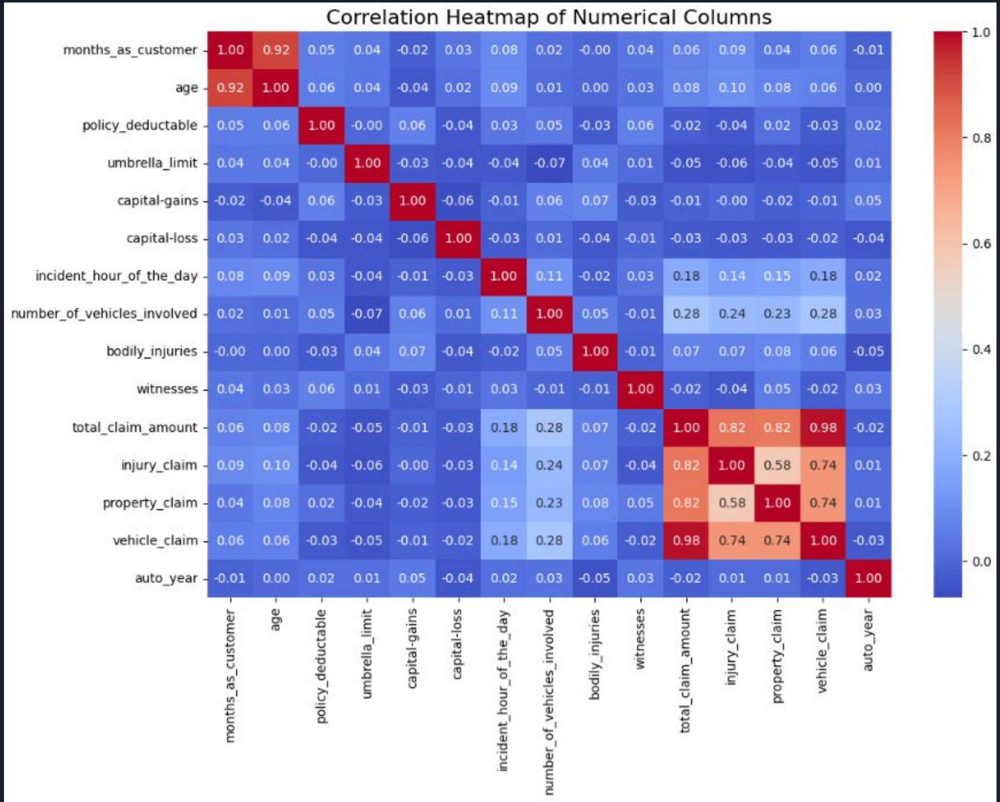
Actions:

- Univariate Analysis
- Correlation Analysis
- Check Class Balance
- Bivariate Analysis

EDA - Univariate Analysis

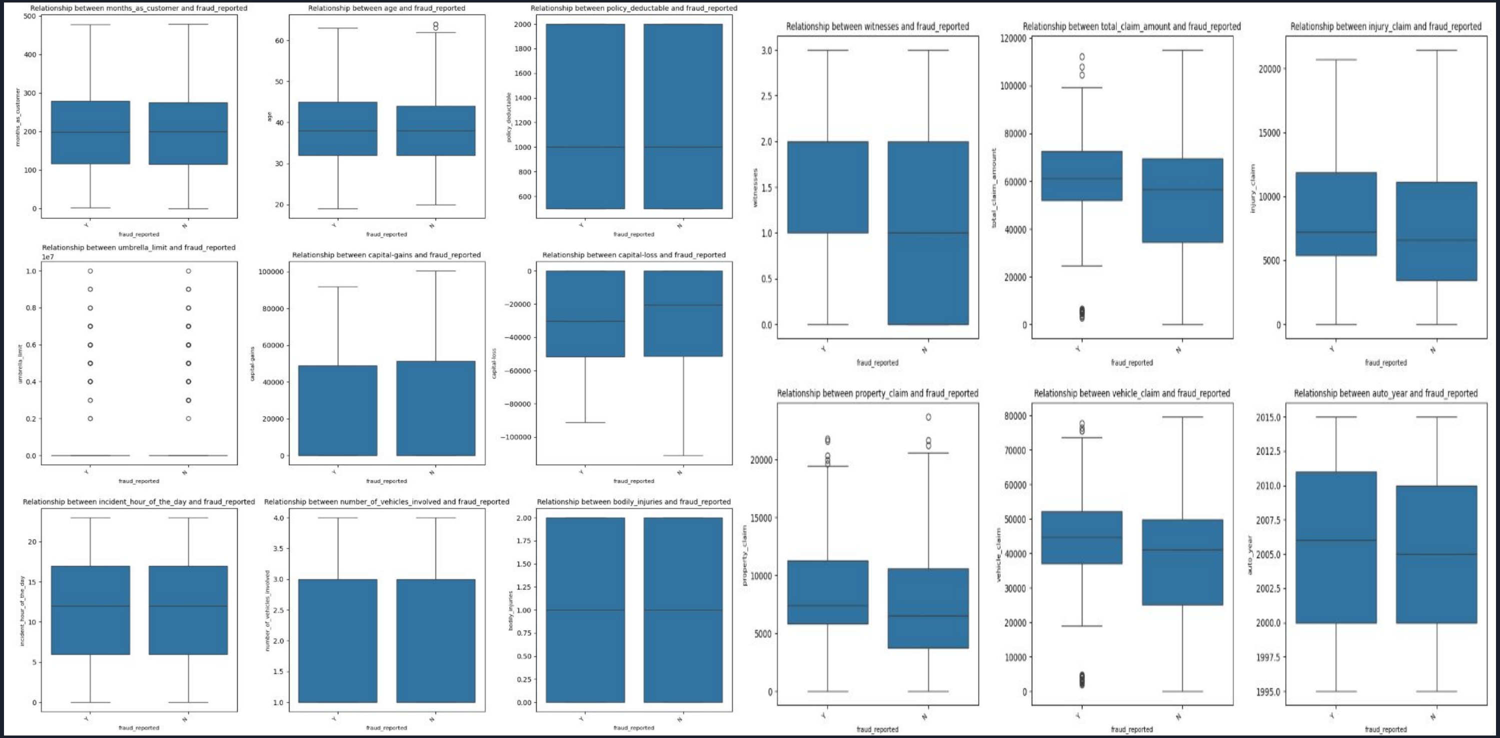


EDA - Correlation Analysis



EDA - Class Distribution

EDA - Bivariate Analysis Results



Feature Engineering

Objective: Transform and create new features to improve model performance and handle data characteristics like class imbalance.

Actions:

1. Perform Resampling
2. Feature Creation - Created new features (year, month, dayofweek) from the incident_date column
3. Handle Redundant Columns - dropped columns with high correlation
4. Combine values in Categorical Columns
5. Dummy Variable Creation
6. Feature Scaling

Model Building - Overview

- Logistic Regression & Random Forest Approaches
- Feature Selection, Training, Evaluation, and Optimization

Logistic Regression: Feature Selection

RFECV (Recursive Feature Elimination with Cross-Validation)

- Identifies most relevant features
- Optimal number of features chosen based on cross-validation performance

```
Summary of Model Building:

Optimal number of features (Logistic Regression): 43
Selected Features (Logistic Regression): ['incident_severity_Minor Damage', 'insured_hobbies_chess', 'incident_severity_Total Loss', '
```

Logistic Regression: Model Building

- Built using statsmodels library for detailed statistical outputs
- Checked P-values for feature significance
- Checked VIFs for multicollinearity detection

Logistic Regression Model Summary:

Logit Regression Results

Dep. Variable:	fraud_reported	No. Observations:	1052
Model:	Logit	Df Residuals:	1008
Method:	MLE	Df Model:	43
Date:	Tue, 12 Aug 2025	Pseudo R-squ.:	0.5065
Time:	18:48:42	Log-Likelihood:	-359.84
converged:	True	LL-Null:	-729.19
Covariance Type:	nonrobust	LLR p-value:	1.601e-127

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.3446	0.405	-0.852	0.394	-1.138	0.448
umbrella_limit	0.3297	0.103	3.206	0.001	0.128	0.531
witnesses	0.4075	0.105	3.883	0.000	0.202	0.613
policy_state_OH	0.3258	0.213	1.532	0.126	-0.091	0.743
policy_csl_250/500	0.4284	0.240	1.787	0.074	-0.042	0.898
policy_csl_500/1000	-0.2636	0.255	-1.035	0.301	-0.763	0.236
insured_education_level_JD	0.8657	0.286	3.031	0.002	0.306	1.425
insured_education_level_MD	1.0819	0.305	3.551	0.000	0.485	1.679
insured_education_level_PhD	0.7061	0.320	2.210	0.027	0.080	1.332
insured_occupation_adm-clerical	1.0016	0.487	2.056	0.040	0.047	1.956
insured_occupation_armed-forces	1.3546	0.491	2.759	0.006	0.392	2.317
insured_occupation_craft-repair	0.6795	0.427	1.590	0.112	-0.158	1.517
insured_occupation_exec-managerial	1.2657	0.435	2.911	0.004	0.414	2.118
insured_occupation_farming-fishing	0.7549	0.528	1.430	0.153	-0.280	1.790
insured_occupation_machine-op-inspct	0.8789	0.420	2.095	0.036	0.057	1.701
insured_occupation_priv-house-serv	-0.1479	0.524	-0.282	0.778	-1.175	0.879
insured_occupation_prof-specialty	1.2676	0.413	3.070	0.002	0.458	2.077
insured_occupation_sales	0.8643	0.483	1.791	0.073	-0.081	1.810
insured_occupation_tech-support	0.7991	0.444	1.801	0.072	-0.071	1.669
insured_occupation_transport-moving	1.7103	0.402	4.251	0.000	0.922	2.499
insured_hobbies_bungie-jumping	-0.6862	0.488	-1.406	0.160	-1.642	0.270
insured_hobbies_chess	6.0343	0.655	9.219	0.000	4.751	7.317
insured_hobbies_skydiving	-0.3722	0.504	-0.739	0.460	-1.359	0.615
insured_hobbies_yachting	0.4412	0.452	0.976	0.329	-0.445	1.327
insured_relationship_not-in-family	0.6808	0.299	2.278	0.023	0.095	1.266
insured_relationship_other-relative	0.3966	0.291	1.362	0.173	-0.174	0.967
insured_relationship_own-child	-0.5901	0.315	-1.876	0.061	-1.207	0.026
insured_relationship_unmarried	0.7753	0.314	2.471	0.013	0.160	1.390
collision_type_Front Collision	0.6978	0.256	2.724	0.006	0.196	1.200
collision_type_Rear Collision	0.7843	0.252	3.114	0.002	0.291	1.278
incident_severity_Minor Damage	-4.0310	0.292	-13.814	0.000	-4.603	-3.459
incident_severity_Total Loss	-3.4324	0.275	-12.492	0.000	-3.971	-2.894
incident_severity_Trivial Damage	-4.0472	0.571	-7.089	0.000	-5.166	-2.928
authorities_contacted_Other	0.4487	0.244	1.839	0.066	-0.030	0.927
incident_state_NY	-0.2170	0.251	-0.863	0.388	-0.710	0.276
incident_state_VA	0.8417	0.335	2.514	0.012	0.186	1.498
incident_state_WV	-0.8190	0.290	-2.828	0.005	-1.387	-0.251
incident_city_Northbrook	-0.7216	0.346	-2.087	0.037	-1.399	-0.044
property_damage_NO	-0.6207	0.226	-2.743	0.006	-1.064	-0.177
auto_make_Audi	1.4614	0.370	3.955	0.000	0.737	2.186
auto_make_BMW	0.5967	0.399	1.495	0.135	-0.186	1.379
auto_make_Dodge	0.6321	0.364	1.734	0.083	-0.082	1.347
auto_make_Nissan	-0.6645	0.423	-1.573	0.116	-1.493	0.164
auto_make_Other	1.0010	0.466	2.149	0.032	0.088	1.914

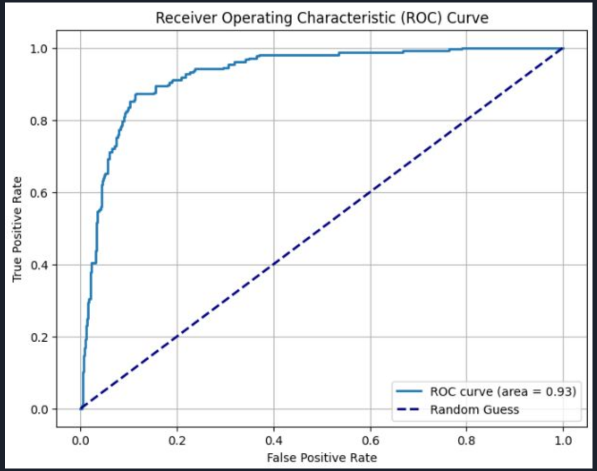
Logistic Regression: Training & Initial Evaluation

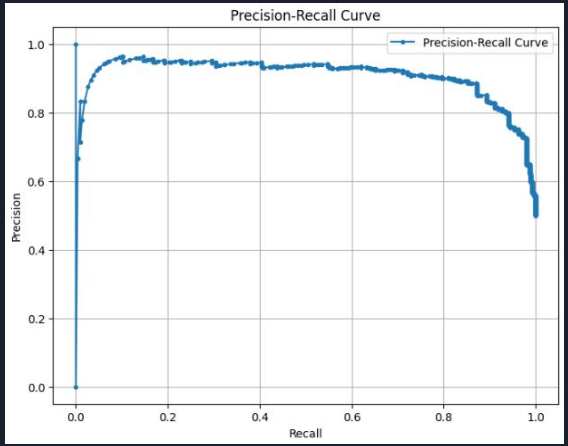
- Trained model on training data
- Evaluated using:
  - Accuracy
  - Confusion Matrix (cutoff = 0.5)

```
Logistic Regression Metrics on Training Data (Cutoff 0.5):
Accuracy: 0.8669201520912547
Confusion Matrix:
 [[453  73]
 [ 67 459]]
True Negative:  453
False Positive:  73
False Negative:  67
True Positive:  459
Sensitivity:  0.8726235741444867
Specificity:  0.8612167300380228
Precision:  0.8627819548872181
Recall:  0.8726235741444867
F1 Score:  0.8676748582230625
```

Logistic Regression: Optimal Cutoff

- Determined threshold by comparing:
  - Sensitivity
  - Specificity
  - Precision
  - Recall
- Used ROC and Precision-Recall curves





Logistic Regression: Final Evaluation

- Predictions made using optimal cutoff
- Re-evaluated model performance on training data

```
Logistic Regression Metrics on Training Data (Optimal Cutoff):
Accuracy: 0.879277566539924
Confusion Matrix:
 [[466  60]
 [ 67 459]]
True Negative:  466
False Positive:  60
False Negative:  67
True Positive:  459
Sensitivity:  0.8726235741444867
Specificity:  0.8859315589353612
Precision:  0.884393063583815
Recall:  0.8726235741444867
F1 Score:  0.8784688995215311
```

Random Forest: Feature Selection

- Obtained feature importance scores from initial model
- Selected features above importance threshold

Random Forest Feature Importance:

	Feature	Importance
30	incident_severity_Minor Damage	0.129818
21	insured_hobbies_chess	0.085115
31	incident_severity_Total Loss	0.077704
2	witnesses	0.048884
32	incident_severity_Trivial Damage	0.042676
1	umbrella_limit	0.036350
38	property_damage_NO	0.027926
36	incident_state_WV	0.027695
3	policy_state_OH	0.024897
28	collision_type_Front Collision	0.024682
4	policy_csl_250/500	0.023952
5	policy_csl_500/1000	0.022861
29	collision_type_Rear Collision	0.022149
34	incident_state_NY	0.021900
33	authorities_contacted_Other	0.021040
25	insured_relationship_other-relative	0.019196
26	insured_relationship_own-child	0.018632
7	insured_education_level_MD	0.018179
27	insured_relationship_unmarried	0.017846
6	insured_education_level_JD	0.017803
24	insured_relationship_not-in-family	0.016585
39	auto_make_Audi	0.015723
8	insured_education_level_PhD	0.015334
12	insured_occupation_exec-managerial	0.015122
41	auto_make_Dodge	0.014649
19	insured_occupation_transport-moving	0.014292
10	insured_occupation_armed-forces	0.014241
16	insured_occupation_prof-specialty	0.014116
37	incident_city_Northbrook	0.013911
35	incident_state_VA	0.013686
18	insured_occupation_tech-support	0.012375
42	auto_make_Nissan	0.010936
11	insured_occupation_craft-repair	0.010761
14	insured_occupation_machine-op-inspct	0.010680
15	insured_occupation_priv-house-serv	0.010105
40	auto_make_BMW	0.009942
17	insured_occupation_sales	0.009551
9	insured_occupation_adm-clerical	0.009536
43	auto_make_Other	0.008955
13	insured_occupation_farming-fishing	0.008633
22	insured_hobbies_skydiving	0.007378
20	insured_hobbies_bungie-jumping	0.007180
23	insured_hobbies_yachting	0.007003
0	const	0.000000

Random Forest: Model Training

- Trained model with selected features
- Evaluated using:
  - Accuracy
  - Confusion Matrix

```
Random Forest Metrics on Training Data (Base Model):
Accuracy: 1.0
Confusion Matrix:
 [[526   0]
 [  0 526]]
True Negative:  526
False Positive:  0
False Negative:  0
True Positive:  526
Sensitivity:  1.0
Specificity:  1.0
Precision:  1.0
Recall:  1.0
F1 Score:  1.0
```

Random Forest: Overfitting Check

- Used Cross-Validation to assess generalization
- Checked for overfitting

```
Random Forest Cross Validation Scores:
[0.90047393 0.96208531 0.97142857 0.94285714 0.94285714]
```

Random Forest: Hyperparameter Tuning

- Grid Search for best hyperparameters
- Optimized performance

```
Random Forest Best Hyperparameters:
{'max_depth': 20, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}
Random Forest CV score: 0.9448973143759873
```

Random Forest: Final Model

- Built model with best hyperparameters
- Trained & evaluated on training data

```
Random Forest Metrics on Training Data (Tuned Model):
Accuracy: 0.9990494296577946
Confusion Matrix:
 [[525   1]
 [  0 526]]
True Negative:  525
False Positive:  1
False Negative:  0
True Positive:  526
Sensitivity:  1.0
Specificity:  0.9980988593155894
Precision:  0.9981024667931688
Recall:  1.0
F1 Score:  0.9990503323836657
```

Predicting and Model Evaluation

Make predictions over validation data using logistic regression model:

- The relevant features were selected for the validation data, and a constant was added.
- Predictions were made on the validation data using the trained Logistic Regression model.
- A DataFrame was created to show the actual values and the predicted probabilities for the validation data.
- Final predictions were made using a cutoff value of 0.5.
- The accuracy, confusion matrix, TP, TN, FP, FN, sensitivity, specificity, precision, recall, and F1-score were calculated and printed for the validation data using the Logistic Regression model.

Predicting and Model Evaluation - Results

```
Summary of Prediction and Model Evaluation:

Logistic Regression Metrics on Validation Data:
Accuracy: 0.31333333333333335
Confusion Matrix:
 [[ 23 203]
 [  3  71]]
True Negative:  71
False Positive:  203
False Negative:  3
True Positive:  23
Sensitivity:  0.8846153846153846
Specificity:  0.2591240875912409
Precision:  0.10176991150442478
Recall:  0.8846153846153846
F1 Score:  0.18253968253968256
```

Predicting and Model Evaluation

Make predictions over validation data using random forest model:

- The important features were selected for the validation data.
- Probability predictions were made on the validation data using the trained Random Forest model.
- The accuracy, confusion matrix, TP, TN, FP, FN, sensitivity, specificity, precision, recall, and F1-score were calculated and printed for the validation data using the Random Forest model with a cutoff of 0.5.

Predicting and Model Evaluation - Results

```
Random Forest Metrics on Validation Data:
Accuracy: 0.78
Confusion Matrix:
 [[195  31]
 [ 35  39]]
True Negative:  195
False Positive:  31
False Negative:  35
True Positive:  39
Sensitivity:  0.527027027027027
Specificity:  0.8628318584070797
Precision:  0.5571428571428572
Recall:  0.527027027027027
F1 Score:  0.5416666666666666
```

Conclusion

- Random Forest clearly outperforms Logistic Regression in overall accuracy, specificity, precision, and F1 score, making it the better choice for balanced classification performance.
- Logistic Regression could still be preferred only if detecting every possible positive case (high recall) is the top priority and false positives are less costly.
- If the goal is balanced performance and fewer false positives, Random Forest is the more suitable model.