Vehicle Insurance Fraud Analysis

Christine Gendron
Mitchel Diaz
Steven Dookhantie

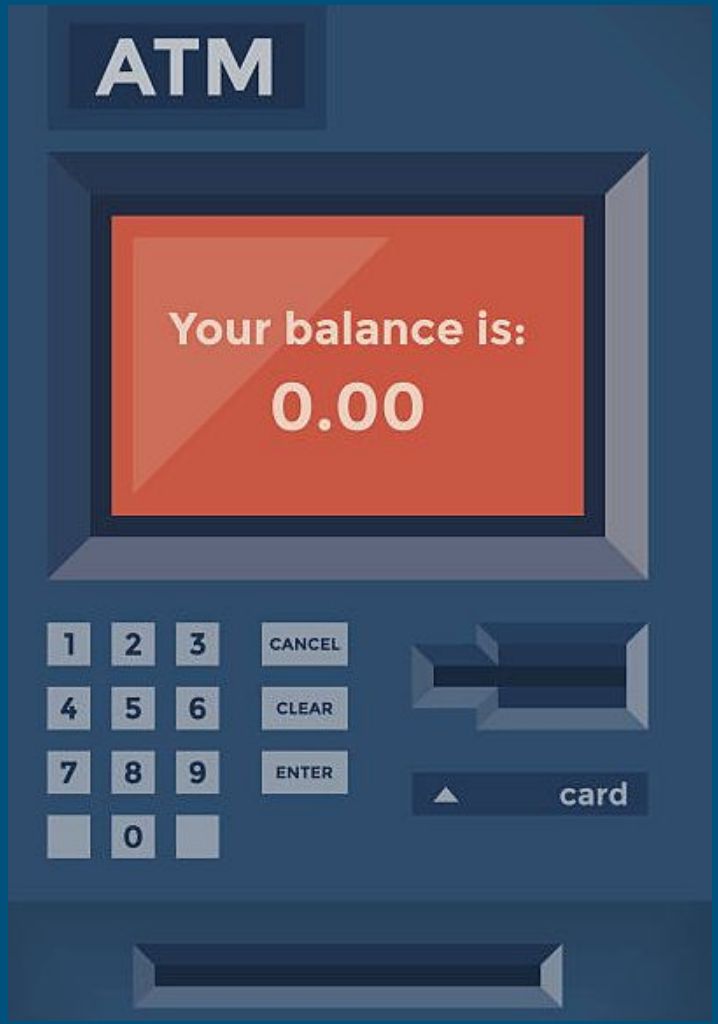
Our Purpose:

Vehicle Insurance Fraud occurs when someone attempts to deceive an insurance company while making a claim, in order to get a larger payout.

This could mean anything from misrepresenting a detail to faking an accident. It is an expensive, increasingly common, and avoidable problem for insurers.

Fraud is not exclusive to auto insurers. The entire financial services industry is experiencing rapidly rising levels of fraud, coming in many different forms. Firms aren't the only ones affected- fraudulent account activity can have devastating effects on consumers.

That's why using machine learning to replace time-intensive manual review with automated fraud detection processes will be essential moving forward.



The Questions:

- Can we accurately predict whether a vehicle insurance claim is fraudulent?

- Which features are correlated with higher likelihood of vehicle insurance fraud?

The Tools:

- Python, including multiple libraries (scikit-learn, pandas)
- The Logistic Regression and Random Forest Algorithms
- SQL
- Tableau
- HTML
- CSS

The Data:

Our dataset was generated by **Angoss KnowledgeSEEKER**, a provider of systems for predictive analytics, and found on **Kaggle**.

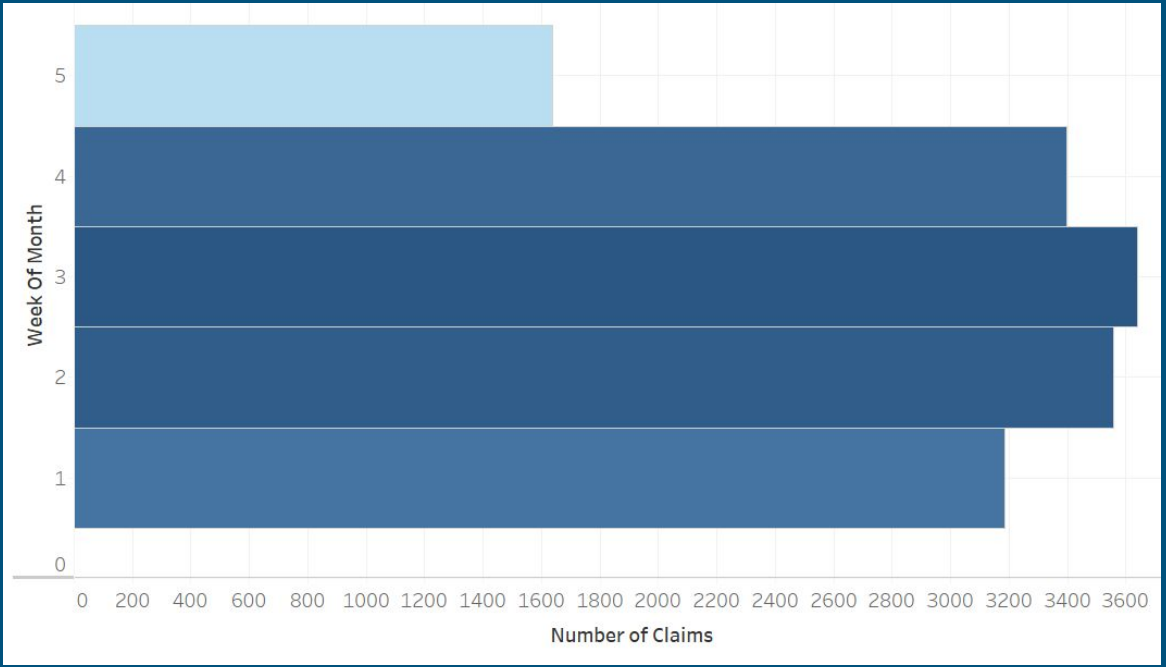
This set features over 15,000 claim samples with 33 columns, including information on:

- *Demographics of claimant*
- *Details on the vehicle in question*
- *Information about the claimants' policies.*
- *Whether fraud was detected*

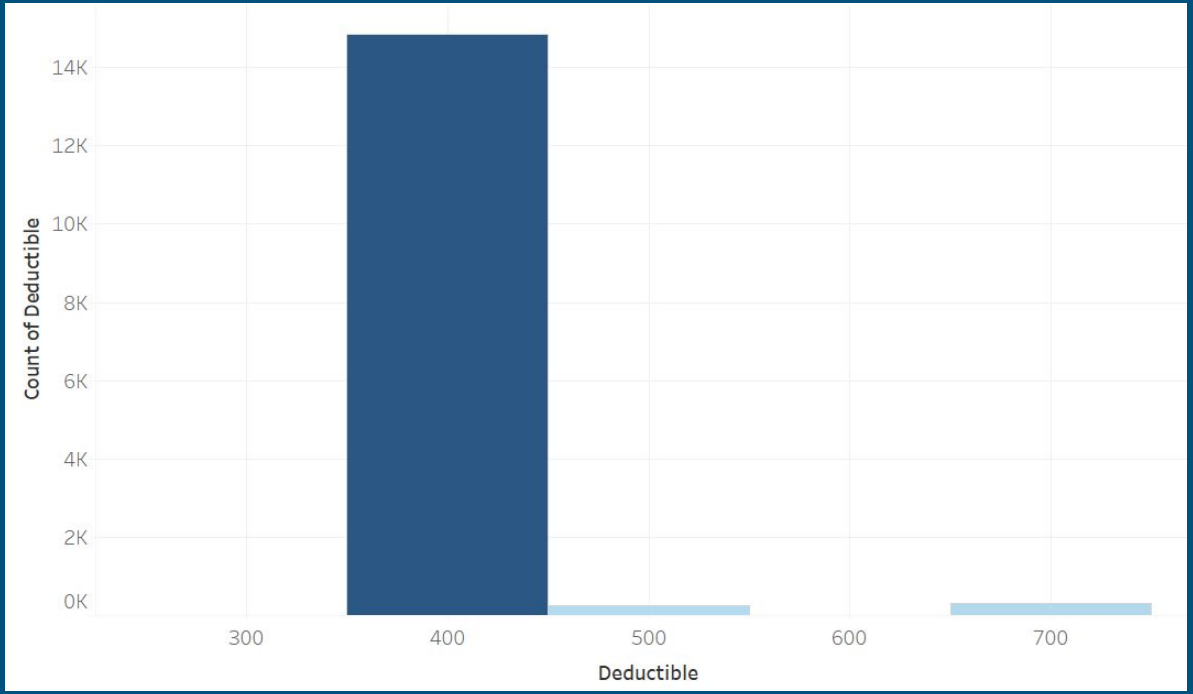
Data Exploration

Dropping Unnecessary Columns

Week Of Month was dropped, since the meaning was not clear- there is another, specific column for the week the claim was made.

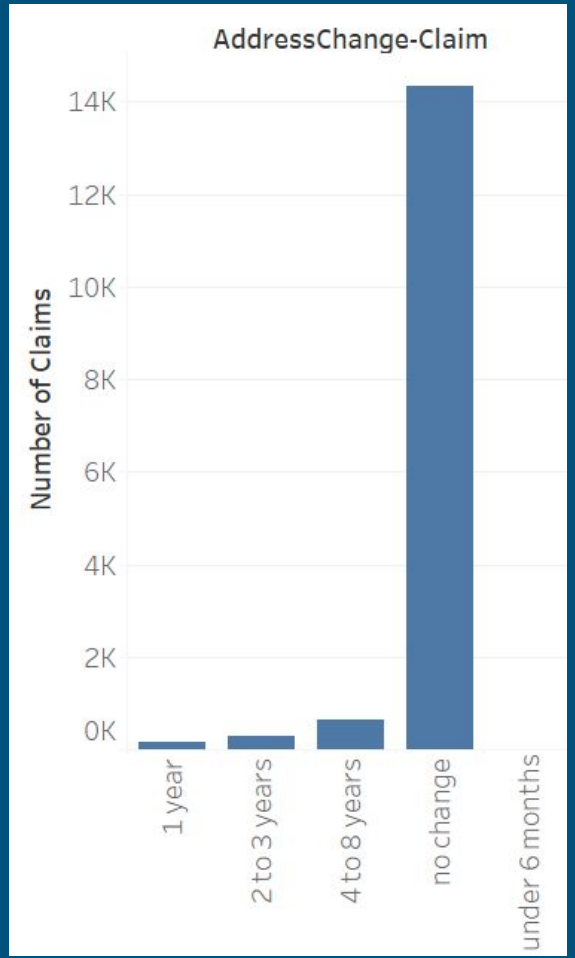


Deductible was dropped, since nearly all claims had a deductible of $400.



AddressChange-Claim was dropped, since most claims fell under “no change”, and the meaning of this column is ambiguous.

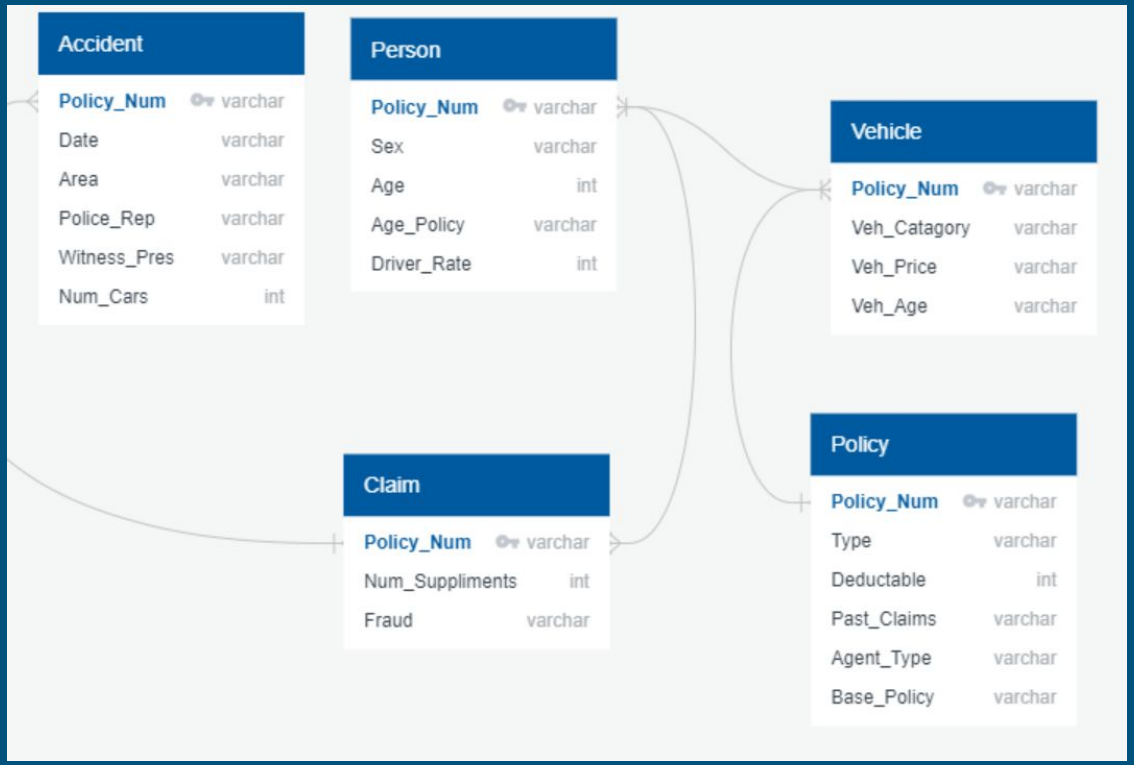
RepNumber was also dropped, since the ID number of the insurance rep handling the claim is not relevant to this analysis.



The Database:

- The data was originally in a csv format
- This static data is stored in a PostgreSQL database
- This database is connected to our Machine Learning model using AWS.

Our ERD:


Accident
Policy_Num varchar
Date varchar
Area varchar
Police_Rep varchar
Witness_Pres varchar
Num_Cars int
Person
Policy_Num varchar
Sex varchar
Age int
Age_Policy varchar
Driver_Rate int
Vehicle
Policy_Num varchar
Veh_Catagory varchar
Veh_Price varchar
Veh_Age varchar
Claim
Policy_Num varchar
Num_Suppliments int
Fraud varchar
Policy
Policy_Num varchar
Type varchar
Deductable int
Past_Claims varchar
Agent_Type varchar
Base_Policy varchar

Phase 2: Analysis

Missing Values

We handled missing values using two functions:

1) **missing_value(*dataframe, value*)** identified any missing values

2) **impute_missing(*cat_df, num_df, list_of_cols, dict_of_encoder*)** used KNNImputer to impute those missing values.

Dropping Highly Correlated Features

A For Loop drops numerical features with a correlation of above 0.8, since those features would make the algorithm more complex without adding significant information.

Logistic Regression:

Our dependent variable (FraudFound) is binary- either fraud was found, or it was not. Logistic Regression is an efficient way of predicting this type of outcome.

Logistic Regression Outcome:

Using Scikit-learn, we fitted and trained a model that predicted fraud to a high degree of accuracy, with a score of **0.94.**

```
prediction = logreg.predict(X_test)
```

```
accuracy_score(y_test, prediction)
```

```
0.9400129701686122
```

Random Forest

The Random Forest algorithm typically generates very accurate predictions by incorporating many decision trees, while preventing overfitting with an element of randomness.

Random Forest Outcome: Accuracy

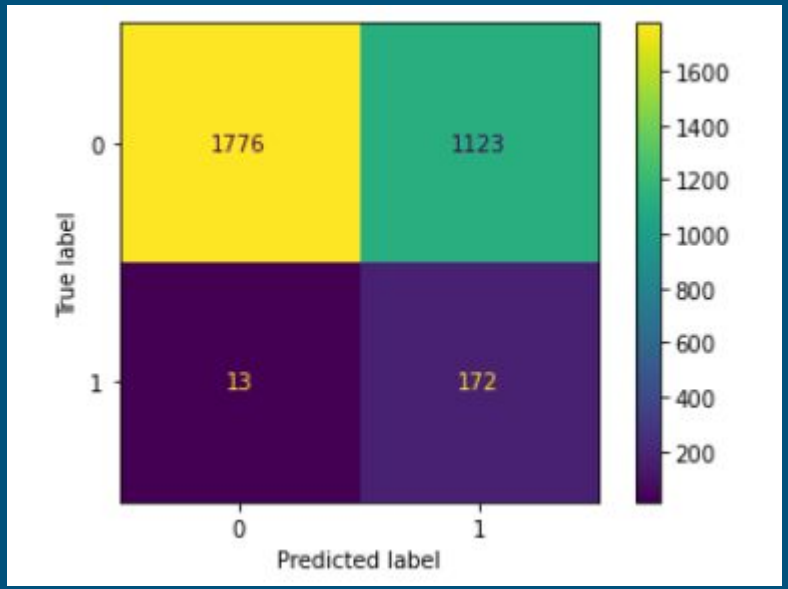
The Random Forest Classifier produced a less robust result, with an accuracy score of **0.632.**

```
# Calculated the balanced accuracy score
print(f"The balance accuracy score is: {accuracy_score(y_test, predictions):.3f}")
```

```
The balance accuracy score is: 0.632
```

Random Forest Outcome: Confusion Matrix

Based on the confusion matrix, the lower accuracy score is caused by a high number of false positives. **1,123** legitimate claims were incorrectly flagged as fraud, compared to only **13** missed fraudulent claims.



Ideas for The Future:

- It would be interesting to explore similar datasets with location data. Where is the most fraud being committed? That could help to hone our model, and would provide opportunities to create interactive maps.

- The Random Forest model could be trained further in order to increase the accuracy score. For example, we could increase or decrease the number of estimators.

The Dashboard:

Our dashboard, built using HTML and CSS, features:

- Tableau visualizations
- Images and summary of our Machine Learning Analysis