Car Insurance Fraud Detection

Starting with the Basics - A Logistic Regression Baseline

More models coming soon!

The Search for the Right Data

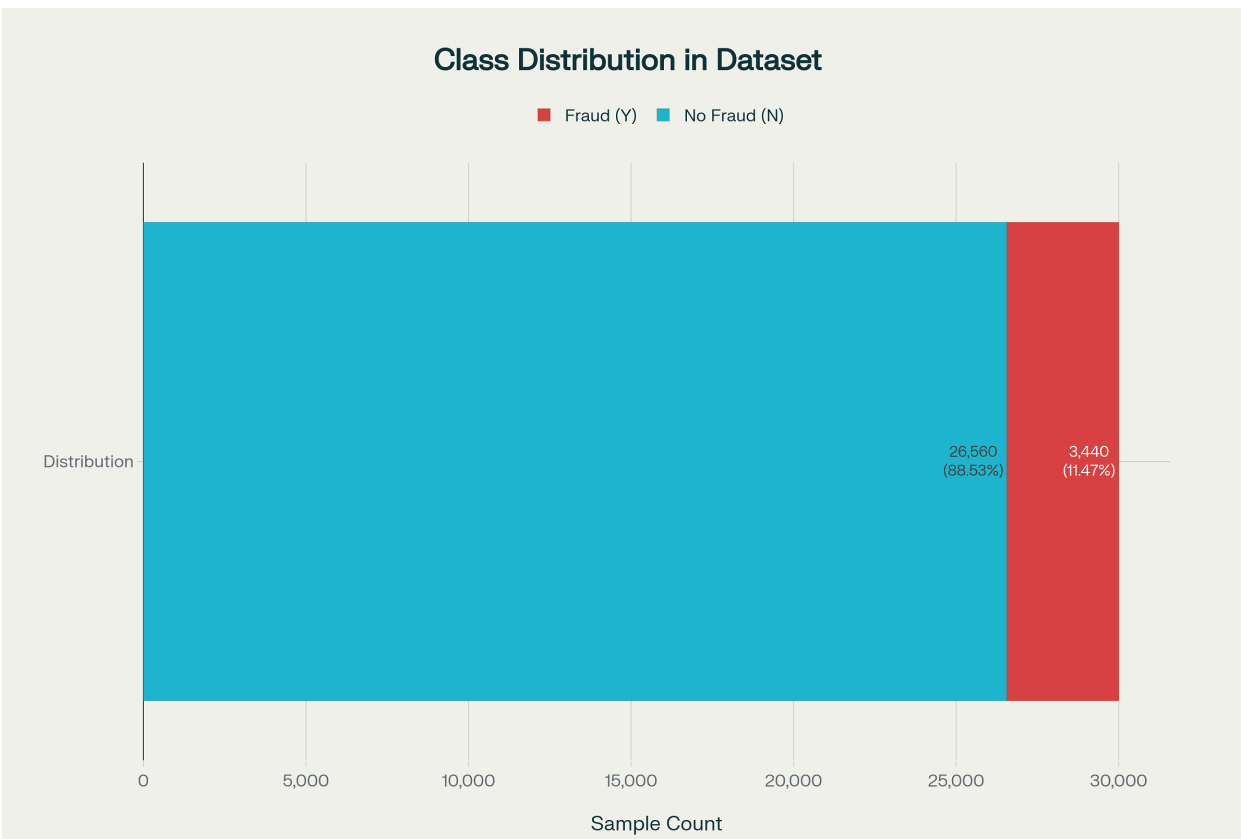
Finding the perfect dataset took most of the time, but it was worth it!

Dataset Highlights

- **Got 30,000 insurance claims to work with** - plenty of data to learn from
- **24 different features to play around with** - lots of angles to explore
- **Target variable:** Did they commit fraud or not? (fraud_reported)
- **The class split:** 88.5% legit claims vs 11.5% fraudulent ones
- **Used 80-20 train-test split** to keep things fair

The imbalanced data was a challenge, but that's real-world stuff!

Class Distribution



The data shows a significant imbalance - 26,560 legitimate claims compared to only 3,440 fraudulent ones. This reflects real-world patterns where fraud is the exception, not the rule.

What I Did with the Data

Feature Engineering & Model Setup

Data Preparation

Starting Materials:

- 15 numeric features (ages, claim amounts, premiums, witnesses, that kind of thing)
- 11 categorical features (states, occupations, incident types, etc.)

Processing Steps:

- Broke down dates into useful parts (day, month, which day of week)
- Used frequency encoding when there were too many categories (>100 unique values)
- One-hot encoded the simpler categorical stuff (≤100 unique values)
- Filled in missing values with median for numbers
- Scaled everything properly with StandardScaler

The Baseline Model

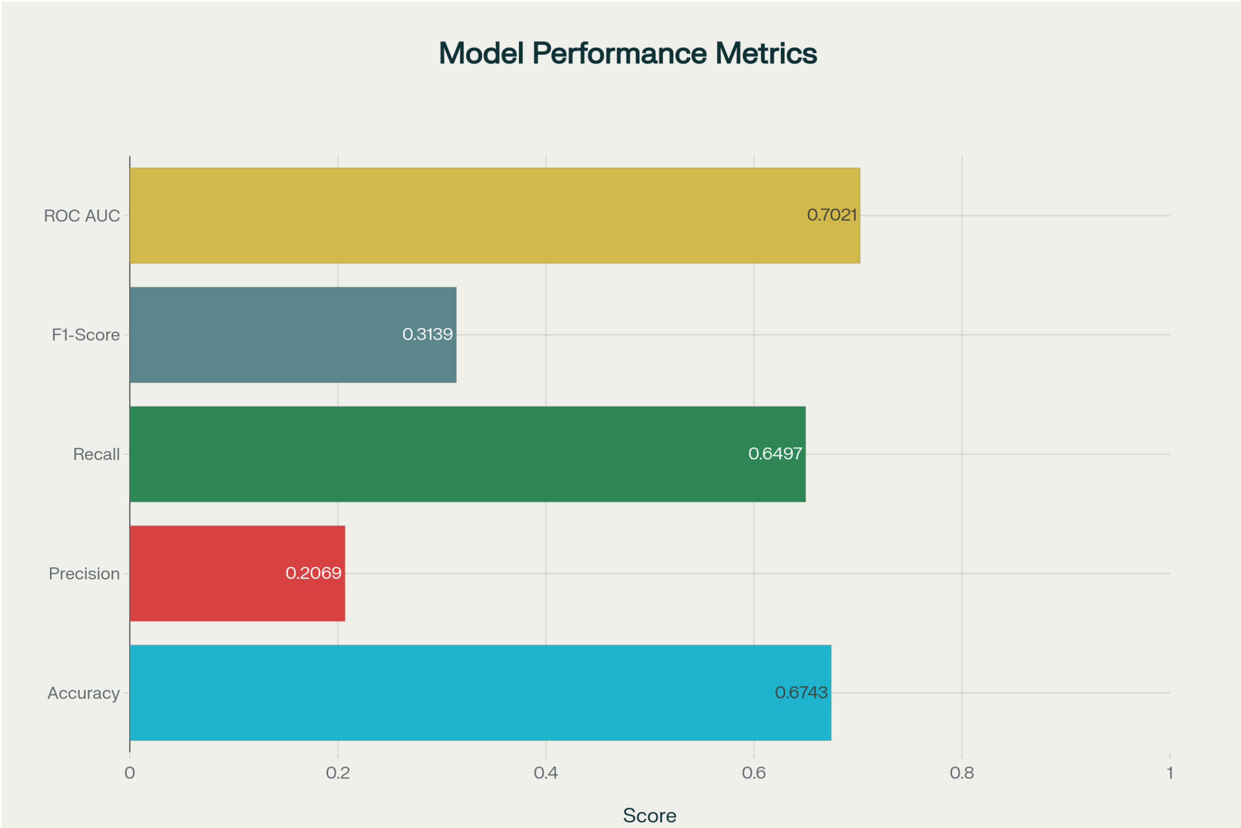
Logistic Regression Configuration:

- Algorithm: Logistic Regression (starting simple to establish baseline)
- Class weight: Balanced (to handle the imbalance)
- Solver: liblinear (works well for this type of problem)
- Max iterations: 1000

> **This is just the baseline - Random Forest is next on the list for experimentation!**

Results So Far

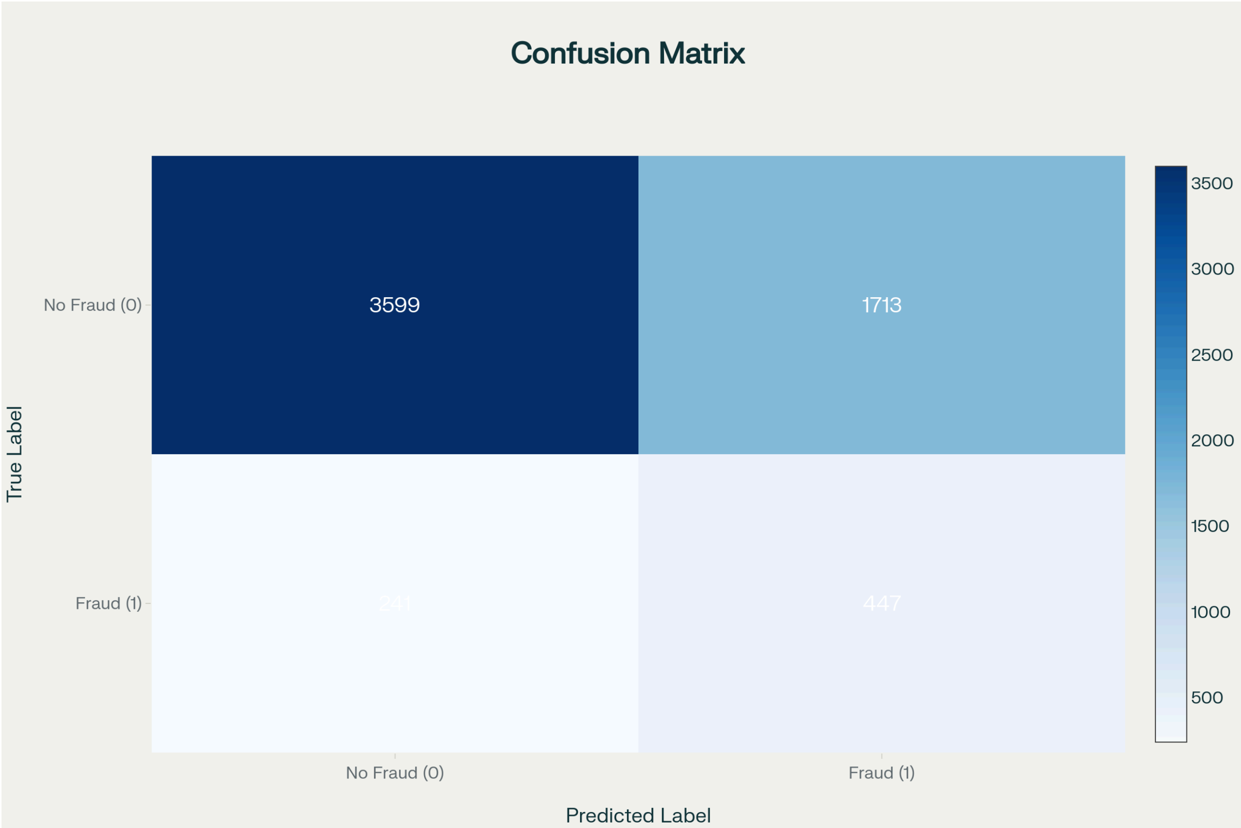
Model Performance Metrics



What the Numbers Say

- **Accuracy: 67%** - decent overall performance
- **Recall: 65%** - catching most fraud cases (that's the important one!)
- **ROC AUC: 0.70** - reasonable starting point for discrimination
- **Precision: 21%** - low, but that's okay for now in fraud screening

The Confusion Matrix Story



Breaking it down:

- ✓ Caught **447 fraud cases** correctly (True Positives)
- ✗ Missed **241 fraud cases** (False Negatives - need to improve this)
- ⚠ Flagged **1,713 as fraud** when they weren't (False Positives)
- ✓ Got **3,599 legitimate claims** right (True Negatives)

The recall is solid, but there's definitely room to improve!

Takeaways & Future Plans

What's Working

Current Strengths:

- The model's catching most fraud cases (65% recall) - not bad for a baseline
- Handled the class imbalance pretty well with balanced weights
- ROC score at 0.70 means it can distinguish fraud from legit claims
- Low precision means lots of false alarms, but better safe than sorry in fraud detection

What's Coming Next

Future Experiments:

1. **Random Forest** - should handle non-linear patterns better than logistic regression
2. **Threshold tuning** - optimize the balance between precision and recall
3. **Domain-specific features** - create new features based on insurance domain knowledge

4. **XGBoost exploration** - if Random Forest shows promise, try gradient boosting
5. **Deployment strategy** - set this up as a screening tool where humans review flagged cases

Project Notes

Most time went into finding and understanding this dataset - the modeling part was the fun part!

The journey of selecting the right dataset was crucial. Having quality data with the right features and sufficient examples made all the difference in building a meaningful baseline model.

This logistic regression model establishes our starting point. The focus now shifts to experimentation with more sophisticated algorithms that can capture complex, non-linear relationships in the data.

Presented by: Anshul | Project: Car Insurance Fraud Detection