Fraudulent Claim Deduction

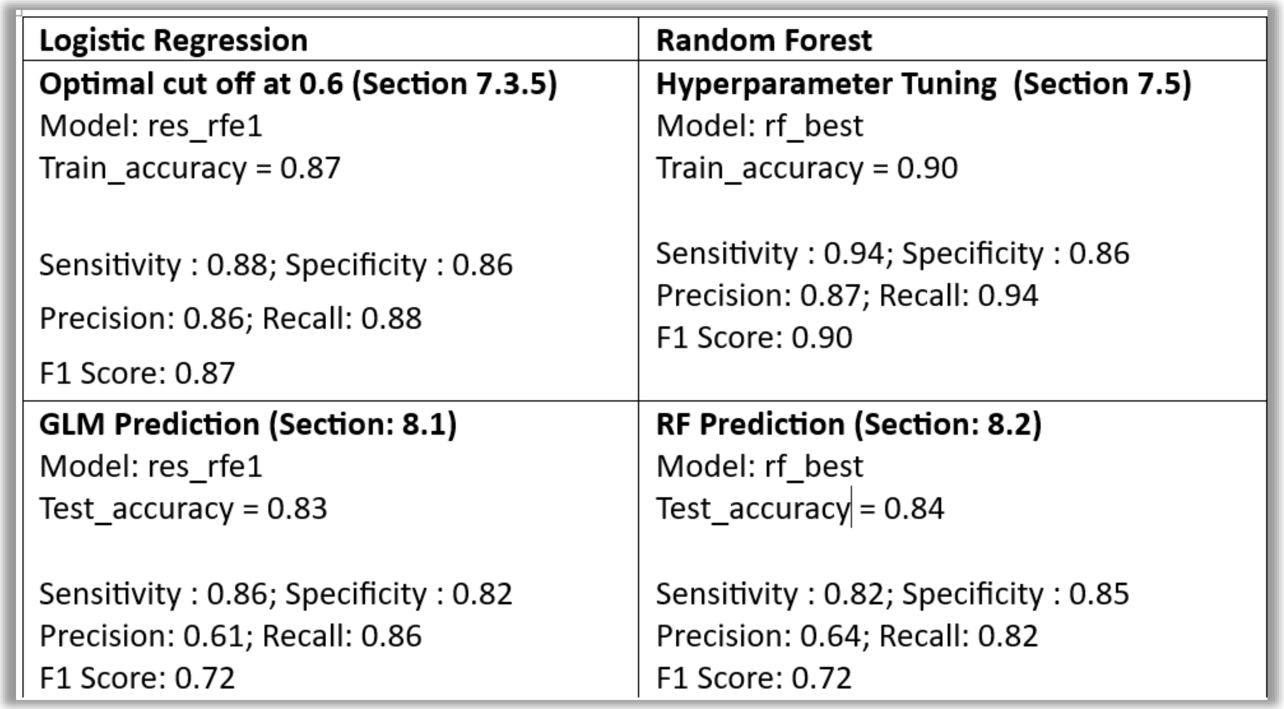
SUBMITTED BY (MLC - 074)

BALAMURUGAN CHANDRAN

DEVEN KARLA

Fraud Deduction Model Summary

Here is the summary of the model accuracy and score comparison



Logistic Regression	Random Forest
Optimal cut off at 0.6 (Section 7.3.5) Model: res_rfe1 Train_accuracy = 0.87 Sensitivity : 0.88; Specificity : 0.86 Precision: 0.86; Recall: 0.88 F1 Score: 0.87	**Hyperparameter Tuning (Section 7.5)** Model: rf_best Train_accuracy = 0.90 Sensitivity : 0.94; Specificity : 0.86 Precision: 0.87; Recall: 0.94 F1 Score: 0.90
GLM Prediction (Section: 8.1) Model: res_rfe1 Test_accuracy = 0.83 Sensitivity : 0.86; Specificity : 0.82 Precision: 0.61; Recall: 0.86 F1 Score: 0.72	**RF Prediction (Section: 8.2)** Model: rf_best Test_accuracy = 0.84 Sensitivity : 0.82; Specificity : 0.85 Precision: 0.64; Recall: 0.82 F1 Score: 0.72

Fraud Deduction Model Summary

Inference from Model

We have got very close result in logistic and random forest models. Both are close to 0.85 to .90 accuracy even in training and test data.

Test accuracy for logistic and random forest prediction is also almost same .83 and .84 respectively.

It will be a close call to take right decision. We are performing fraud deduction which means, identifying fraud person is very important though we include some false positive identified person. This implies **Sensitivity** should be higher

GLM Prediction sensitivity (0.86) is higher than **RF Prediction sensitivity** (0.81), so I suggest to go with ***LGR mode rse_rfe1 as best model for consideration***

How can we analyze historical claim data to detect patterns that indicate fraudulent claims?

Perform the below steps for building best model and predict the fraudulent claims.

1. Data Preparation – Load the data, create data frame and analyze the data using info() for empty, not null, total records, data types etc

2. Data Cleaning – Remove insignificant data (highly unique values), may be empty columns, minimal rows with empty values, handle redundant values, Fix data types

3. Train Validation Split 70-30 – Using train_test_split split the data into desired portion (May be70/30 here) with random_state=42 so that data split is same for every run. 70% data is used for training the data. 30% data for evaluating the model

4. EDA on Training Data – Perform univariate analysis and make sure numerical and categorical data is making sense or change the data types to appropriate ones. Visualize the distribution of data using histplot / boxplot / countplot to try to get inference on the data. Perform correlation analysis, bivariate analysis of target likelihood and coeff for categorical variables, thus helps to reduce insignificant feature data

5. Feature Engineering - RandomOverSampler technique to balance the data and handle class imbalance. Create new features from existing ones to enhance the model's ability to capture patterns in the data. Create dummy variables for categorical features. Apply StandardScaler to numerical features. Use RFE to get more significant features explaining the target

How can we analyze historical claim data to detect patterns that indicate fraudulent claims?

Continues….

6. Model Building

Logistic Regression Model

Feature Selection using RFECV – Identify the most relevant features using Recursive Feature Elimination with Cross-Validation. Model Building and Multicollinearity Assessment – Build the logistic regression model and analyse statistical aspects such as p-values and VIFs to detect multicollinearity.

Model Training and Evaluation on Training Data – Fit the model on the training data and assess initial performance.

Finding the Optimal Cutoff – Determine the best probability threshold by analysing the sensitivity-specificity tradeoff and precision-recall tradeoff.

Final Prediction and Evaluation on Training Data using the Optimal Cutoff – Generate final predictions using the selected cutoff and evaluate model performance.

Random Forest Model

Get Feature Importances - Obtain the importance scores for each feature and select the important features to train the model.

Model Evaluation on Training Data – Assess performance metrics on the training data.

Check Model Overfitting using Cross-Validation – Evaluate generalization by performing cross-validation.

Hyperparameter Tuning using Grid Search – Optimize model performance by fine-tuning hyperparameters.

Final Model and Evaluation on Training Data – Train the final model using the best parameters and assess its performance.

How can we analyze historical claim data to detect patterns that indicate fraudulent claims?

Continues….

7. Predicting and Model Evaluation

Evaluate the model's performance using metrics such as accuracy, sensitivity, specificity, precision, and recall.

Make predictions over validation data using logistic regression model & random forest models created

Compare the metrics to choose the optimal best model

We are performing fraud deduction which means, identifying fraud person is very important though we include some false positive identified claim. This implies Sensitivity should be higher

Use the final best optimized model to predict the fraudulent claims and help insurance company make the right decision

Which features are the most predictive of fraudulent behavior?

Already in the summary concluded as GLM model with optimal cut off with 0.6 (res_rfe1) is chosen to be best. Let's take a look at the features and its coefficients. Highlighted most predictive features

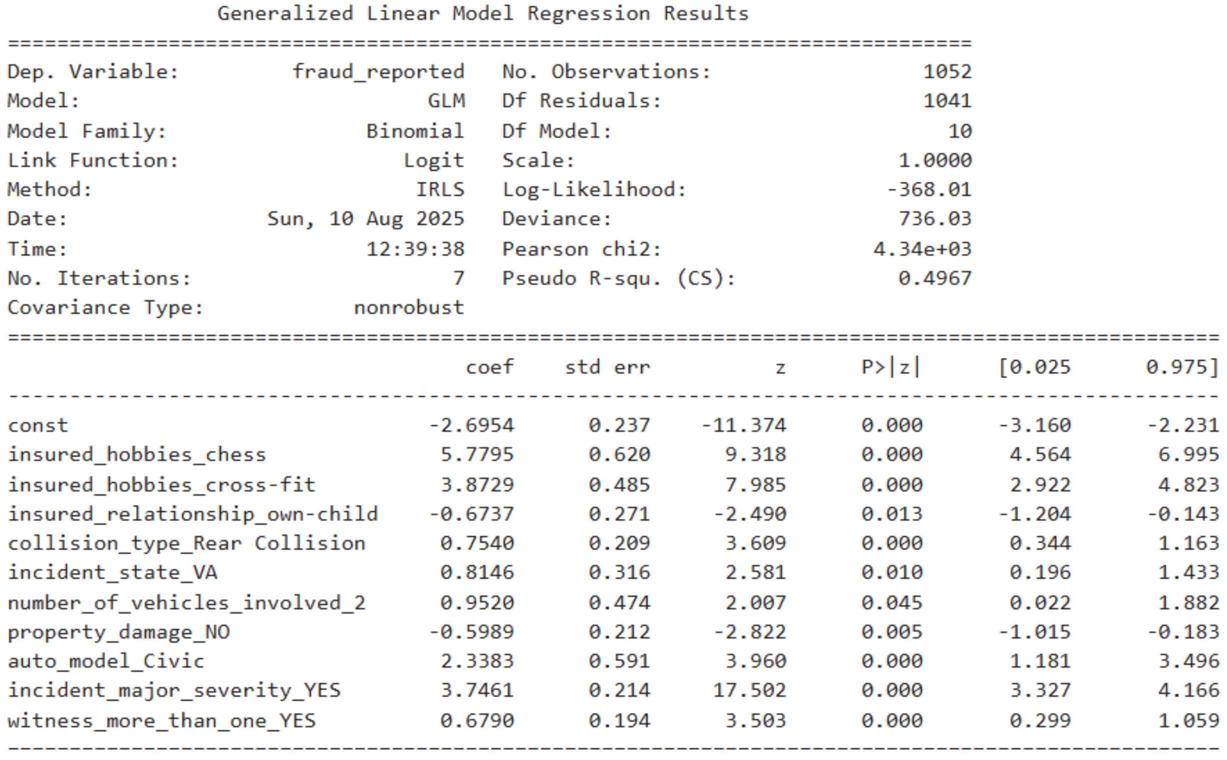
High Positive Correlated Features

- Insured_hobbies_chess
- Insured_hobbies_cross-fit
- Insured_major_severity_YES
- auto_model_civic
- number_of_vehicles_involved_2 (Note: coeff > .9)
- Incident_state_VA
- Collision_type_Rear Collision
- Witness_more_than_one_YES

High Negative Correlated Features

Insured_relationship_own-child

Property_damage_NO



```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:         fraud_reported   No. Observations:                 1052
Model:                            GLM   Df Residuals:                     1041
Model Family:                Binomial   Df Model:                           10
Link Function:                  Logit   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -368.01
Date:                Sun, 10 Aug 2025   Deviance:                       736.03
Time:                        12:39:38   Pearson chi2:                 4.34e+03
No. Iterations:                     7   Pseudo R-squ. (CS):             0.4967
Covariance Type:            nonrobust
===================================================================================================
                                      coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------------------
const                              -2.6954      0.237    -11.374      0.000      -3.160      -2.231
insured_hobbies_chess               5.7795      0.620      9.318      0.000       4.564       6.995
insured_hobbies_cross-fit           3.8729      0.485      7.985      0.000       2.922       4.823
insured_relationship_own-child     -0.6737      0.271     -2.490      0.013      -1.204      -0.143
collision_type_Rear Collision       0.7540      0.209      3.609      0.000       0.344       1.163
incident_state_VA                   0.8146      0.316      2.581      0.010       0.196       1.433
number_of_vehicles_involved_2       0.9520      0.474      2.007      0.045       0.022       1.882
property_damage_NO                 -0.5989      0.212     -2.822      0.005      -1.015      -0.183
auto_model_Civic                    2.3383      0.591      3.960      0.000       1.181       3.496
incident_major_severity_YES         3.7461      0.214     17.502      0.000       3.327       4.166
witness_more_than_one_YES           0.6790      0.194      3.503      0.000       0.299       1.059
---------------------------------------------------------------------------------------------------
```

Based on past data, can we predict the likelihood of fraud for an incoming claim?

Yes, we can apply the GLM model res_rfe1 and predict the fraud for an incoming claim

```
# Select the relevant features for validation data
  ◦ X_test_best = X_test1[col]

# Add constant to X_validation
  ◦ X_test_sm = sm.add_constant(X_test_best)


  y_validation_pred = res_rfe1.predict(X_test_sm)

predictions = pd.DataFrame({"Test_actual": y_test, "Validation": y_validation_pred})predictions

metrics.accuracy_score(predictions["Test_actual"], predictions["Predicted"])
```

What insights can be drawn from the model that can help in improving the fraud detection process?

Based on the high significant features identified,

- Insurer with hobbies, **chess** or **cross-fit** have more high chance of applying for fraudulent claim
- Fraudulent claim is likely applying for **Major severity** to get most benefit of claim amount
- Auto model **civic** is another high significance of factor to be a fraud
- Fraudulent claim is applied mostly with **2 vehicles involved**
- Fraud claim comes more from **VA** state
- Fraud claim mostly applies with **Rear collision** type
- **Witness** is likely to be **more than 1** for a fraudulent claim
- Insured being **own child, fraudulent claim is likely to be less**
- Most fraudulent claim applied with **property damage Yes** to get most benefit of claim amount