Fraud Detection: How can Machine Learning Help?

Navarun Jain
Lux Actuaries & Consultants
Dubai
Agenda

Insurance Fraud – the Issue
ML as a Concept
Tree-based ML Algorithms
• CART
• C5.0
• Gradient Boosting Machines
• Random Forests
Non-tree-based ML Algorithms
• Neural Networks
Applications to Fraud Detection – A Simple Case Study
• Data Description
• Models and Performance Evaluation
• Interpretation
Key Takeaways and Conclusions
Insurance Fraud
Insurance Fraud
Insurance Fraud

• Becoming a rapidly growing issue worldwide
Insurance Fraud

• Becoming a rapidly growing issue worldwide
• UK fraud activity reached an estimated £17 million in 2018
Insurance Fraud

• Becoming a rapidly growing issue worldwide
• UK fraud activity reached an estimated £17 million in 2018
• Biggest lines are Motor, Medical and Workmen’s Compensation – fake car crashes, personal injury scams, faked death claims
Insurance Fraud

• Becoming a rapidly growing issue worldwide

• UK fraud activity reached an estimated £17 million in 2018

• Biggest lines are Motor, Medical and Workmen’s Compensation – fake car crashes, personal injury scams, faked death claims

• With advancing technology, it can become easier to detect fraudulent claims when they are received
Machine Learning
The "teaching a kid math" analogy
The Roadmap

All about patterns!!!

Computer systems learn from data

We train the system → System learns → Then performs operations on its own
The Roadmap

All about patterns!!!

Computer systems **learn**
from data

We **train** the system → System **learns** → Then performs operations **on its own**

Training phase 1: data is fed into the algorithm, relevant fields and records sorted from data to retrieve **active dataset**

02 September 2019
All about patterns!!!

Computer systems learn from data

We train the system

System learns

Then performs operations on its own

Training phase 2: Model Fitting – algorithm decodes hidden patterns and relationships in the data
All about patterns!!!

Computer systems learn from data

We train the system → System learns → Then performs operations on its own

Testing phase: new data fed into system, algorithm uses patterns & relationships learnt during the training phase to predict new cases
Types of Algorithms
With ML, no need to…
With ML, no need to…

• …make assumptions about distributions
• …worry about possible correlations between predictors
• …look for interactions between predictors
How can ML help?

<table>
<thead>
<tr>
<th>RULE-BASED FRAUD DETECTION</th>
<th>ML-BASED FRAUD DETECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can catch obvious and known fraud scenarios only</td>
<td>Can find not-so-obvious fraud scenarios due to the ability to detect hidden patterns/correlations in data</td>
</tr>
<tr>
<td>Requires manual work to determine criteria for fraud scenarios</td>
<td>Can automatically detect and create rules for fraud scenarios</td>
</tr>
<tr>
<td>Longer processing and verification times due to manual nature</td>
<td>Quicker processing and verification times since algorithms are automatically generated and verified</td>
</tr>
</tbody>
</table>
Tree-Based ML Algorithms
Decision Trees

Model is grown by recursively splitting the data into decision boundaries using the feature space.
Types of Decision Tree Algorithms

SINGLE TREE MODELS
1. CART
2. C5.0

ENSEMBLE MODELS
1. GBM
2. RANDOM FOREST
Single Tree Models
Creating a Decision Tree

Data is split in a way that **maximizes** the gain in information between *parent* and *child* nodes.

Goal is to split data points in a way that makes the subgroups as homogenous as possible.
Measuring Information Gain
Measuring Information Gain

**Gini Impurity**

\[
Gini(t) = \sum_{k=1}^{h} p_k (1 - p_k)
\]

- \(p_k\) – Probability of choosing item with label \(k\) in set \(t\)

Measures how often a randomly chosen element would be incorrectly labeled if it were labeled according to its distribution in the data

Used as splitting criterion for the CART algorithm

**Entropy**

\[
H(t) = - \sum_{k=1}^{h} \{ p_k \log_b p_k \}
\]

- \(p_k\) – Probability of choosing item with label \(k\) in set \(t\)
- \(b\) – Logarithmic base

Measures how “mixed up” the data is

Used as splitting criterion for the C5.0 algorithm
Ensemble Learning Models
Gradient Boosting Machines
Boosting

• Converts weak learners into a single strong learner by aggregating them
Boosting

• Converts weak learners into a single strong learner by aggregating them
Random Forests
Breaking Down the “Random Forest”
Breaking Down the “Random Forest”

• RF based on the concept of Bagging (Bootstrap Aggregating)
Breaking Down the “Random Forest”

- RF based on the concept of **Bagging (Bootstrap Aggregating)**

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Time stamp</th>
<th>ID</th>
<th>Age group</th>
<th>Fresh fruit</th>
<th>Seafood</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>April</td>
<td>1</td>
<td>Senior</td>
<td>Orange</td>
<td>Tuna</td>
</tr>
<tr>
<td>2</td>
<td>April</td>
<td>2</td>
<td>Teenager</td>
<td>Banana</td>
<td>Tuna</td>
</tr>
<tr>
<td>3</td>
<td>April</td>
<td>3</td>
<td>Teenager</td>
<td>Banana</td>
<td>Tuna</td>
</tr>
<tr>
<td>4</td>
<td>June</td>
<td>1</td>
<td>Senior</td>
<td>Orange</td>
<td>Anchovy</td>
</tr>
<tr>
<td>5</td>
<td>June</td>
<td>3</td>
<td>Adult</td>
<td>Banana</td>
<td>Tuna</td>
</tr>
<tr>
<td>6</td>
<td>July</td>
<td>1</td>
<td>Senior</td>
<td>Orange</td>
<td>Anchovy</td>
</tr>
<tr>
<td>7</td>
<td>July</td>
<td>2</td>
<td>Teenager</td>
<td>Banana</td>
<td>Tuna</td>
</tr>
<tr>
<td>8</td>
<td>July</td>
<td>2</td>
<td>Teenager</td>
<td>Orange</td>
<td>Tuna</td>
</tr>
<tr>
<td>9</td>
<td>July</td>
<td>4</td>
<td>Teenager</td>
<td>Banana</td>
<td>Tuna</td>
</tr>
<tr>
<td>10</td>
<td>December</td>
<td>2</td>
<td>Teenager</td>
<td>Banana</td>
<td>Tuna</td>
</tr>
<tr>
<td>11</td>
<td>December</td>
<td>3</td>
<td>Adult</td>
<td>Orange</td>
<td>Anchovy</td>
</tr>
<tr>
<td>12</td>
<td>December</td>
<td>1</td>
<td>Senior</td>
<td>Orange</td>
<td>Anchovy</td>
</tr>
</tbody>
</table>

Random sample of $p$ columns

Repeat $N$ times

Random sample of $k$ rows

**Diagram:**

- $x_5 = \text{True}$
  - Yes
    - $y = \text{True}$
    - $y = \text{False}$
  - No
    - $y = \text{False}$
Artificial Neural Networks
Structured Sequential model

**Structured**: A Neural Network has a defined structure that consists of 3 types of layers

**Sequential**: Information flows in a sequence from one layer to the next, undergoing operations at each layer – almost like an assembly line
How ANN’s Work
How ANN’s Work

![Diagram showing how ANNs work](image)

1. **Input Signals**
   - $x_1$, $x_2$, ..., $x_m$

2. **Weights**
   - $w_{k1}$, $w_{k2}$, ..., $w_{km}$

3. **Bias**
   - $b_k$

4. **Summation**
   - $\sum$

5. **Activation Function**
   - $\varphi(.)$

6. **Output**
   - $y_k$

The diagram illustrates how input signals are weighted, summed, and passed through an activation function to produce an output.
How ANN’s Work

• Data in every neuron is transformed by an activation function:

\[ h_k(x) = g(\beta_{0k} + \sum_{i=1}^{n} x_i \beta_{ik}) \]

- \( h_k(x) \) – \( k^{th} \) neuron in a hidden layer
- \( \beta_{ik} \) – coefficient of the \( i^{th} \) previous-layer neuron on above neuron
How ANN’s Work

• Data in every neuron is transformed by an activation function:

\[ h_k(x) = g(\beta_{0k} + \sum_{i=1}^{n} x_i \beta_{ik}) \]

- \( h_k(x) \) – \( k^{th} \) neuron in a hidden layer
- \( \beta_{ik} \) - coefficient of the \( i^{th} \) previous-layer neuron on above neuron

• Activation function transforms the linear combination of inputs from one layer and sends it to the next layer.
How ANN’s Work
How ANN’s Work

At first, each neuron is randomly assigned a weight – this measures the contribution of that neuron to the next layer.
How ANN’s Work

Data flows through network, predicted values calculated
How ANN’s Work

Predictions are compared with actuals based on a loss function
How ANN’s Work

Weights are updated to reduce value of loss function
Case Study: Classifying Motor Insurance Fraud
The Data

• Claim-level information with an indicator for whether a claim was flagged as a fraud or not

• Data points for each claim include –
  – Driver demographics (age, marital status, gender)
  – Vehicle information (age, price, body type, country or origin)
  – Policy information (policy cover type, number of vehicles insured, deductible, agent type)
  – Accident/Claim information (when was the claim filed, whether there were witnesses present during the accident, party at fault, whether a police report was filed)
Summary of Results

• GBM, Random Forest performed best, followed by Neural Networks
• C5.0, CART poor
• Logistic Regression did not perform well
• Driver Age, Policy Type, Fault, Past Number of Claims most important predictors of fraudulent behavior
• Details in following slides
The Data
The Data

![Bar Chart]

- **Fault: Policy Holder**
- **Fault: Third Party**

Count vs. FraudFound (0 and 1)
The Data

[Graph showing the count of fraud found in vehicle price categories: Vehicle Price: < 20k, Vehicle Price: 20k-29k, Vehicle Price: 30k-39k, Vehicle Price: 40k-59k, Vehicle Price: 60k-69k, Vehicle Price: > 69k]
Models
Models

- Data split 75-25 for training and validation
- C5.0 trained using standard algorithm
- CART pruned using cost-complexity
- GBM, Random Forest and Neural Networks tuned using Cartesian Hyperparameter Grid Search
Grid Search Example – H2O

H2O Grid Details

Grid ID: gbm_grid
Used hyper parameters:
- col_sample_rate
- learn_rate
- max_depth
- ntree
- sample_rate
Number of models: 144
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by decreasing f1

<table>
<thead>
<tr>
<th>col_sample_rate</th>
<th>learn_rate</th>
<th>max_depth</th>
<th>ntree</th>
<th>sample_rate</th>
<th>model_ids</th>
<th>f1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>0.1</td>
<td>25</td>
<td>5000</td>
<td>gbm_grid_model_66</td>
<td>0.26259541984732826</td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
<td>0.1</td>
<td>25</td>
<td>2000</td>
<td>gbm_grid_model_54</td>
<td>0.26259541984732826</td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
<td>0.1</td>
<td>25</td>
<td>8000</td>
<td>gbm_grid_model_78</td>
<td>0.26259541984732826</td>
</tr>
<tr>
<td>4</td>
<td>1.0</td>
<td>0.1</td>
<td>25</td>
<td>10000</td>
<td>gbm_grid_model_90</td>
<td>0.26259541984732826</td>
</tr>
<tr>
<td>5</td>
<td>0.8</td>
<td>0.1</td>
<td>25</td>
<td>10000</td>
<td>gbm_grid_model_89</td>
<td>0.2612085769980507</td>
</tr>
</tbody>
</table>
---

<table>
<thead>
<tr>
<th>col_sample_rate</th>
<th>learn_rate</th>
<th>max_depth</th>
<th>ntree</th>
<th>sample_rate</th>
<th>model_ids</th>
<th>f1</th>
</tr>
</thead>
<tbody>
<tr>
<td>139</td>
<td>0.8</td>
<td>0.1</td>
<td>60</td>
<td>5000</td>
<td>gbm_grid_model_23</td>
<td>0.23591549295774647</td>
</tr>
<tr>
<td>140</td>
<td>0.8</td>
<td>0.1</td>
<td>60</td>
<td>2000</td>
<td>gbm_grid_model_11</td>
<td>0.23591549295774647</td>
</tr>
<tr>
<td>141</td>
<td>1.0</td>
<td>0.1</td>
<td>10</td>
<td>5000</td>
<td>gbm_grid_model_15</td>
<td>0.233983286908078</td>
</tr>
<tr>
<td>142</td>
<td>1.0</td>
<td>0.1</td>
<td>10</td>
<td>2000</td>
<td>gbm_grid_model_3</td>
<td>0.233983286908078</td>
</tr>
<tr>
<td>143</td>
<td>1.0</td>
<td>0.1</td>
<td>10</td>
<td>8000</td>
<td>gbm_grid_model_27</td>
<td>0.233983286908078</td>
</tr>
<tr>
<td>144</td>
<td>1.0</td>
<td>0.1</td>
<td>10</td>
<td>10000</td>
<td>gbm_grid_model_39</td>
<td>0.233983286908078</td>
</tr>
</tbody>
</table>
Model Performance

• Evaluated using the following criteria
  – Accuracy
  – AUC
  – F1 Score

• All metrics based on **Confusion Matrix**

• AUC also related to **Receiver Operating Characteristics (ROC) Curve**
Model Performance

Accuracy = \frac{TP+TN}{TP+FP+FN+TN}

Precision = \frac{TP}{TP+FP}

Recall = \frac{TP}{TP+FN}

F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
Model Performance

**ROC Curve**: Plots True Positive Rate vs. False Positive Rate at different probability thresholds

**AUC**: Area under ROC Curve

Measure of how well can a model distinguish between 2 classes
Model Performance

• Evaluated using the following criteria
  – Accuracy
  – AUC
  – F1 Score

• All metrics based on **Confusion Matrix**

• AUC also related to **Receiver Operating Characteristics (ROC) Curve**
Model Performance

![Model Performance Chart]

The chart shows the accuracy of different models: C5.0, GBM, Neural Network, RF, Logistic Regression, and CART. The accuracy ranges from 60 to 100, with C5.0 and GBM achieving the highest accuracy, followed by Neural Network and RF. Logistic Regression and CART have similar accuracy, which is lower than the other models.
Model Performance

![Graph showing model performance comparison]

- GBM
- RF
- Neural Network
- CART Logistic Regression
- C5.0

F1 scores are displayed on the y-axis, with each model's performance relative to the others.
Model Performance

- RF
- GBM
- Neural Network
- CART Logistic Regression
- CS:O

AUC
Variable Importance

Variable Importance: GBM

- DriverAge
- BasePolicy
- Fault
- PastNumberOfClaims
- NumberOfSupplements
- AgeOfVehicle
- Deductible
- IncidentYear
- VehiclePrice
- VehOrigin
- DriverMaritalStatus
- DriverGender
- NumberOfCars
- VehicleCategory
- PoliceReportFiled
- DaysPolicyAccident
- AgentType
- DaysPolicyClaim
- WitnessPresent

0.0 0.2 0.4 0.6 0.8 1.0
Variable Importance

Variable Importance: Deep Learning

PastNumberOfClaims, more than 4
BasePolicy Liability
NumberOfCars, 2 vehicles
Deductible, 700
NumberOfSupplements, 1 to 2
VehiclePrice, more than 69000
DriverAge
PastNumberOfClaims, none
NumberOfSupplements, more than 5
BasePolicy, Collision
AgeOfVehicle, 7 years
PastNumberOfClaims, 2 to 4
VehiclePrice, less than 20000
VehOrigin, USA
PoliceReportFiled, Yes
VehOrigin, Europe
IncidentYear, 1995
BasePolicy, All Perils
VehOrigin, Germany
AgentType, Internal
NumberOfCars, 3 to 4
WitnessPresent, Yes
DaysPolicyAccident, 15 to 30
DriverMaritalStatus, Married
DaysPolicyAccident, none
Key Takeaways & Conclusions
Conclusions

• ML can be a powerful tool

• Results from classification models could be used to proactively flag claims as fraudulent and minimize unnecessary losses

• Models can also help understand customer behavior, eg. which groups contribute most to insurance fraud
The views expressed in this presentation are those of invited contributors and not necessarily those of the IFoA. The IFoA do not endorse any of the views stated, nor any claims or representations made in this presentation and accept no responsibility or liability to any person for loss or damage suffered as a consequence of their placing reliance upon any view, claim or representation made in this presentation.

The information and expressions of opinion contained in this publication are not intended to be a comprehensive study, nor to provide actuarial advice or advice of any nature and should not be treated as a substitute for specific advice concerning individual situations. On no account may any part of this presentation be reproduced without the written permission of the author.