Detecting Fraud in Insurance Claims

Technical details

Analysis by: Manchala Krishna Kumar

Da

Dataset is on car insurance claims. Dataset provides details about customer, insurance policy, incident and cost.

- Q1 2015
- 1000 incidents (only)
- 1 record per incident
- 40 features

Use Case

Insurance companies handle lots of claims, some of which are fraudulent. Detecting fraudulent claims before payment would enable company to act and by doing so, company could improve efficiency and reduce cost.

Scientific approach can help the insurance company to:

1. **Explore influential factors** that correlate with fraudulent claims.

2. **Predict fraudulent claims** in automated way using Machine Learning algorithm.

Technology and Plaform

- **Python** with with data science libraries (numpy, pandas, seaborn, scikit-learn and keras)
- **IBM Watson Cloud** with **jupyter** notebooks
- Export from data warehouse in **.csv** format

Data Quality Assessment

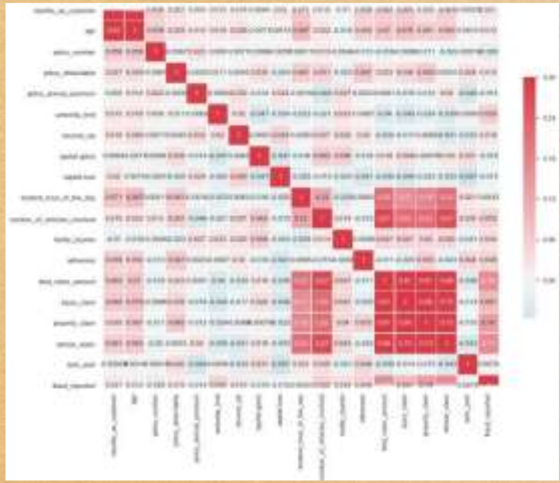
Numerical analysis + visualizations with **pandas** and **seaborn**.

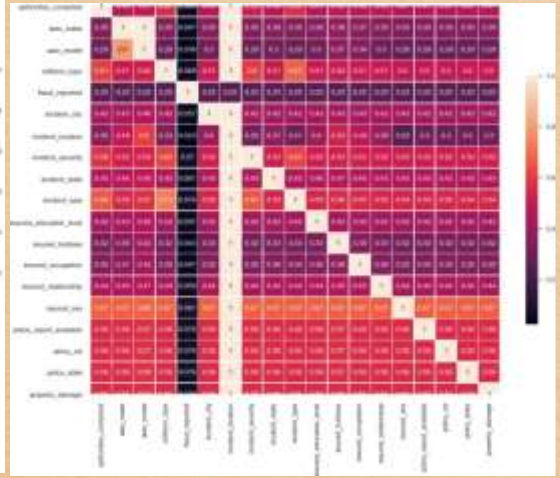
- **Data Types**
  Conversion and validation of Datetime fields, object to categories
- **Ranges**
  Numerical variables range check with *df.describe().* Non-negative claim amounts, valid age ranges, valid auto years, ..
- **Emptiness**
  Checking for null entries, dropping columns or rows, identifying unknown vs. missing values.
- **Uniqueness**
  Are duplicates present where undesired? E.g. incident IDs
- **Set memberships**
  Are only allowed values chosen for categorical or ordinal fields? E.g. sex, occupation, hobbies, relationship, state, city, …
- **Regular Expressions**
  State abbreviation must have 2 letters only

Data Quality Assessment cont.

Between variable correlation analysis

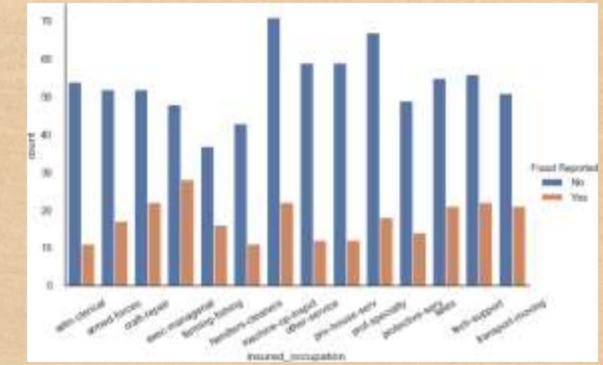
- Pearson, Cramer's V

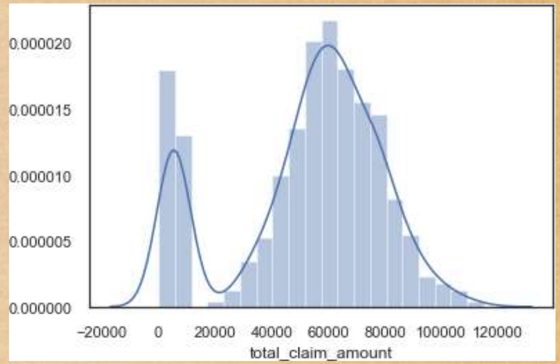


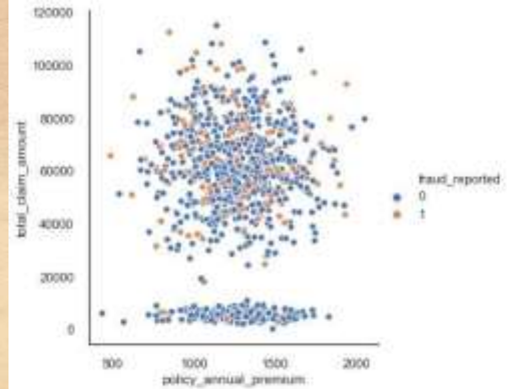


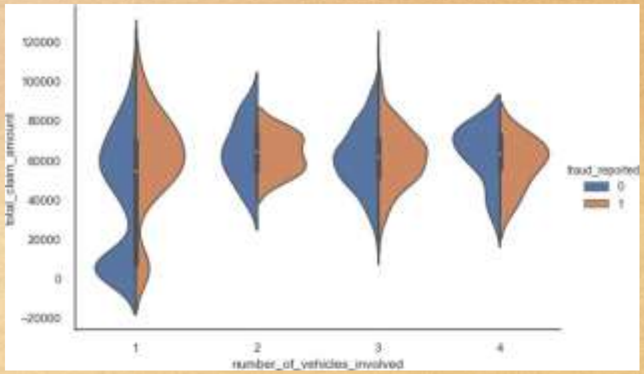
Within variable distribution to target correlation

- Histograms, scatterplots, boxplots, violinplots









Data Transformation and Feature Extraction

- **Filtering**
  Dataset is consistent and pre-cleaned since it originates from DWh.
- **Discretizing**
  Based on EDA and feature correlation to target variable, extracted *age_groups* and *risky_hobbies*
- **Normalizing**
  Center numerical features around zero and scale values to a standard deviation of one
- **One-hot-encoding**
  After reducing the categorical dimensionality, OHE all categorical features
- **Parts of Date**
  Creating an additional features containing the *month*, *week* and *weekend indicator*

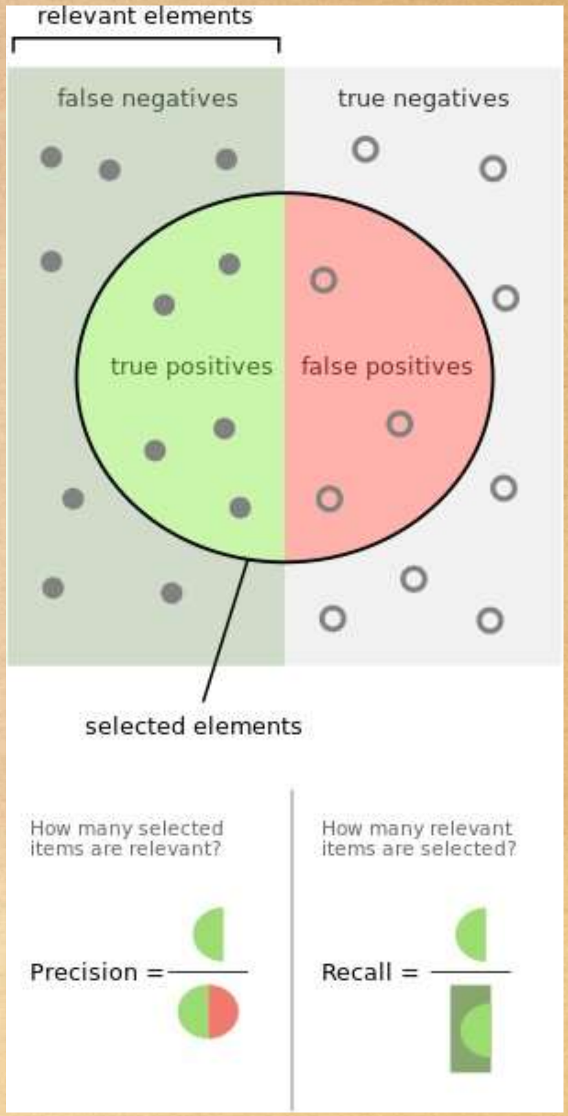
Performance Metric

We are dealing with binary classification and an unbalanced dataset, so the chosen evaluation metric is **f1-score**.

- f1-score considers both the *precision p* and the *recall r* of the test to compute the score.
- This makes it a very robust measure in which false positives and false negatives are penalized.

70/30 split to **Train** and **Test** dataset

- Models are trained using **cross-validation** with goal to optimize val_f1score.
- Final performance is measured on test dataset and f1score for positive values (detecting fraud)



Tested Algorithms

Model	CV val_f1-score	Comment
Majority class prediction	0.00	Predicting all cases are not fraud. No value from this model.
Logistic Regresion	0.03	Not useful at all.
Decision Tree	0.56	This class of models is suited for the problem. Bias toward training set.
Random Forests	0.32	Too much noise from week learners.
LightGMB	**0.65**	Big step forward. Generalizes really well. f1-score on unseen data **0.64**
XGBoost	0.64	Similar performance. Generalizes really well.
Neural Network, no tuning	0.51	1 hidden layer, # neurons (95 or 45), no Dropout
Neural Network, tuned	**0.72**	1 hidden layer, 80 neurons, Dropout (0.1), uniform initializer, logcosh loss, Nadam optimizer f1-score on unseen data **0.63**

LightGBM

Summary

- Best performing model is **LightGBM** with **f1-score**
- **0.64** for detecting fraud on unseen data.
  - 'boosting': 'gbdt', 'num_leaves': 30,
  - 'feature_fraction': 0.5,
  - 'bagging_fraction': 0.5,
  - 'bagging_freq': 20,
  - 'learning_rate': 0.05,

Model is trained with clean dataset with most influential features extracted.

Classification report

	precision	recall	f1-score	support
0	0.88	0.86	0.87	220
1	0.63	0.66	0.65	80
micro avg	0.81	0.81	0.81	300
macro avg	0.75	0.76	0.76	300
weighted avg	0.81	0.81	0.81	300

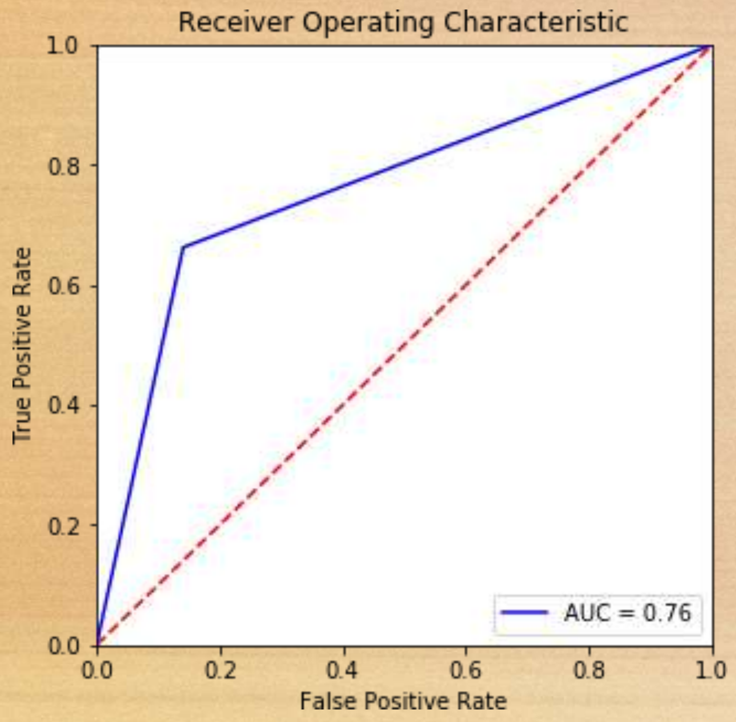
Confusion matrix:

```
[[189  31]
 [ 27  53]]
```

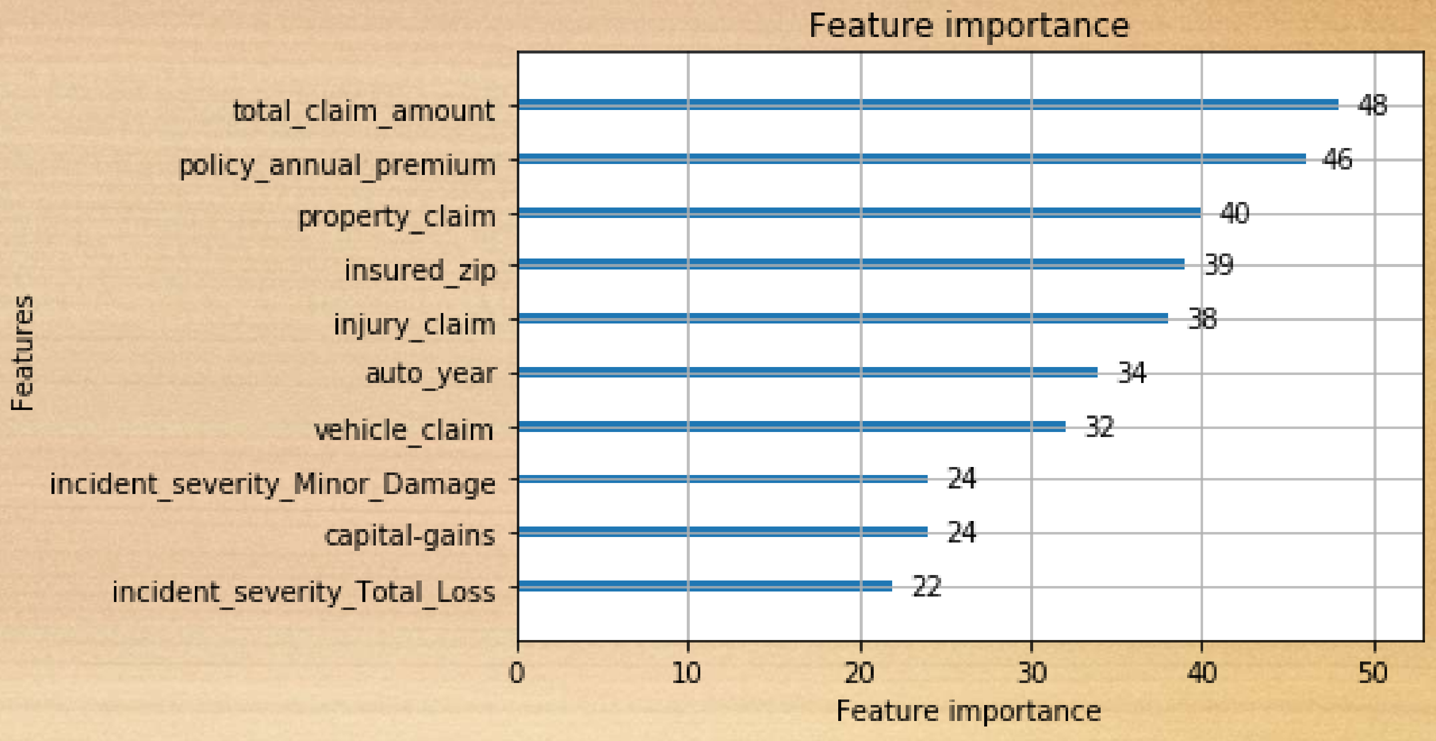
LightGBM

Performance curve and important features

Area under the curve is **0.76**



Most important features



Neural Network

Summary

Best performing model is **fully connected, single hidden layer NN** with **f1-score 0.63** for detectin g fraud on unseen data.

- ‘learning rate’: 1.5
- 'first_neuron’: 80
- 'batch_size’: 23
- 'epochs’: 250
- 'dropout’: 0.1,
- 'kernel_initializer': uniform
- 'optimizer': Nadam
- 'losses': logcosh
- 'activation': elu,
- 'last_activation': sigmoid

Classification report

	precision	recall	f1-score	support
0	0.86	0.87	0.87	220
1	0.64	0.62	0.63	80
micro avg	0.81	0.81	0.81	300
macro avg	0.75	0.75	0.75	300
weighted avg	0.81	0.81	0.81	300

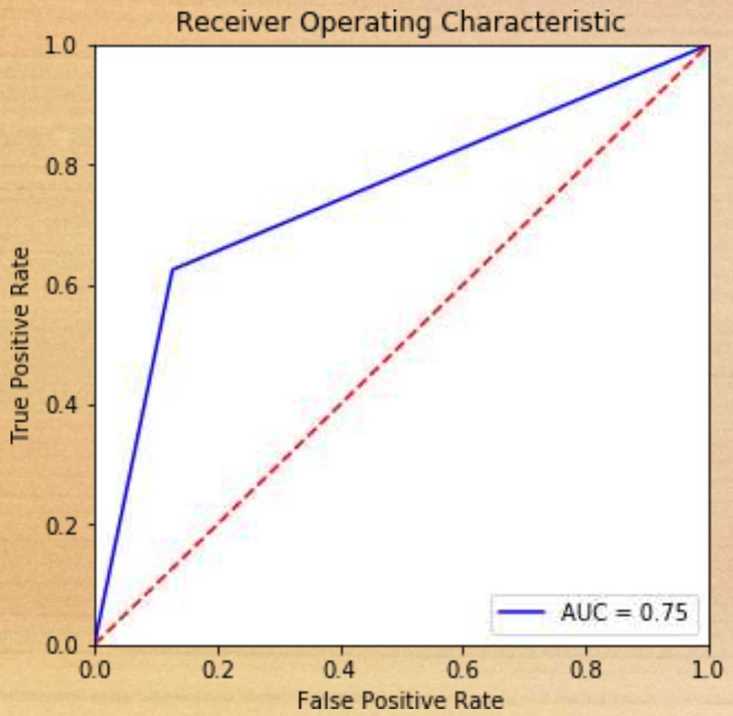
Confusion matrix:

```
[[192  28]
 [ 30  50]]
```

Neural Network

Performance curve

Area under the curve is **0.75**



Meaning that model gives *76% probability* that a randomly chosen *positive instance (fraud) is ranked higher* than a randomly chosen *negative instance* (not fraud).

Using the model

Trained model can be used when new insurance claim is received.

1. Raw data is collected and stored.
2. Script for data cleaning and transformation is run on raw data.
3. Prepared data is fed into the pre-trained model.
4. Model returns a probability that claim is fraudulent.
5. This information can be used to adjust further steps in processing the claim.

Thank you