IBM Advanced Data Science Specialization

Capstone Project

Filip Gvardijan

Date: 30.10.2019

Data set - fraud detection

Data set is on auto insurance claims (link).

- Q1 2015
- 1000 claims (only)
- 1 record per claim
- 40 features
- Fraud is labeled

Data set provides details about customer, insurance policy, incident and cost.

Use case – binary classification

Early detection of fraudulent claims enables company to act.

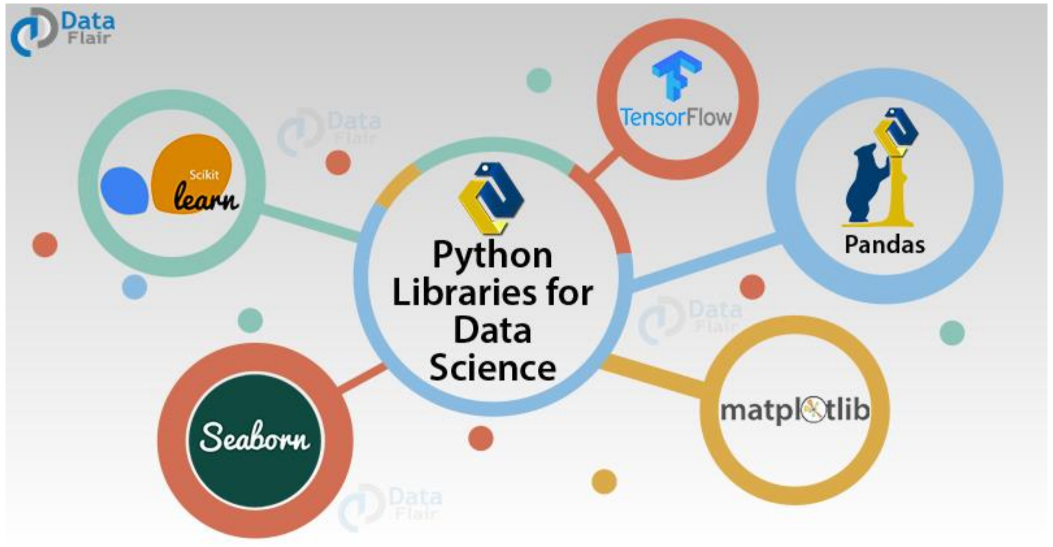
- Improve efficiency in handling claims.
- Reduce cost of fraud.

Scientific approach can help the insurance company to:

- **Explore influential factors** that correlate with fraudulent claims.
- **Predict fraudulent claims** in automated way using Machine Learning algorithms.

Technology

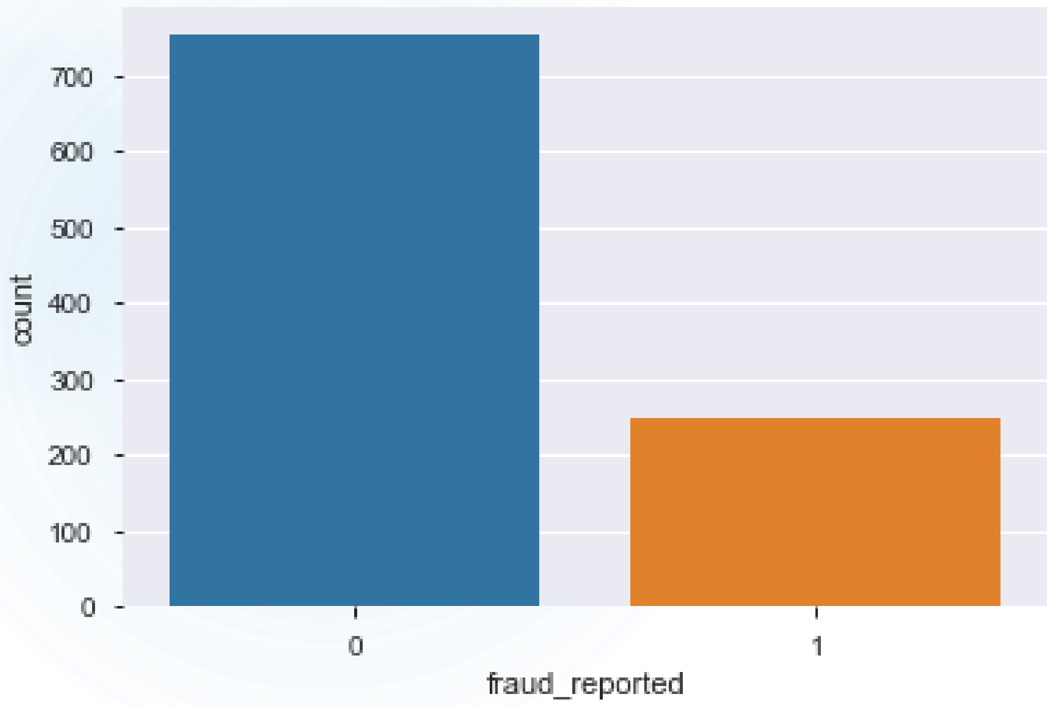
- **IBM Watson Cloud** with jupyter notebooks
- **Python** with with data science libraries (numpy, pandas, seaborn, scikit-learn and keras)



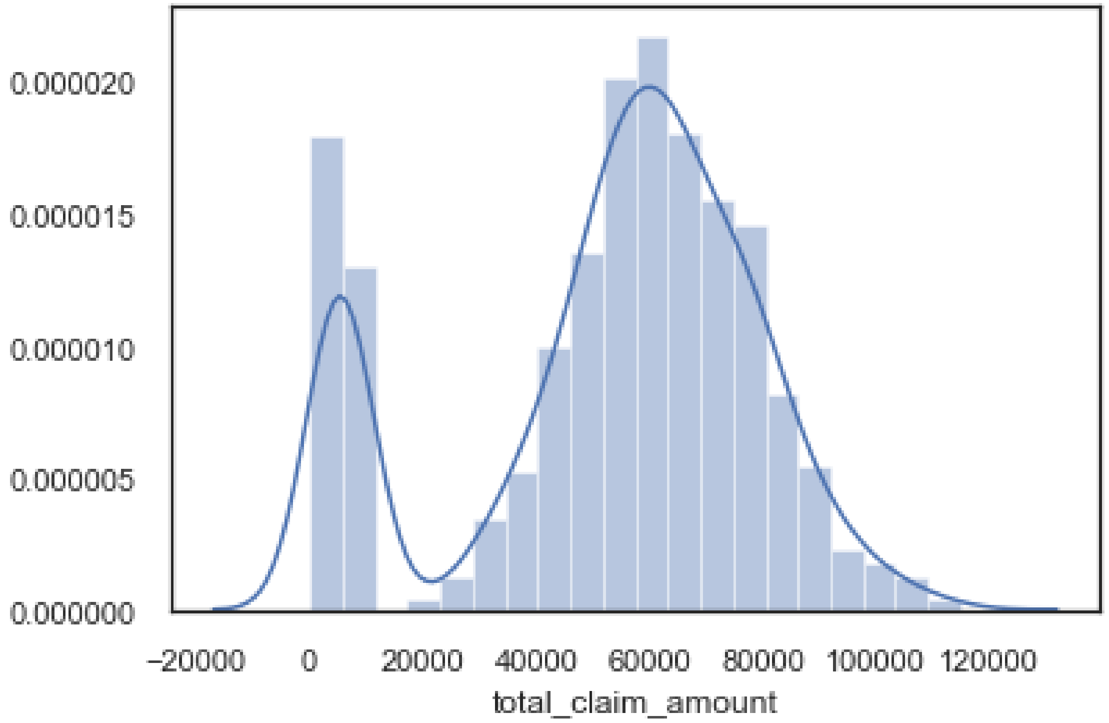
Data assessment – target variable

Historical data shows that 24.7% of claims are frauds.

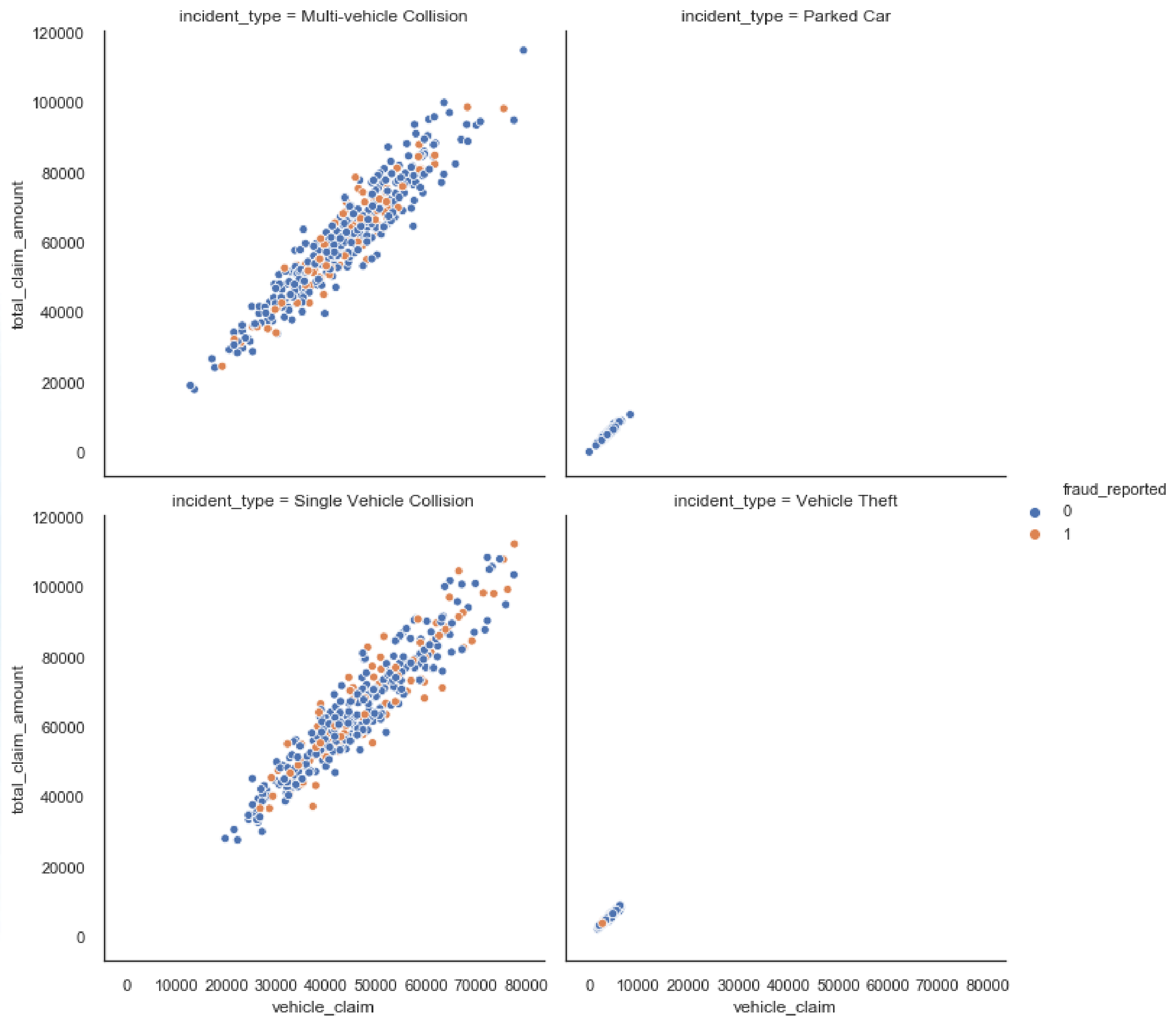
753 247



Total claim amount follows a normal distribution



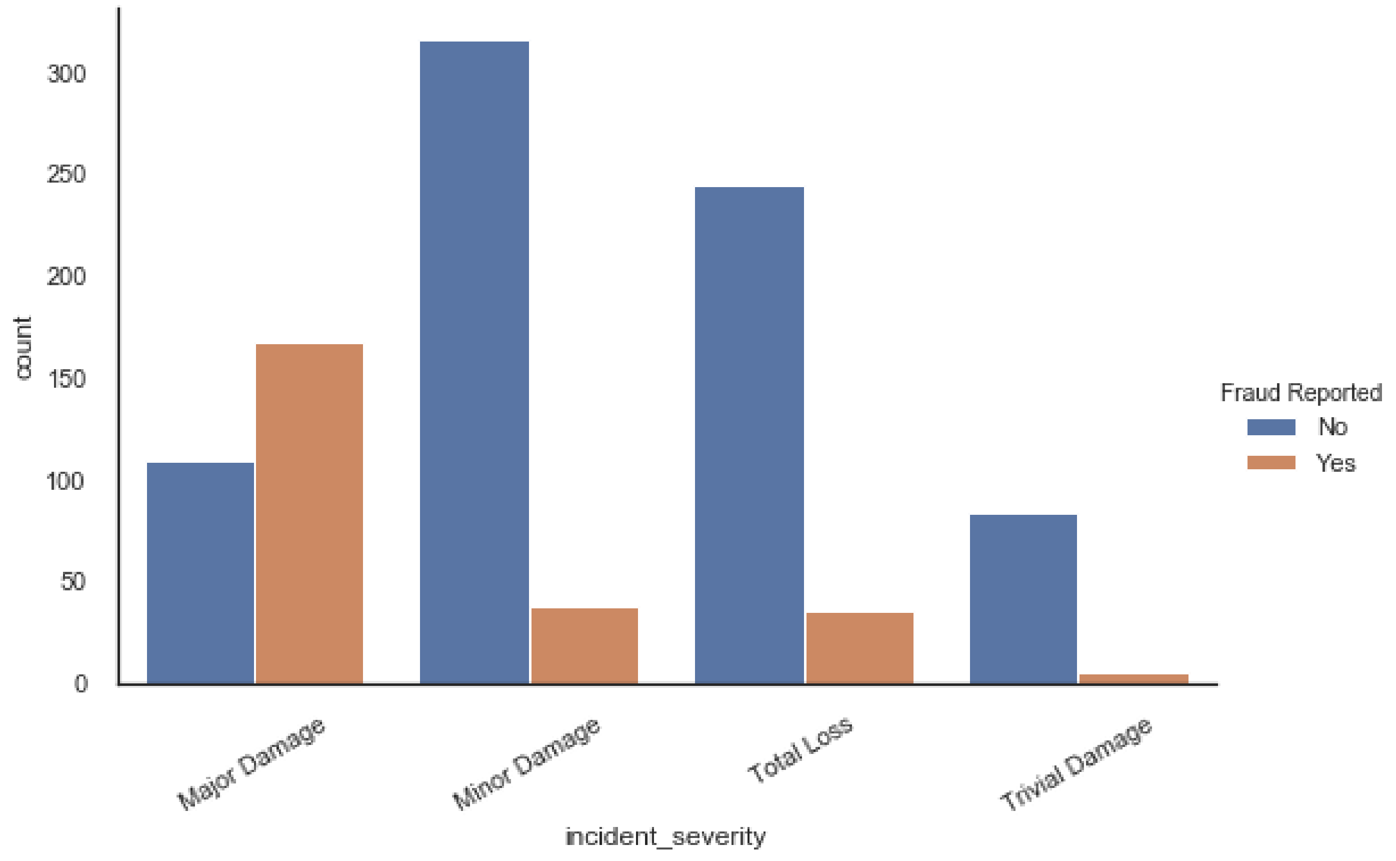
Data assessment – incident type



- Collisions fraud is more frequent and costs are higher
- Vehicle claim amount is major contributor to total claim amount

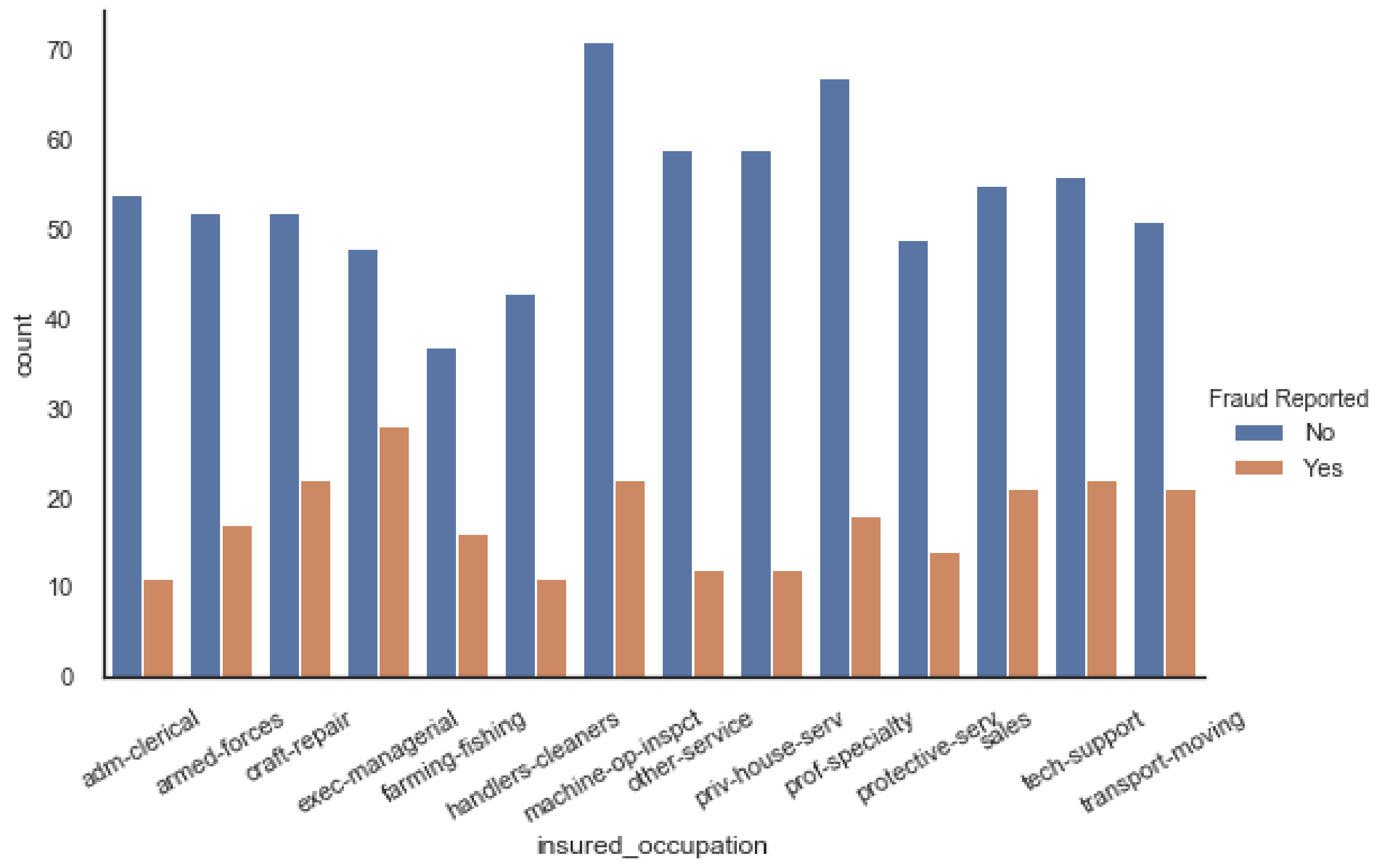
Data assessment – incident severity

Distribution of regular and fraud claims per incident severity



Data assessment – insured occupation

Distribution of regular and fraud claims per customer occupation



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. gender, occupation, hobbies, relationship, state, city, …
- **Regular Expressions**
  State abbreviation must have 2 letters only

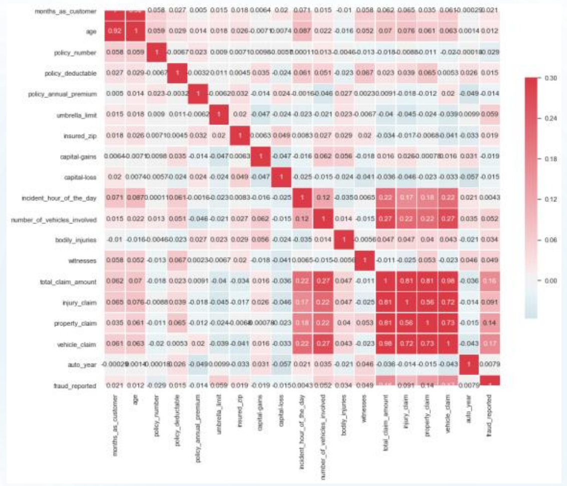
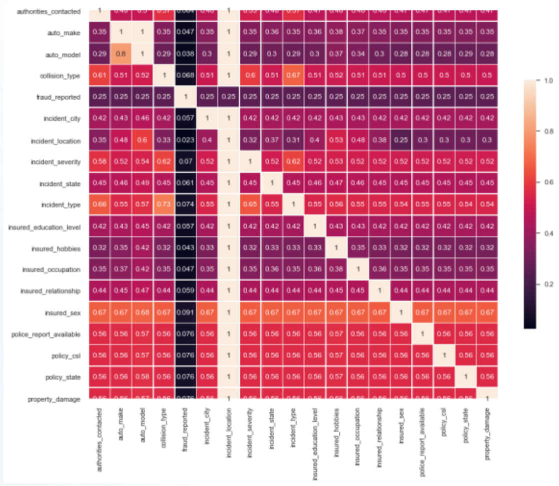
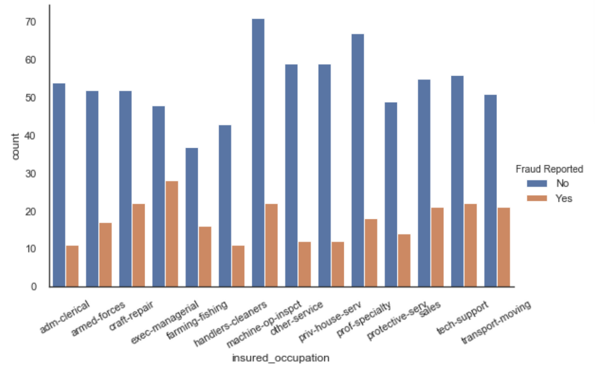
Feature correlations

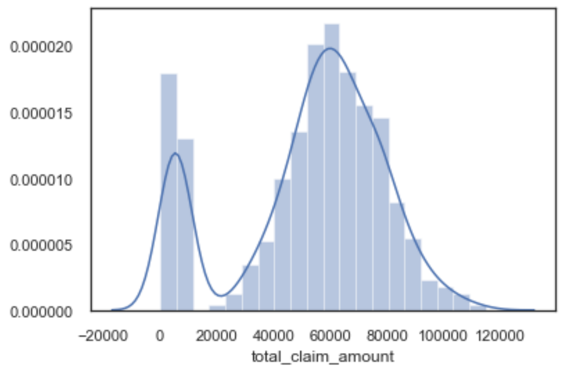
Between variable correlation analysis

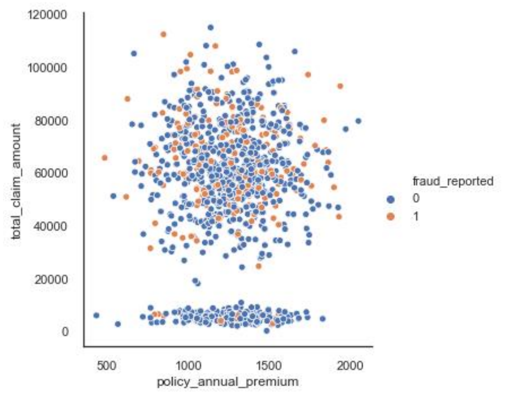
- Pearson, Cramer's V

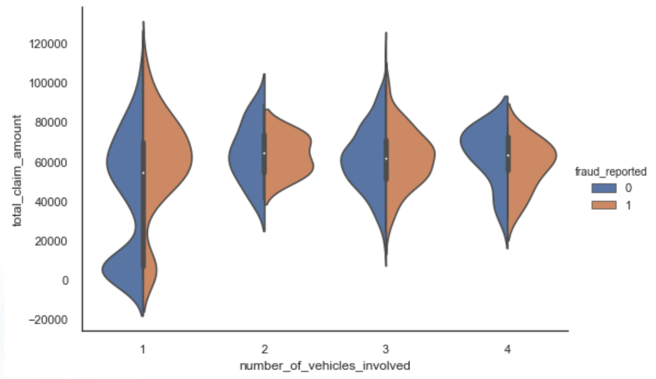
Within variable distribution to target correlation

- Histograms, scatterplots, boxplots, violinplots







Performance metric

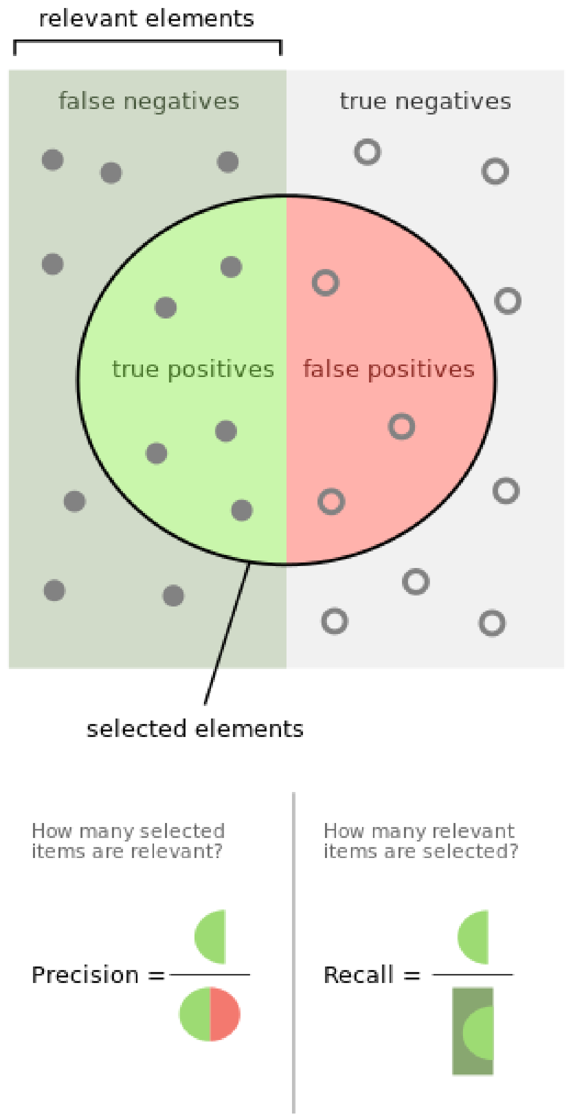
We are dealing with binary classification and an unbalanced dataset, so the chosen evaluation metric is **f1-score**.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

- robust measure which penalizes false positives and false negatives

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 claims as not fraud. No value from this model.
Logistic Regresion	0.03	Poor results. Not very useful.
Decision Tree	0.56	This class of models is well 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,

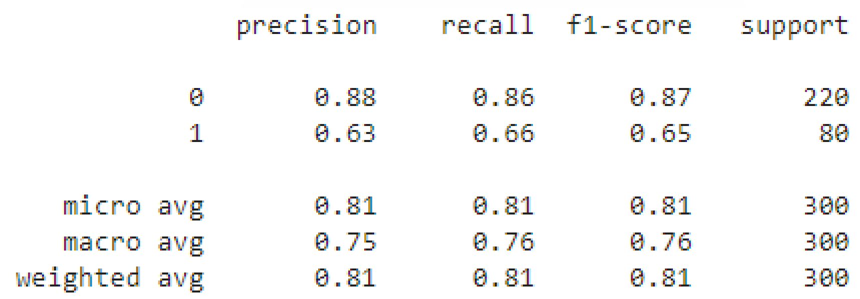
'bagging_fraction': 0.5,

'bagging_freq': 20,

'learning_rate': 0.05,

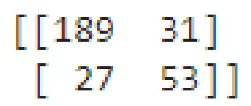
Model is trained with clean dataset with most influential features extracted.

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
```

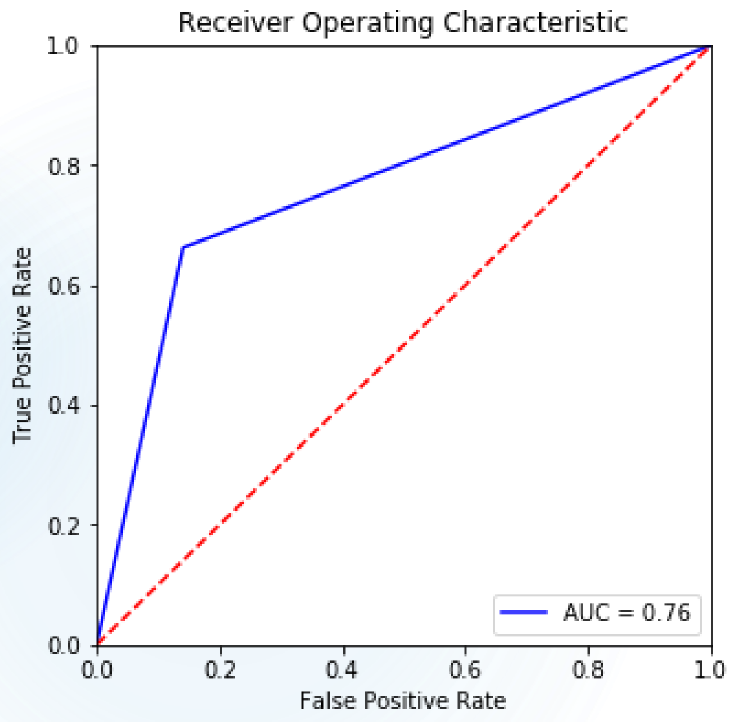
Confusion matrix:



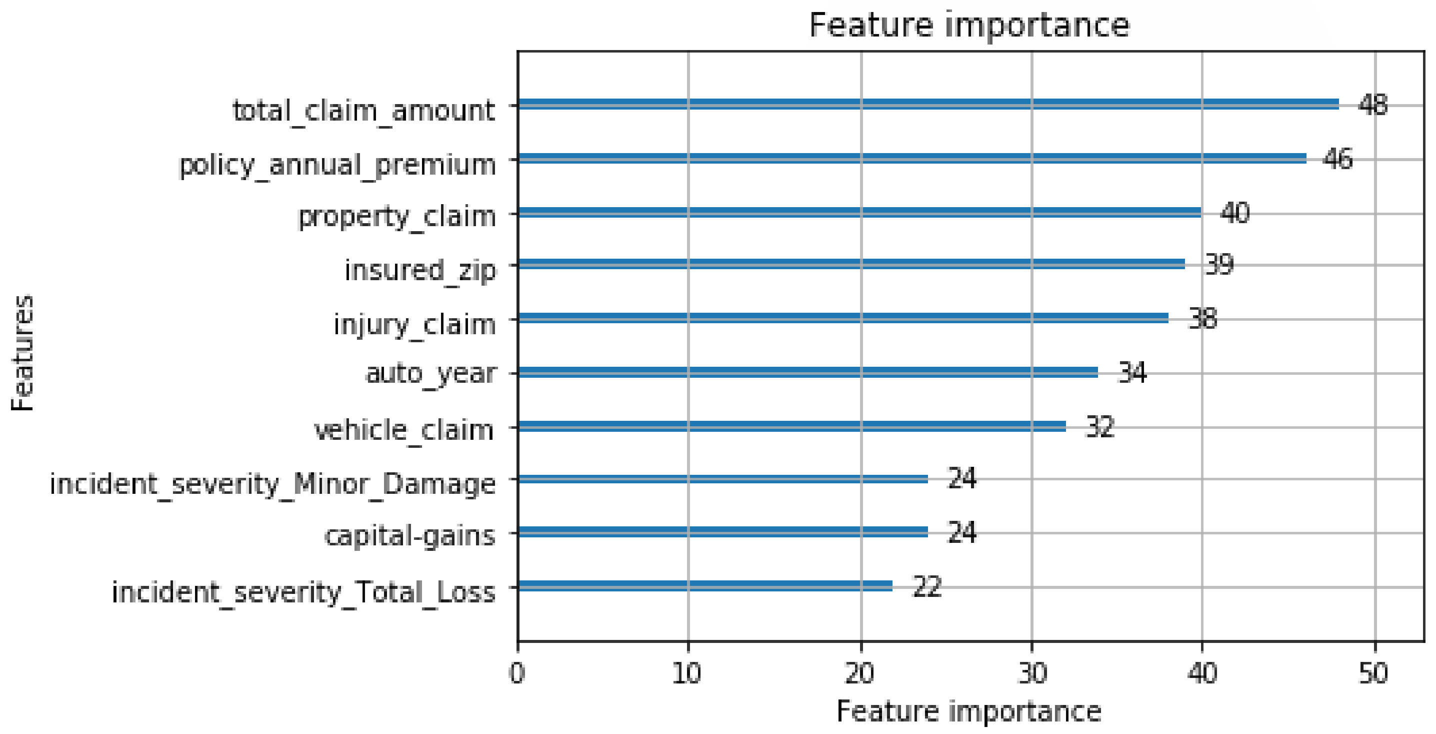
```
[[189  31]
 [ 27  53]]
```

LightGBM
Performance curve and important features

Area under the curve is **0.76**



Most important features



Neural Network
Summary

2nd best performing model is a **fully connected, single hidden layer NN** with **f1-score 0.63** for detecting 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

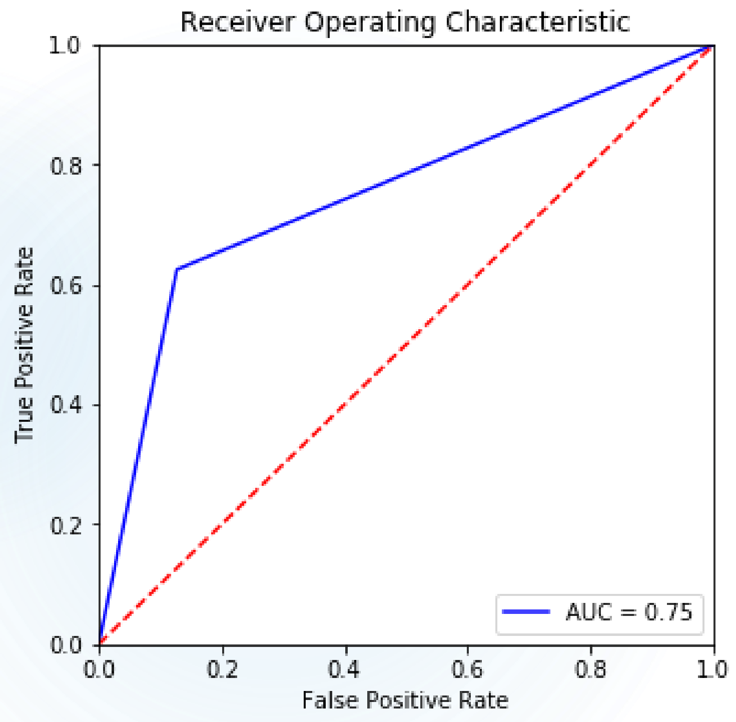
```
              precision    recall  f1-score   support

           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**



Using the model

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

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

Further resources

- Kaggle dataset - https://www.kaggle.com/buntyshah/auto-insurance-claims-data
- Talos framework - https://github.com/autonomio/talos
- Jupyter notebooks on IBM Cloud – see video description