MACHINE LEARNING CASE STUDY

Giancarlo Pozzo

Process overview

- **Study data**
  - Check data is meaningful
  - Clean
  - Add features
  - Remove features
  - Deal with outliers

- **Learning**
  - Standardize features (remove the mean and scale to unit variance)
  - Split data into train, cross validation and test subsets
  - Fit logistic regression
  - Measure model Performance

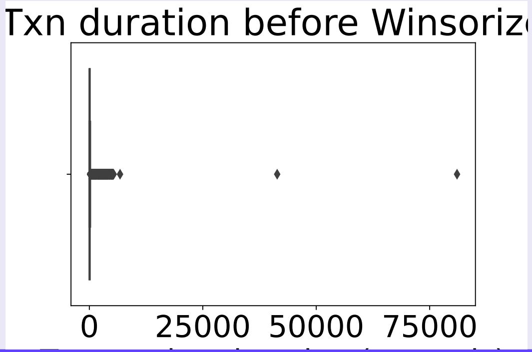
- **Improving the model**
  - Maximise F1 score over the regularisation parameter and threshold
  - Performance
  - Modify F1 score

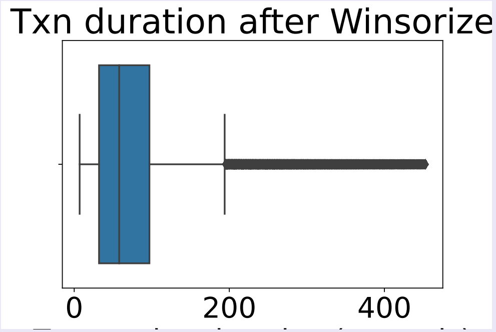
Check and clean

- **Dates makes sense:**
  - Initial trans. time < transaction completed time
  - First transaction date $\leq$ transaction completed time
  - First transaction date < today
- **Converted all "objects" to numbers, eg:**
  - Mozilla/5.0 (Linux; Android 11; SM-M215F) Apple… → 27
  - "IN" → 14, "SG" → 32, … both in GeoIpCountry and Alpha2Code (used same dictionary)
- **Replaced NaN with reasonable values:**
  - If there is no "FirstEmailDate" but there is "Email_Id", then substitute "FirstEmailDate" with "FirstTransactionDate"
  - If there are no reasonable values, substitute with 0

Check and clean

- **Added 10 features, eg:**
  - Duration of transactions
  - Number of transactions from IP
  - Interactions: (“Duration of transactions”) * (“UniquePaymentChannel”)
- **Outliers with Tukey's method**
  - About 5% of outliers have “Flag 1”, we keep them and use Winsorize method (set a border and put all outliers at the border)





First model

- **Define the input (X) and output (Y) variables**
- **Scale the input variables so that the mean $\mu = 0$ and variance $\sigma^2 = 1$**
- **Divide data in 60% train, 20% cross validation, 20% test**
- **Fit logistic regression with options**
  - class_weight="balanced": since we have skewed data we weight more the less represented class, that is we adjust weights inversely proportional to class frequencies in the input data
  - penalty='l2', the default one
- **Measure model performance on the test set using threshold = 0.5**
  - Given x_{test} *if* the model predicts probability bigger than threshold 0.5, the prediction is y_{pred}=1
  - Then we check the prediction y_{pred} against y_{test}
  - Calculate

$$F1_{\text{score}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = 0.14$$

	Y_test = 1	Y_test = 0
Y_pred = 1	True positive 374 = 0.92%	False positive 4538 =11.21%
Y_pred = 0	False negative 105 = 0.26%	True negative 35461=87.61%

Improved Model 1

- **Find regularisation C and thresholds T hyperparameters that maximise F1 score**
  - Define a function of C and T that returns F1 score
    - Fit logistic regression with C over the training set
    - For x in the cross validation set and given the threshold T, predict y
    - Given the predicted y and the cross validation set, calculate F1 score
  - Maximise the F1 score function with bounds 0 < C < inf and 0 < T < 1
- **Measure the model performance with the found optimised hyperparameters**

$$F1_{\text{score}}(C_{\text{opt}}, T_{\text{opt}}) = 2 \times \frac{\text{precision}_{\text{opt}} \times \text{recall}_{\text{opt}}}{\text{precision}_{\text{opt}} + \text{recall}_{\text{opt}}}$$

$$= 0.38$$

	Y_test = 1	Y_test = 0
Y_pred = 1	True positive 198 = 0.49%	False positive 371 =0.92%
Y_pred = 0	False negative 281 = 0.69%	True negative 39628=97.90%

Improved Model 2

- **Tripled F1 score BUT we worry about the decrease of true positive cases**
- **Modify F1 score function:**
  - more weight to recall = true_positive / actual_positive
  - less weight to precision = true_positive / actual_positive
- **Maximise modified function and measure performance**

$$F1_{\text{modified}}(C,T) = 2 \times \frac{\text{precision} \times \text{recall}^4}{\text{precision} + \text{recall}^4}$$

$$F1_{\text{score}}(C'_{\text{opt}}, T'_{\text{opt}}) = 0.24$$

	Y_test = 1	Y_test = 0
Y_pred = 1	True positive 330 = 0.81%	False positive 1999 =4.94%
Y_pred = 0	False negative 149 = 0.37%	True negative 38000=93.88%

Some possible next steps and algorithms

- **Improve features**
  - Convert all prices in one currency
  - Add frequency of transactions for each user
  - Add polynomial terms
- **Clustering**
  - discover new structures and features in data
- **Gaussian mixture models**
  - Multivariate anomaly detection
  - sklearn.mixture learn and estimate complex outliers
- **Suggests for data collection**
  - In email, hash separately username and domain

Thank You