Detecting Fraud with Data Science

...

David Hogue, Jeff Grana, Neil Bezdek, Ryan Busby

galvanize

Overview

Objective: Examine Eventbrite data for fraudulent events

Dataset: 43 features + 1 fraud label

Workflow: Feature engineering - 37 features

Model building - Random Forest, Gradient Boost, ADABoost

Profit Curve Analysis

Deliverables: Dashboard to predict fraud from live stream

Examine fraud cases to guide feature engineering

"BEST EVENT EVER -- BELIEVE ME"

Events with high percentage of capitalized text more likely to be fraud

$$$

Fraudulent events more likely to promote high-end “luxury” products

Feature Engineering

Dropped:

'object_id'
'sale_duration'
'user_created'
'user_type'
'ticket_type'
'venue_address'
'venue_latitude'
'venue_longitude'
'venue_country'
'venue_name'
'venue_state'
'payout_type'
'email_domain'
'description'
'approx_payout_date'
'event_start'
'event_end'
'event_published'
'event_created'
'currency'
'country'
'delivery_method'
'acct_type'
'name'

Kept:

'body_length'
'gts'
'has_logo'
'num_order'
'show_map'
'channels'
'has_analytics'
'num_payouts'
'user_age'
'name_length'
'sale_duration_2'
'fb_published'

Edited:

'has_header'
'org_desc'
'listed'
'org_facebook'
'org_twitter'
'payee_name'
'org_name'
'event_published'
'blank_venue_name'
'time_to_pay'
'time_to_pay2'
'planning_time'
'has_pub_date'

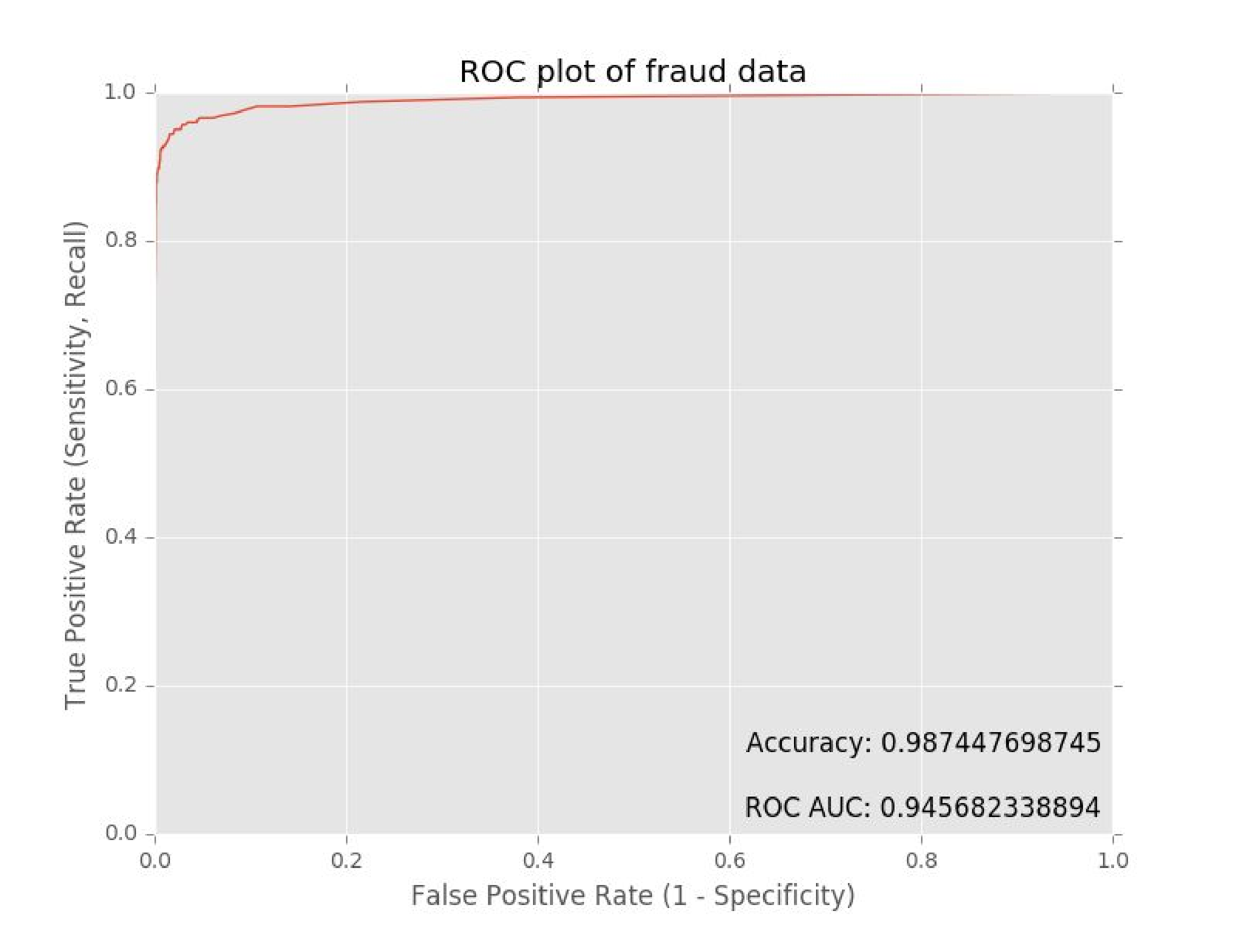
Engineered:

'avg_ticket_price'
'max_ticket_price'
'ticket_tiers'
'total_available'
'pct_caps'
'previous_payouts'
'blank_venue_address'
'same_countries'
'org_facebook_nan'
'nan_venue'
'org_twitter_nan'
'duration'

Feature Importance

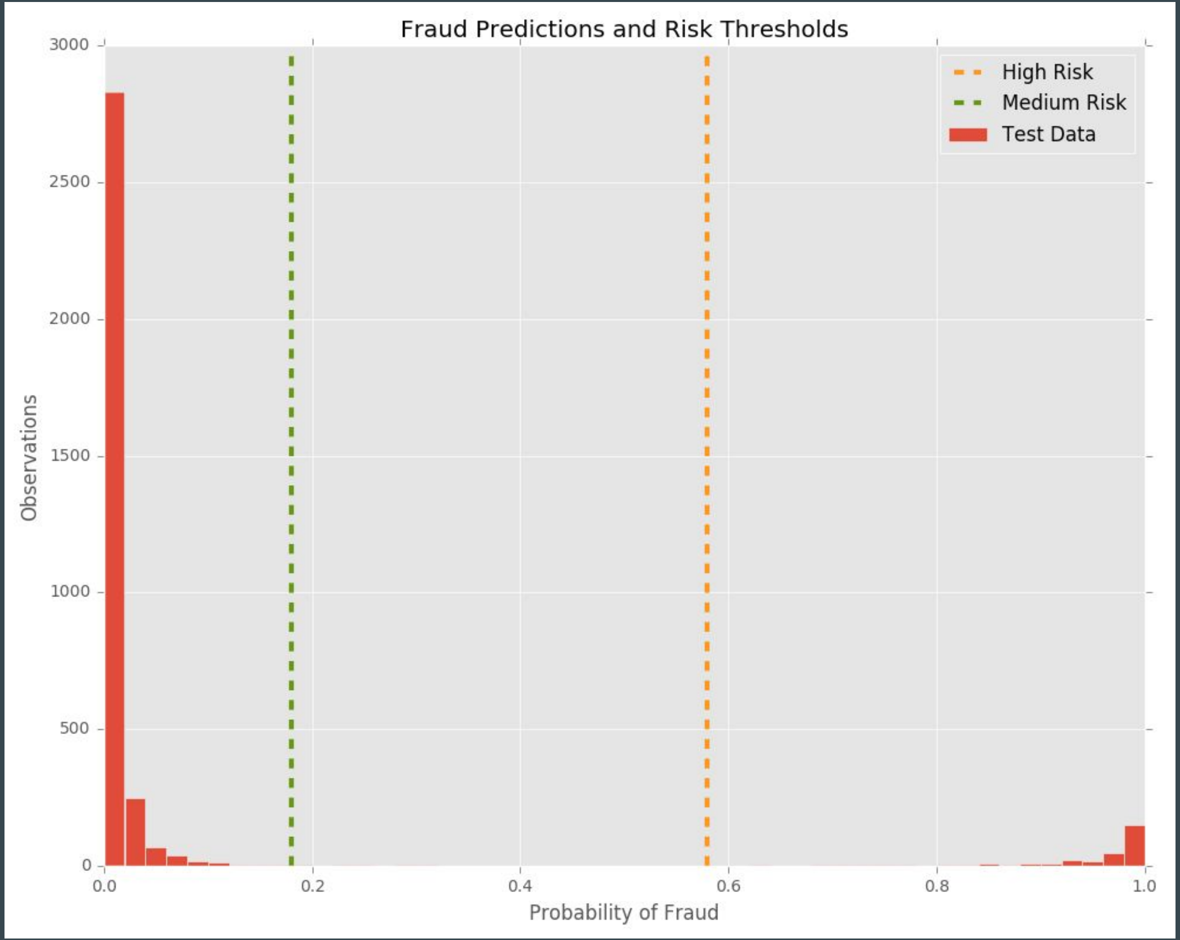
(u'previous_payouts',	**0.3371785400880879),**	**Engineered: df['previous_payouts'].apply(len)**
('time_to_pay2',	**0.11508351431934936),**	**Engineered: df['approx_payout_date'] - df['event_start']**
(u'sale_duration2',	0.08619493544756405 6),	Kept
(u'gts',	0.067157648378076376),	Kept
(u'user_age',	0.057314953453766054),	Kept
(u'num_order',	0.053524360465484186),	Kept
('pct_caps',	**0.029795932631783963),**	**Engineered: percent capitals from df[‘name’]**
(u'num_payouts',	0.027944121385088033),	Kept
(u'body_length',	0.025926269426975432),	Kept
(u'name_length',	0.022668285447869457),	Kept
('avg_ticket_price',	**0.02188740770758547),**	**Engineered: Weighted Average (# tickets * cost @ each tier)**
('max_ticket_price',	**0.021178139089864909),**	**Engineered: Max price of all ticket types**
(u'org_facebook',	0.01733205830640748),	Edited: df['org_facebook'].fillna(0.)
(u'org_twitter',	0.012126223484773999),	Edited: df['org_twitter'].fillna(0.)
('duration',	**0.011166827233625016),**	**Engineered: df['event_end'] - df['event_start']**
('time_to_pay',	**0.0109335820590898),**	**Engineered: df['approx_payout_date'] - df['event_start']**
('planning_time',	**0.010801764808993627)**	**Engineered: f['approx_payout_date'] - df['event_start']**

ROC Curve



ROC plot of fraud data
True Positive Rate (Sensitivity, Recall)
False Positive Rate (1 - Specificity)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Accuracy: 0.987447698745
ROC AUC: 0.945682338894

Model Probabilities



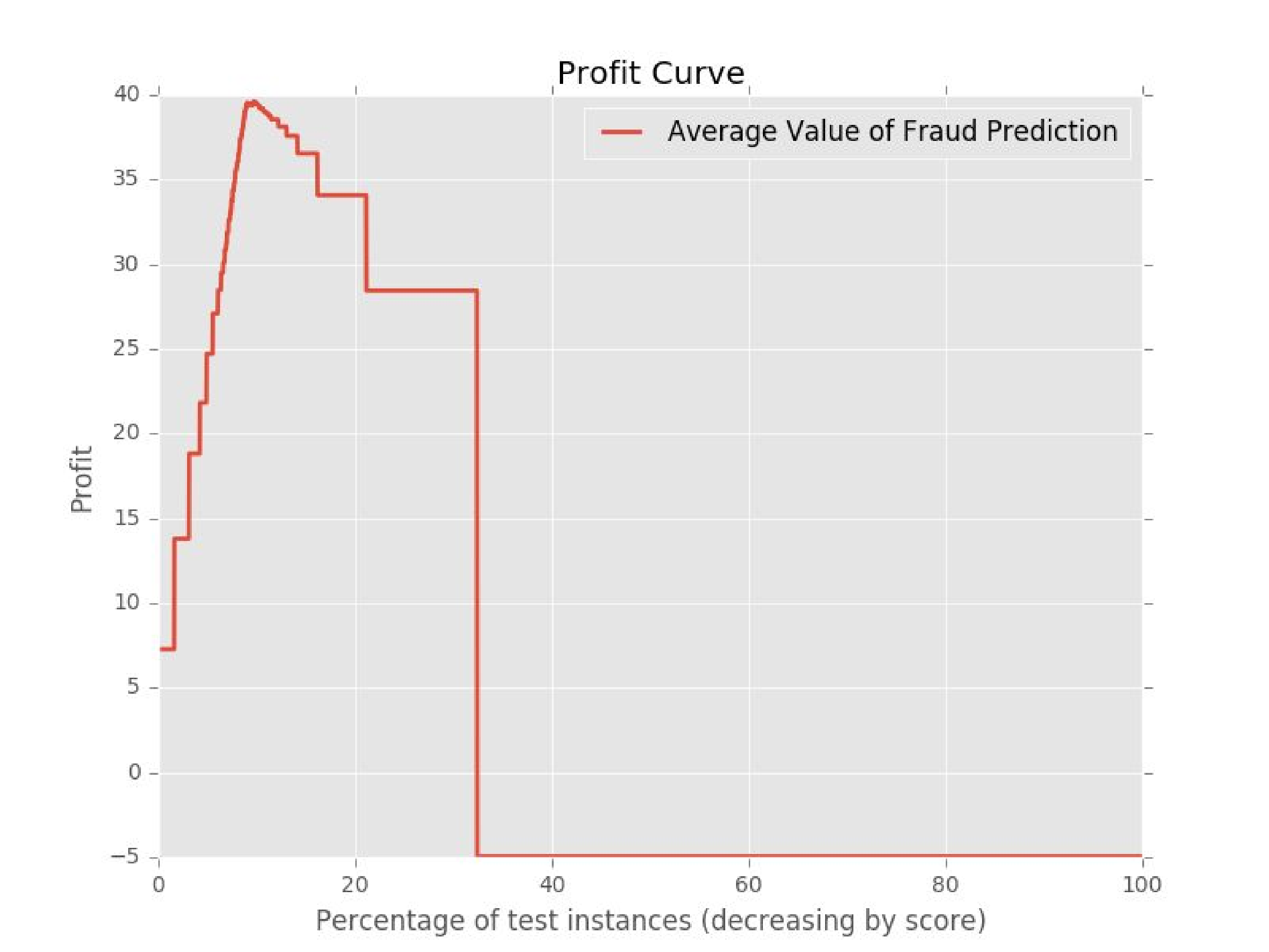
Profit Curve

TP 450	FP -50
0 FN	0 TN

Threshold = 0.18
Profit = 39.59

TP 450	FP -1000
0 FN	0 TN

Threshold = 0.58
Profit = 38.05



Conclusions

EDA and feature engineering contributed to model accuracy. Use common sense before feeding numbers to your model!

Random Forest model was most accurate model in this case.

Application of model was informed by analysis of cost function. In this case, classification changed from .5 to .1 as a result.

We are better at data science than web development or network engineering.