Automation, Robots, AI, Credit Analytics

Sanjiv R. Das

ISB | FinTech | 2018

What is FinTech?

- FinTech refers to various financial technologies used to automate processes in the financial sector, from routine, manual tasks to non-routine, cognitive decision-making.

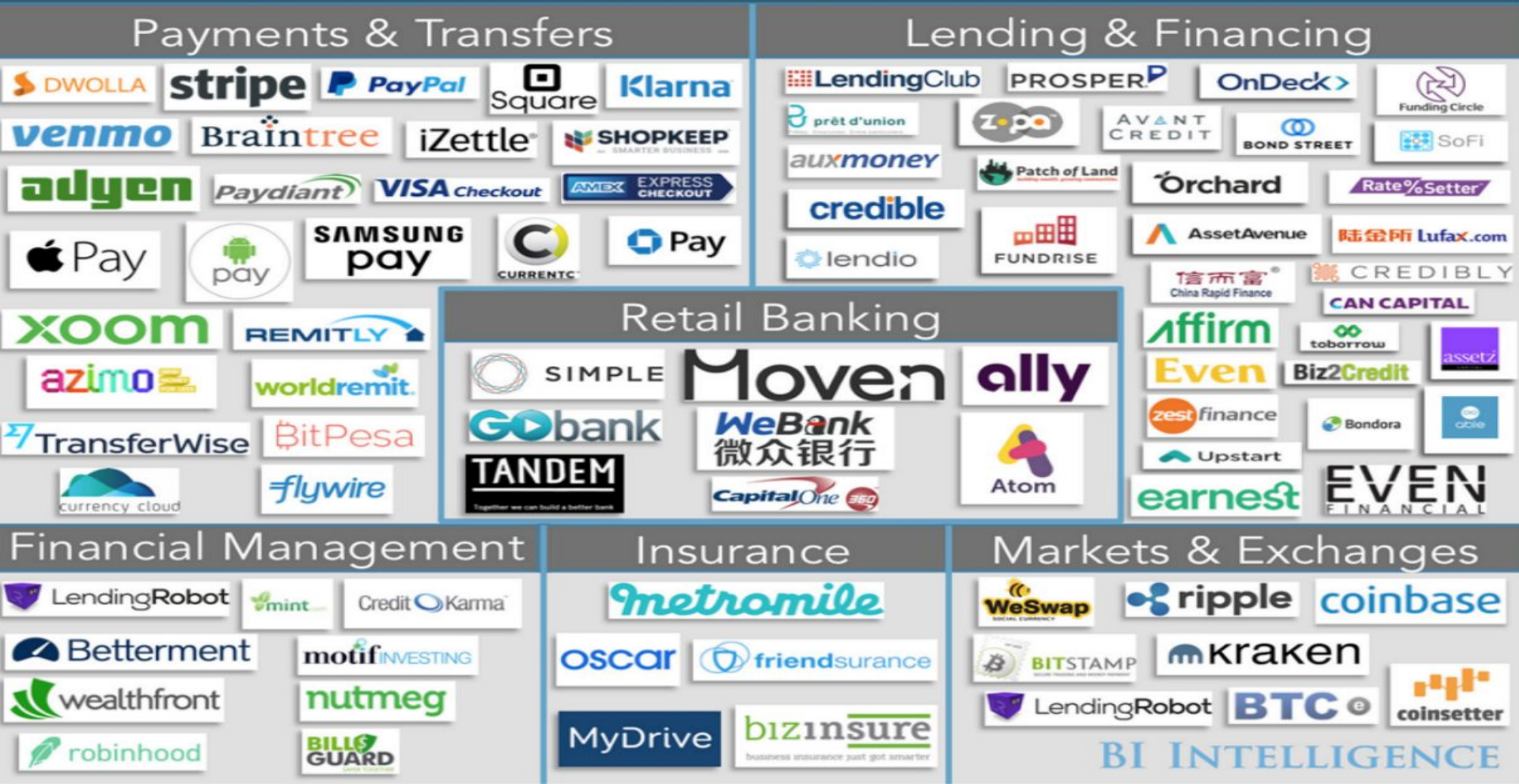
- FinTech may be characterized by technological change in three broad areas of finance: (This framework was proposed by my colleague, Professor Seoyoung Kim.)
  1. raising capital,
  2. allocating and investing capital,
  3. transferring capital.

- My definition: "FinTech is any technology that eliminates or reduces the cost of the middleman in finance."

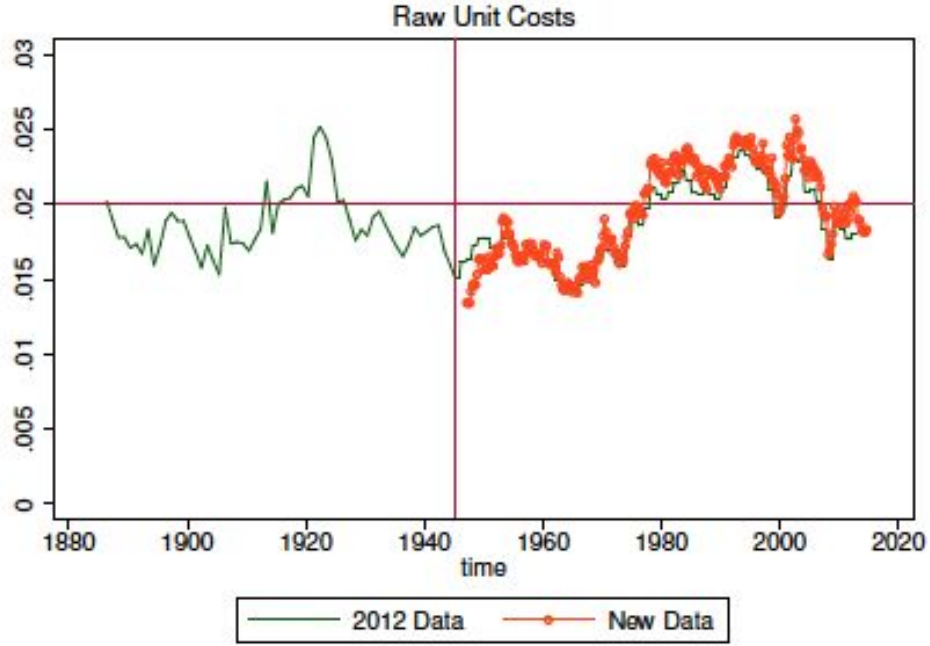
There is now a growing interest and literature: - http://lfe.mit.edu/research/fintech/

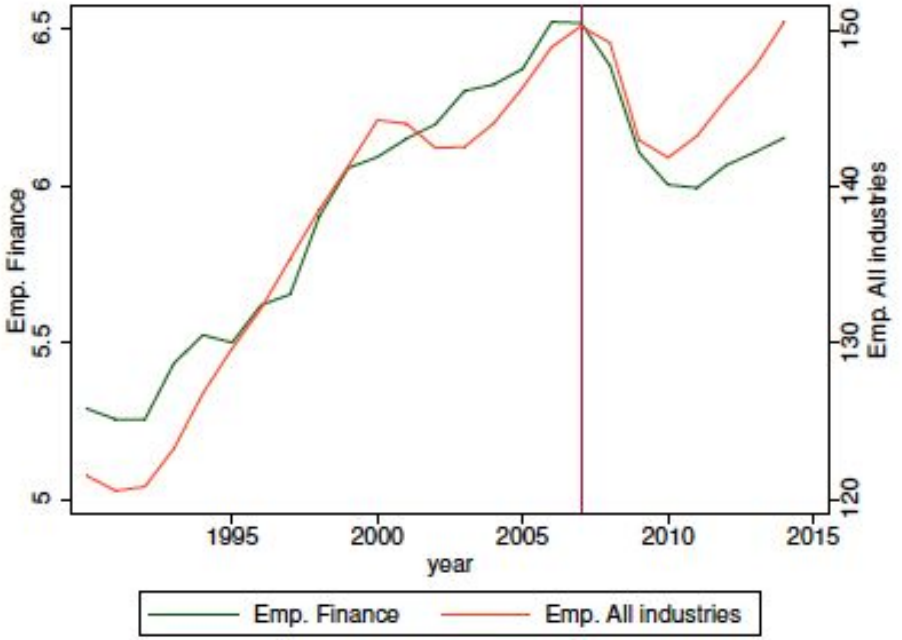
- Risk and Risk Management in the Credit Card Industry (Florentin Butaru, Qingqing Chen, Brian Clark, Sanmay Das, Andrew Lo, Akhtar Siddique), *Journal of Banking and Finance* 72(2016), 218–239.

THE FINTECH ECOSYSTEM



Costs of Financial Intermediation (Philippon 2016)





FinTech Framework

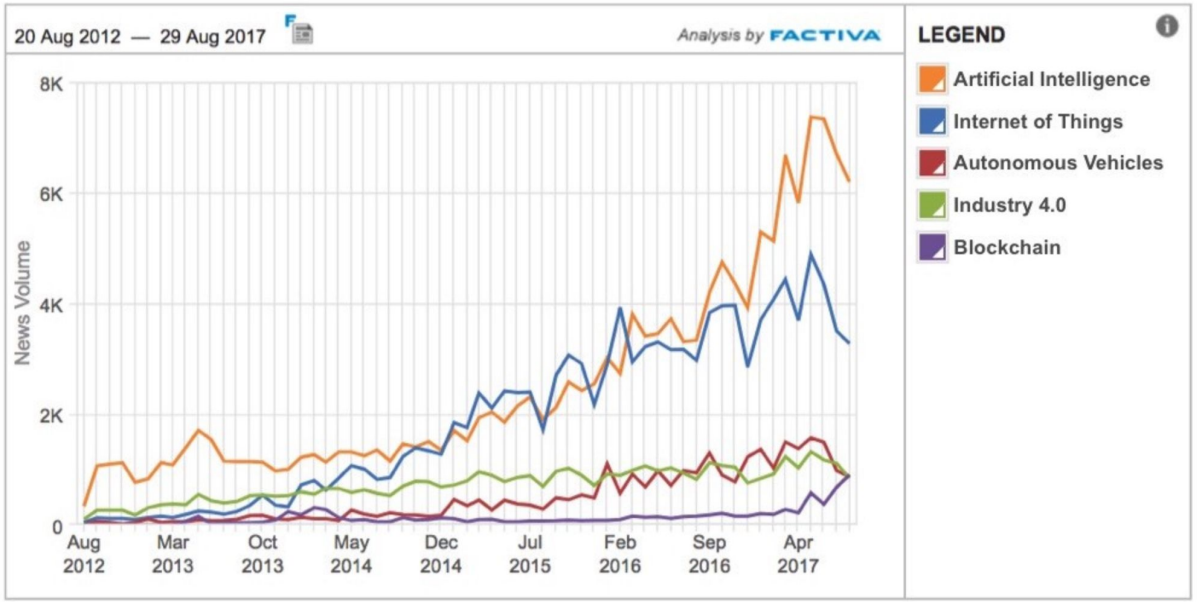
1. Machine Learning, AI, and Deep Learning.
2. Network Models.
3. Personal and Consumer Finance.
4. Nowcasting.
5. Cybersecurity.
6. Fraud Detection.
7. Payment and Funding Systems.
8. Automated and High-Frequency Trading.
9. Blockchain and Cryptocurrencies.
10. Text Analytics.

Examples

- Monitoring corporate buzz.
- Analyzing data to detect, analyze, and understand the more profitable customers or products.
- Targeting new clients.
- Customer retention.
- Lending activity (automated)
- Market prediction and trading.
- Risk management.
- Automated financial analysts.
- Financial forensics to prevent rogue employees.
- Credit cards: optimizing use, marketing offers.
- Fraud detection.
- Detecting market manipulation.
- Social network analysis of clients.
- Measuring institutional risk from systemic risk.

AI and the Technological Singularity

https://hackernoon.com/why-ai-is-now-on-the-menu-at-dinner-even-with-my-non-tech-friends-44c666348de4



Source: Factiva; Methodology: Searches for key terms in Factiva classified "Top News Sources" in English Language. 'Artificial Intelligence' search also includes OR'd terms 'AI' and 'Machine Learning', 'Internet of Things' includes 'IoT', 'Autonomous Vehicles' includes 'Driverless Car' and 'Self Driving Car', 'Industry 4.0' includes '3D Printing' and 'Advanced Manufacturing', 'Blockchain' includes 'Bitcoin'.

Transformation vs Change

Kurzweil claims that AI will pass the Turing test in 2029, and the singularity will come in 2045.

https://futurism.com/kurzweil-claims-that-the-singularity-will-happen-by-2045/

Definition of AI

- Intelligence exhibited by machines
- Narrow or Weak AI: “Expert systems that match or exceed human intelligence in a narrowly defined area, but not in broader areas” (Dvorsky G., 2013) e.g. Siri.
- Artificial General Intelligence: An artificial neural network not preprogrammed with fixed rules. Rewire itself to reflect patterns in the data, adaptable to its environment, in which (hopefully) advanced skills emerge organically.
- “Humans don’t learn to understand language by memorizing dictionaries and grammar books, so why should we possibly expect our computers to do so?” (LEWIS-KRAUS G, 2016).
- And, Super AI?

http://io9.gizmodo.com/how-much-longer-before-our-first-ai-catastrophe-464043243 (Dvorsky G., 2013)https://mobile.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html?_r=0&referer= (LEWIS-KRAUS G, 2016)

Two Types of AI

- Rule-Based AI
- Data-driven AI
- Example: Checkers. Albert Samuel @IBM began writing code for a checkers game program in 1949. In 1956, the program was demonstrated to the public on live television. In 1962, the computer beat checkers master player Robert Nealey, and IBM's stocks rose 15 percent overnight.

Rule-based AI can never be more intelligent than its creators, but data-driven AI can!

Machine Learning

- Machine learning is the subfield of computer science that gives computers the ability to learn without being explicitly programmed." (Arthur Samuel)

- Definition (Tom Mitchell): "A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P* if its performance at tasks in *T*, as measured by *P*, improves with experience *E*."

- Supervised vs Unsupervised.

THE AI IN FINTECH MARKET MAP

CREDIT SCORING / DIRECT LENDING

Affirm, AVANT, zestfinance, argon, ADF, habito, CREAM FINANCE, aire, float, naborly, Upstart, creditvidya, james, WecashUP

ASSISTANTS / PERSONAL FINANCE

digit, MoneyLion, personetics, Kasisto, clinc, Penny, TRIM, claritymoney, cleo., homebot, Active.Ai, change

QUANTITATIVE & ASSET MANAGEMENT

wealthfront, sentient technologies, SIGOPT, ALPINE DATA, Clone Algo, NUMERAI, fount, domeyard LP, Alpaca, AIM, ForwardLane, TRUMID, ALGORIZ, it.ai, AIDYIA, FINATIX

REGULATORY, COMPLIANCE, & FRAUD DETECTION

TRIFACTA, Digital Reasoning, WorkFusion, Paxata, DataRobot, siftscience, onfido, feedzai, SKYTREE, SOCURE, bigstream, trooly, BehavioSec, ComplyAdvantage, cortical.io, Dimebox, Fraugster, neurensic, PredicSis, Jewel Paymentech, CheckRecipient, Fortia

INSURANCE

CAPE ANALYTICS, captricity, TRACTABLE, riskgenius, CYENCE, Shift Technology, Zendrive, Lemonade, UNDERSTORY

MARKET RESEARCH / SENTIMENT ANALYIS

Dataminr, alphasense, Orbital Insight, Descartes Labs, iSENTIUM, indico, acuity insights, SIGNAL, FeedStock

GENERAL PURPOSE / PREDICTIVE ANALYTICS

OPERA, AYASDI, KENSHO, NarrativeScience, context relevant, H2O.ai, smartzip, CognitiveScale, Anodot, Lucena Quantitative Analytics, Numenta

BUSINESS FINANCE & EXPENSE REPORTING

fyle, NetChain², AppZen, ZEITGOLD., MINT

DEBT COLLECTION

TrueAccord, collectAI

CBINSIGHTS

Investing in AI

QUOTES & COMPANIES

VIEW ALL COMPANIES

AI Powered Equity ETF

AIEQ (U.S.: NYSE Arca)

REAL TIME 3:29 PM EST 11/06/17

$24.7101 USD

0.1101 0.45% ▲

Volume
164,231

65 Day Avg Vol
420,342

1 Day Range
24.55 - 24.707

52 Week Range
24.50 - 25.99
(11/03/17 - 10/23/17)

ADVANCED CHARTING

COMPARE

Open 24.55 Prior Close 24.60 (11/03/17)

1 Day AIEQ 0.45% ▲ DJIA 0.13% ▲ S&P 500 0.20% ▲ Nasdaq 0.37% ▲

OVERVIEW

ALL SECTIONS

Investment Information AIEQ

POWERED BY LIPPER

Category	Multi-Cap Growth	Net Expense Ratio *	
Style	Growth	Turnover %	.00

* Expense ratio updated annually from fund's year-end report.

Investment Policy

The Fund seeks capital appreciation. The Fund invests primarily in US exchange listed equity securities based on the results of a quantitative model which identifies approximately 30 to 70 companies with the greatest potential over the next 12 months for appreciation and their corresponding weights.

FUND DETAILS

Net Assets
70.90 M

NAV
$24.65 (11/03/17)

Shares Outstanding
N/A

Yield
N/A

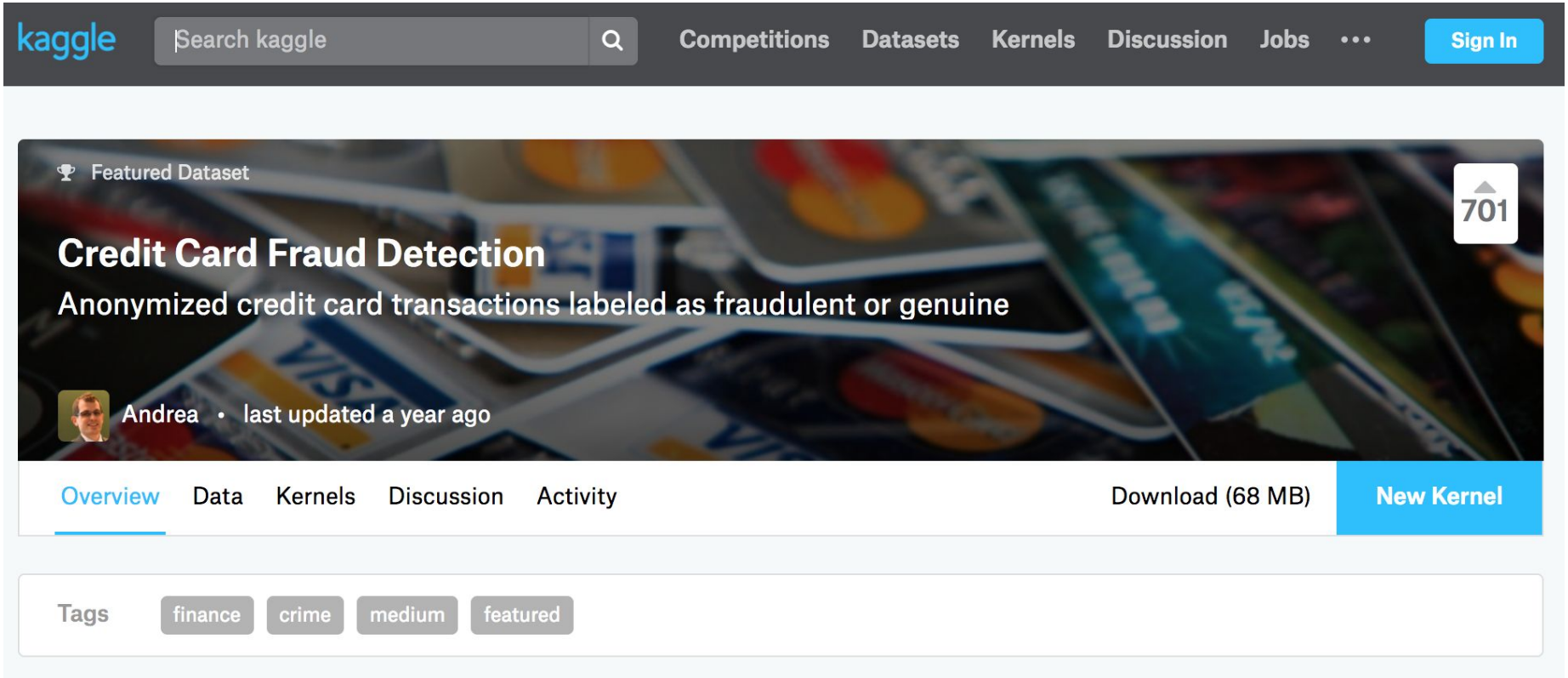
Latest Dividend

Detecting Financial Fraud

- Financial fraud allows perpetrator to be removed from the scene of the crime.
- Therefore, logging all financial activity to enable traceback is critical.
- Limited defense at the authentication stage if there is a data breach.
- Widespread use of machine learning.
- Social media based; highly consumer-centric. Device usage, Email use, customer location at time of transaction.
- Adaptive behavioral analytics, e.g., Bionym, EyeVerify, BioCatch.
- Anomaly detection is a hard problem, with unbalanced data.

Ayasdi: https://www.ayasdi.com/ uses Topological Data Analysis
Simility: https://simility.com/ uses device tracking techniques.

Credit Card Fraud Detection



Butaru et al (2016): http://www.sciencedirect.com/science/article/pii/S0378426616301340

Credit Card Fraud Detection Analysis

https://www.kaggle.com/dalpozz/creditcardfraud (See Jupyter NB = CC_Fraud_RF)

The datasets contains transactions made by credit cards in September 2013 by european cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-senstive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

Given the class imbalance ratio, we recommend measuring the accuracy using the Area Under the Precision-Recall Curve (AUPRC). Confusion matrix accuracy is not meaningful for unbalanced classification.

The dataset has been collected and analysed during a research collaboration of Worldline and the Machine Learning Group (http://mlg.ulb.ac.be) of ULB (Université Libre de Bruxelles) on big data mining and fraud detection. More details on current and past projects on related topics are available on http://mlg.ulb.ac.be/BruFence and http://mlg.ulb.ac.be/ARTML

Please cite: Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson and Gianluca Bontempi. Calibrating Probability with Undersampling for Unbalanced Classification. In Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2015

Kaggle's Credit Card Fraud Dataset - RF

In this notebook I'll apply a Random Forest classifier to the problem, but first I'll address the severe class imbalance of the set using the SMOTE ENN over/under-sampling technique.

Data is from: https://www.kaggle.com/dalpozz/creditcardfraud

```
%pylab inline
import pandas as pd
from sklearn.model_selection import train_test_split
from imblearn.combine import SMOTEENN
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import roc_curve,auc
from sklearn.metrics import confusion_matrix
```

```
Populating the interactive namespace from numpy and matplotlib
```

Credit Card Data

```
%%time
data = pd.read_csv('creditcard.csv')
print(data.shape)
print(data.head())
```

```
(284807, 31)
   Time        V1        V2        V3        V4        V5        V6        V7
0   0.0 -1.359807 -0.072781  2.536347  1.378155 -0.338321  0.462388  0.239599
1   0.0  1.191857  0.266151  0.166480  0.448154  0.060018 -0.082361 -0.078803
2   1.0 -1.358354 -1.340163  1.773209  0.379780 -0.503198  1.800499  0.791461
3   1.0 -0.966272 -0.185226  1.792993 -0.863291 -0.010309  1.247203  0.237609
4   2.0 -1.158233  0.877737  1.548718  0.403034 -0.407193  0.095921  0.592941

         V8        V9  ...         V21       V22       V23       V24  \
0  0.098698  0.363787  ...   -0.018307  0.277838 -0.110474  0.066928
1  0.085102 -0.255425  ...   -0.225775 -0.638672  0.101288 -0.339846
2  0.247676 -1.514654  ...    0.247998  0.771679  0.909412 -0.689281
3  0.377436 -1.387024  ...   -0.108300  0.005274 -0.190321 -1.175575
4 -0.270533  0.817739  ...   -0.009431  0.798278 -0.137458  0.141267

        V25       V26       V27       V28  Amount  Class
0  0.128539 -0.189115  0.133558 -0.021053  149.62      0
1  0.167170  0.125895 -0.008983  0.014724    2.69      0
2 -0.327642 -0.139097 -0.055353 -0.059752  378.66      0
3  0.647376 -0.221929  0.062723  0.061458  123.50      0
4 -0.206010  0.502292  0.219422  0.215153   69.99      0

[5 rows x 31 columns]
CPU times: user 2.46 s, sys: 112 ms, total: 2.57 s
Wall time: 2.59 s
```

Quick Class counts

```
data[["Class","V1"]].groupby(["Class"]).count()
```

	V1
Class	
0	284315
1	492

Class Summary

Mean Amount in Each class

```
data[["Class","Amount"]].groupby(["Class"]).mean()
```

Class	Amount
0	88.291022
1	122.211321

```
X_train, X_test, y_train, y_test = train_test_split(data.drop('Class',axis=1), data['Class'], test_size=0.33)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(190820, 30)
(190820,)
(93987, 30)
(93987,)
```

Sample Calibration (Culkin & Das 2017)

```
from sklearn.model_selection import train_test_split
from imblearn.combine import SMOTEENN
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import roc_curve,auc
from sklearn.metrics import confusion_matrix
```

```
%%time
data = pd.read_csv('/Users/srdas/GoogleDrive/Papers/De
print(data.shape)
```

```
(284807, 31)
CPU times: user 2.34 s, sys: 143 ms, total: 2.48 s
Wall time: 3.43 s
```

```
data[["Class","V1"]].groupby(["Class"]).count()
```

Class	V1
0	284315
1	492

```
data[["Class","Amount"]].groupby(["Class"]).mean()
```

Class	Amount
0	88.291022
1	122.211321

Feature Engineer the data

Slide Type

```
X_train, X_test, y_train, y_test = train_test_split(data.drop('Class',axis=1), data['Class'], test_size=0.33)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(190820, 30)
(190820,)
(93987, 30)
(93987,)
```

Slide Type

```
sme = SMOTEENN()
X_train, y_train = sme.fit_sample(X_train, y_train)
print(X_train.shape)
print(y_train.shape)
unique(y_train, return_counts=True)
```

```
(357479, 30)
(357479,)

(array([0, 1]), array([174864, 182615]))
```

What about "downsampling"?

SMOTE: Synthetic Minority Over-sampling Technique

Nitesh V. Chawla[1], Kevin W. Bowyer[2],
Lawrence O. Hall[1], W. Philip Kegelmeyer[3]

[1]Department of Computer Science and Engineering, ENB 118
University of South Florida
4202 E. Fowler Ave.
Tampa, FL 33620-5399, USA

[2]Department of Computer Science and Engineering
384 Fitzpatrick Hall
University of Notre Dame
Notre Dame, IN 46556, USA

[3]Sandia National Laboratories
Biosystems Research Department, P.O. Box 969, MS 9951
Livermore, CA, 94551-0969, USA

chawla@csee.usf.edu, kwb@cse.nd.edu, hall@csee.usf.edu, wpk@ca.sandia.gov

Under/over-sample with SMOTE ENN to overcome class imbalance

While a Random Forest classifier is generally considered imbalance-agnostic, in this case the severity of the imbalance resuts in overfitting to the majority class.

The Synthetic Minority Over-sampling Technique (SMOTE) is one of the most well-known methods to cope with it and to balance the different number of examples of each class.

The basic idea is to oversample the minority class, while trying to get the most variegated samples from the majority class.

Different types of Re-sampling methods: (see http://sci2s.ugr.es/noisebor-imbalanced)

1. SMOTE in its basic version. The implementation of SMOTE used in this paper considers 5 nearest neighbors, the HVDM metric to compute the distance between examples and balances both classes to 50%.
2. SMOTE + Tomek Links. This method uses tomek links (TL) to remove examples after applying SMOTE, which are considered being noisy or lying in the decision border. A tomek link is defined as a pair of examples x and y from different classes, that there exists no example z such that d(x,z) is lower than d(x,y) or d(y,z) is lower than d(x,y), where d is the distance metric.
3. SMOTE-ENN. ENN tends to remove more examples than the TL does, so it is expected that it will provide a more in depth data cleaning. ENN is used to remove examples from both classes. Thus, any example that is misclassified by its three nearest neighbors is removed from the training set.
4. SL-SMOTE. This method assigns each positive example its so called safe level before generating synthetic examples. The safe level of one example is defined as the number of positive instances among its k nearest neighbors. Each synthetic example is positioned closer to the example with the largest safe level so all synthetic examples are generated only in safe regions.
5. Borderline-SMOTE. This method only oversamples or strengthens the borderline minority examples. First, it finds out the borderline minority examples P, defined as the examples of the minority class with more than half, but not all, of their m nearest neighbors belonging to the majority class. Finally, for each of those examples, we calculate its k nearest neighbors from P (for the algorithm version B1-SMOTE) or from all the training data, also with majority examples (for the algorithm version B2-SMOTE) and operate similarly to SMOTE. Then, synthetic examples are generated from them and added to the original training set.

Apply SMOTE

```
## Keep original training data before SMOTE
X_train0 = X_train
y_train0 = y_train
```

```
sme = SMOTEENN()
X_train, y_train = sme.fit_sample(X_train, y_train)
print(X_train.shape)
print(y_train.shape)
unique(y_train, return_counts=True)
```

```
(354935, 30)
(354935,)

(array([0, 1]), array([172898, 182037]))
```

```
mean(y_train)  #Corresponds to counts from previous block
```

```
0.5128741882316481
```

```
a = X_train[:,29]   #Collect the Amount column
print(mean(a))
print(mean(a[y_train==0]))
print(mean(a[y_train==1]))
```

```
95.6875383234
86.585894458
104.332242538
```

Train & Predict

```
#DEFAULT:
#class sklearn.ensemble.RandomForestClassifier(n_estimators=10,
#criterion='gini', max_depth=None, min_samples_split=2,
#min_samples_leaf=1, min_weight_fraction_leaf=0.0,
#max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0,
#min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=1,
#random_state=None, verbose=0, warm_start=False, class_weight=None)[source]

clf = RandomForestClassifier(n_estimators=10)
clf = clf.fit(X_train,y_train)

y_test_hat = clf.predict(X_test)
```

Evaluate Predictions

While the standard accuracy metric makes our predictions look near-perfect, we should bear in mind that the class imbalance of the set skews this metric.

```
#In sample
y_train_hat = clf.predict(X_train0)
accuracy_score(y_train0,y_train_hat)
```

```
0.99989518918352371
```

```
#Out of sample
accuracy_score(y_test,y_test_hat)
```

```
0.99955312968814836
```

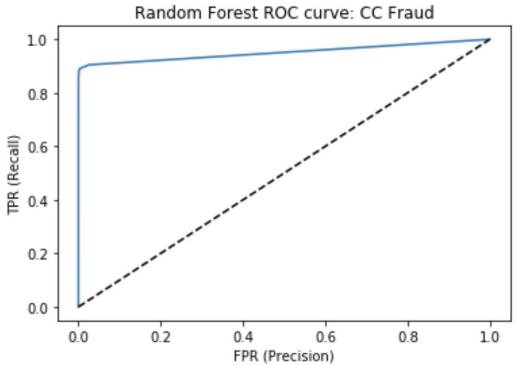
SciKitLearn's classification report gives us a more complete picture.

```
#print classification_report(y_test, y_test_hat)
print(classification_report(y_test, y_test_hat))
```

```
             precision    recall  f1-score   support

          0       1.00      1.00      1.00     93840
          1       0.87      0.84      0.86       147

avg / total       1.00      1.00      1.00     93987
```

```
y_score = clf.predict_proba(X_test)[:,1]
fpr, tpr, _ = roc_curve(y_test, y_score)

title('Random Forest ROC curve: CC Fraud')
xlabel('FPR (Precision)')
ylabel('TPR (Recall)')

plot(fpr,tpr)
plot((0,1), ls='dashed',color='black')
plt.show()
#print 'Area under curve (AUC): ', auc(fpr,tpr)
print('Area under curve (AUC): ', auc(fpr,tpr))
```

ROC



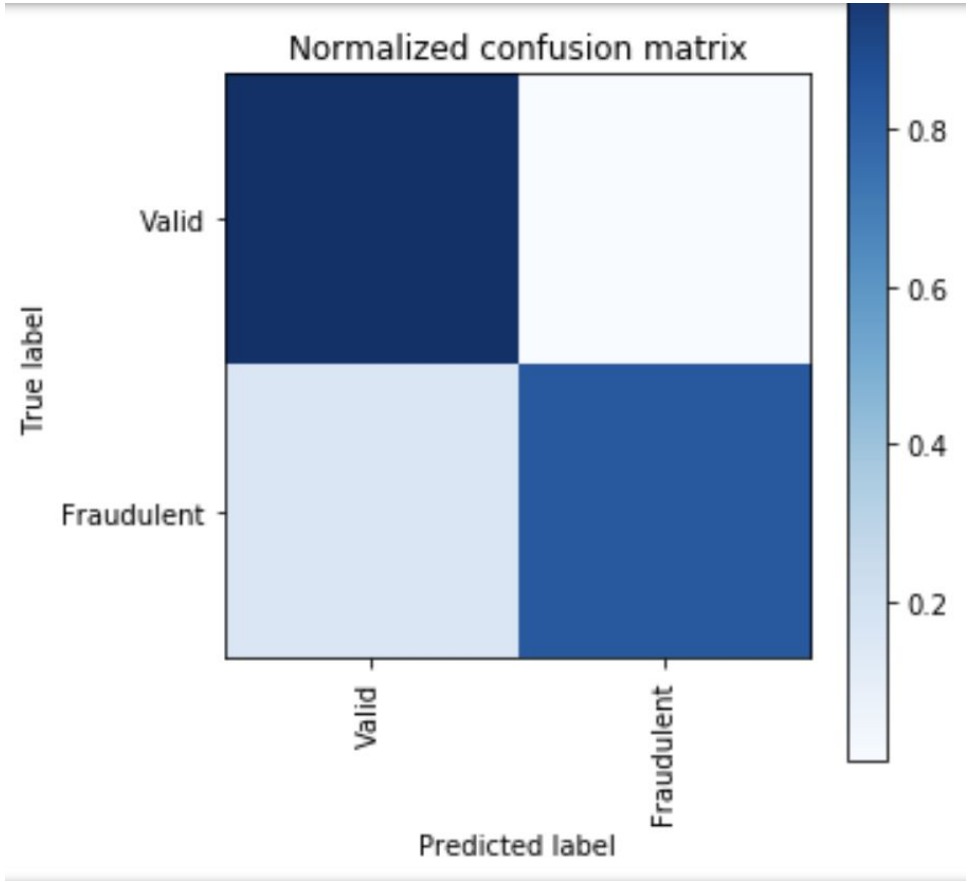
AUC

```
Area under curve (AUC):  0.950867303443
```

Confusion Matrix

```
def plot_confusion_matrix(cm, title='Confusion matrix', cmap=plt.cm.Blues):
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    class_labels = ['Valid','Fraudulent']
    plt.colorbar()

    tick_marks = np.arange(len(class_labels))
    plt.xticks(tick_marks, class_labels, rotation=90)
    plt.yticks(tick_marks, class_labels)
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
```

```
cm = confusion_matrix(y_test, y_test_hat)
cm_normalized = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
plt.figure(figsize=(5,5))
plot_confusion_matrix(cm_normalized, title='Normalized confusion matrix')
```




```
#Out of sample
print(cm)
print("False positive rate = ",cm[1][0]/sum(cm[1]))
```

```
[[93821    19]
 [   23   124]]
False positive rate =  0.156462585034
```

```
#Recheck in sample
cm = confusion_matrix(y_train0, y_train_hat)
print(cm)
print("False positive rate = ",cm[1][0]/sum(cm[1]))
```

```
[[190455     20]
 [     0    345]]
False positive rate =  0.0
```

Training and Test

Train and Predict

```
clf = RandomForestClassifier(n_estimators=10)
clf = clf.fit(X_train,y_train)
y_test_hat = clf.predict(X_test)
```

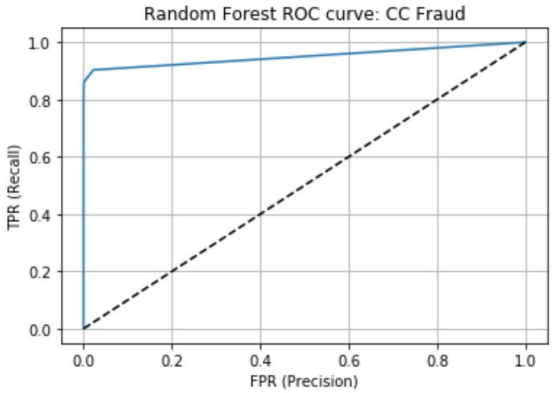
```
def plot_confusion_matrix(cm, title='Confusion matrix', cmap=plt.cm.Blues):
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    class_labels = ['Valid','Fraudulent']
    plt.colorbar()

    tick_marks = np.arange(len(class_labels))
    plt.xticks(tick_marks, class_labels, rotation=90)
    plt.yticks(tick_marks, class_labels)
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
```

```
#Out of sample
print("Accuracy = ",accuracy_score(y_test,y_test_hat))
cm = confusion_matrix(y_test, y_test_hat)
print("Confusion Matrix")
print(cm)
print("False positive rate = ",cm[1][0]/sum(cm[1]))
```

```
Accuracy =  0.999446731995
Confusion Matrix
[[93782    19]
 [   33   153]]
False positive rate =  0.177419354839
```

ROC Curve and AUC

```
y_score = clf.predict_proba(X_test)[:,1]
fpr, tpr, _ = roc_curve(y_test, y_score)

title('Random Forest ROC curve: CC Fraud')
xlabel('FPR (Precision)')
ylabel('TPR (Recall)')

plot(fpr,tpr); grid()
plot((0,1), ls='dashed',color='black')
plt.show()
#print 'Area under curve (AUC): ', auc(fpr,tpr)
print('Area under curve (AUC): ', auc(fpr,tpr))
```




```
Area under curve (AUC):  0.950015320698
```

Precision, Recall, F

- Precision = True positives/(True positives + False positives)
- Recall = True positives/(True positives + False negatives)
- F = 2 (Precision x Recall)/(Precision + Recall)

Recall = 153/186 = 0.8226

Precision = 153/172 = 0.8895

F = 0.8547

False positive rate = 1 - Precision = 0.1105

Actual

```
Accuracy =  0.999446731995
Confusion Matrix
[[93782    19] FP
 [   33   153]]
     FN    TP
```

Recall : fraction of correct cases marked by the algorithm. (1 - Type 1) (Sensitivity)

Precision: number of cases marked correct that are actually correct. (1 - Type 2) (Specificity)

Credit Scoring with Social Data

MARKETING SCIENCE
Articles in Advance, pp. 1–25
ISSN 0732-2399 (print) | ISSN 1526-548X (online)

http://dx.doi.org/10.1287/mksc.2015.0949

Credit Scoring with Social Network Data

Yanhao Wei
Department of Economics, University of Pennsylvania, Philadelphia, Pennsylvania 19104, yanhao@sas.upenn.edu

Pinar Yildirim, Christophe Van den Bulte
Marketing Department, The Wharton School, University of Pennsylvania, Philadelphia, Pennsylvania 19104
{pyild@wharton.upenn.edu, vdbulte@wharton.upenn.edu}

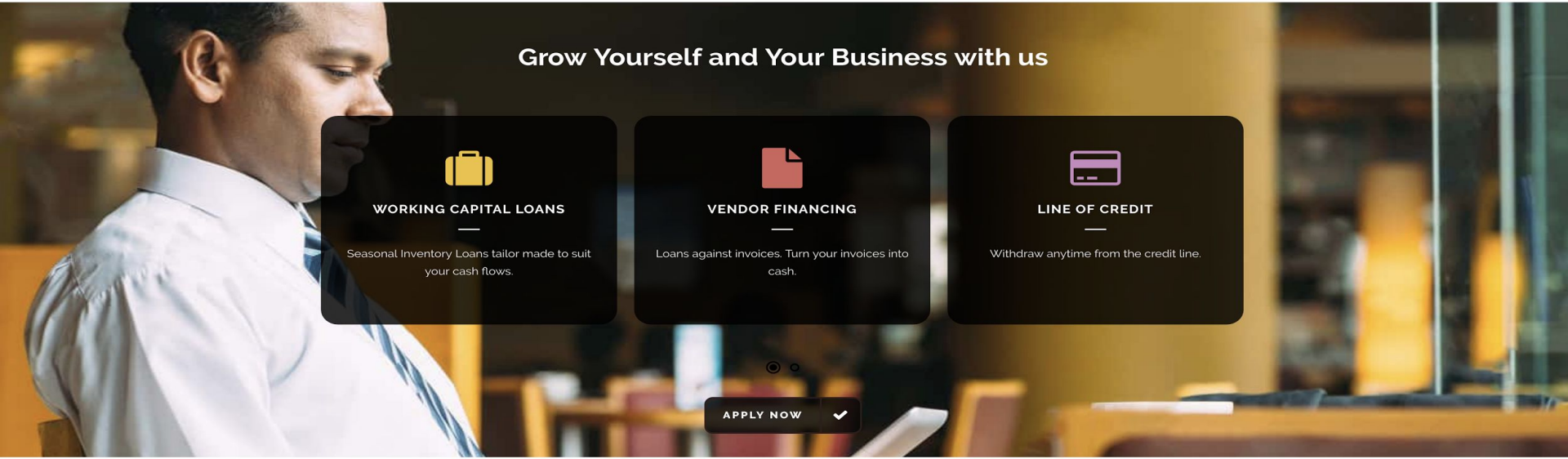
Chrysanthos Dellarocas
Information Systems Department, Questrom School of Business, Boston University, Boston, Massachusetts 02215, dell@bu.edu

Motivated by the growing practice of using social network data in credit scoring, we analyze the impact of using network-based measures on customer score accuracy and on tie formation among customers. We develop a series of models to compare the accuracy of customer scores obtained with and without network data. We also investigate how the accuracy of social network-based scores changes when consumers can strategically construct their social networks to attain higher scores. We find that those who are motivated to improve their scores may form fewer ties and focus more on similar partners. The impact of such endogenous tie formation on the accuracy of consumer scores is ambiguous. Scores can become more accurate as a result of modifications in social networks, but this accuracy improvement may come with greater network fragmentation. The threat of social exclusion in such endogenously formed networks provides incentives to low-type members to exert effort that improves everyone's creditworthiness. We discuss implications for managers and public policy.

Keywords: social networks; credit score; customer scoring; social status; social discrimination; endogenous tie formation

FLEXILOANS

LOGIN



Grow Yourself and Your Business with us

WORKING CAPITAL LOANS

Seasonal Inventory Loans tailor made to suit your cash flows.

VENDOR FINANCING

Loans against invoices. Turn your invoices into cash.

LINE OF CREDIT

Withdraw anytime from the credit line.

APPLY NOW

Our Process

FAST, FLEXIBLE AND PAPERLESS BUSINESS LOANS WITHOUT COLLATERAL

FILL ONLINE FORM

Fill the online Application Form in less than 15 minutes

UPLOAD DOCUMENTS

On submission, FlexiLoans generates Credit Score in less than a day

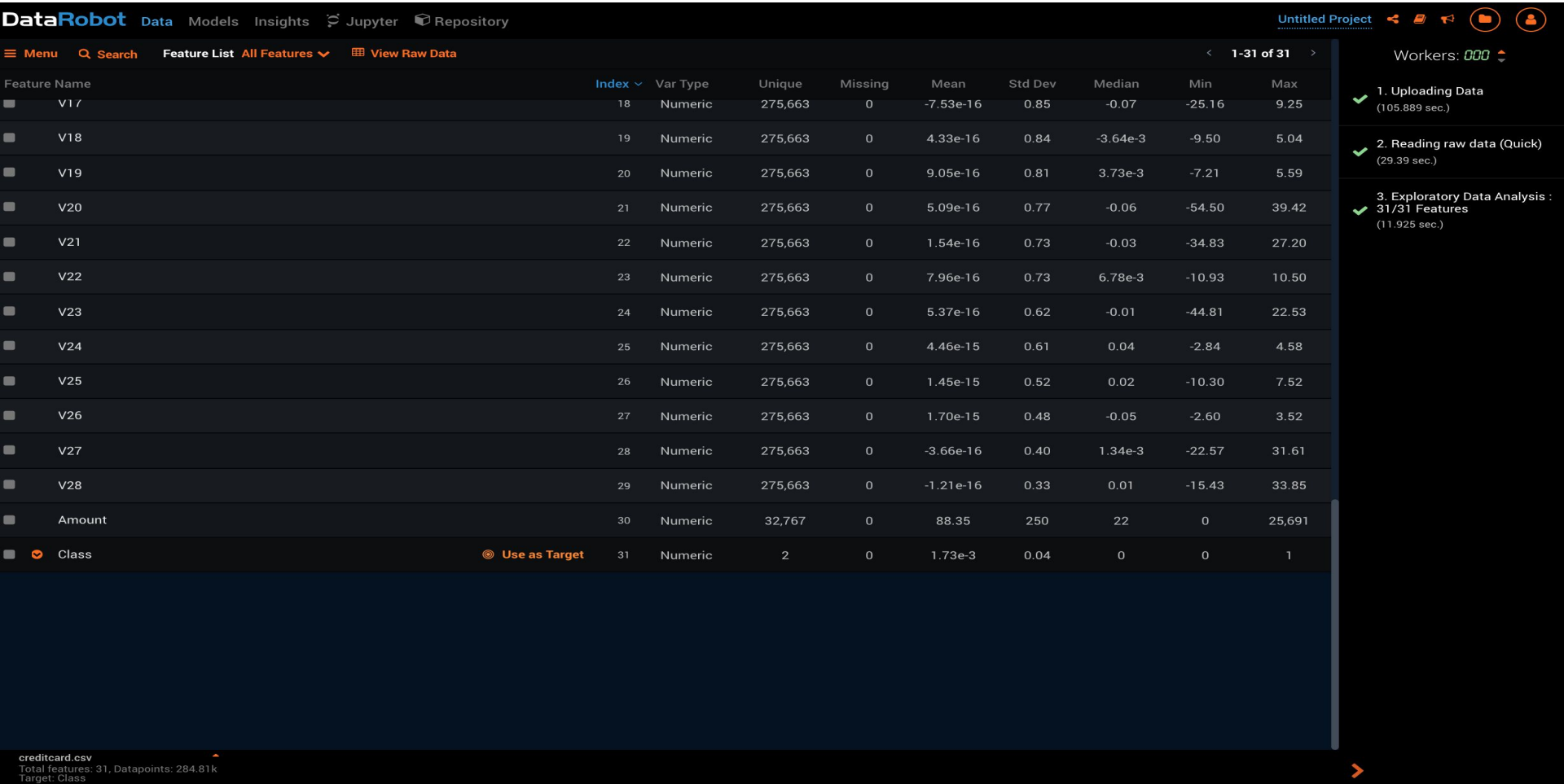
GET APPROVED

Sign the documents and get the loan approved in less than 48hrs

Fraud Detection with DataRobot

- Kaggle Credit Card Fraud Detection:
  - Notebook: CC_Fraud_RF.ipynb
  - Data: creditcard.csv
- Lending Club Data Analysis

Using DataRobot for Credit Card Fraud Detection



Feature Name		Index	Var Type	Unique	Missing	Mean	Std Dev	Median	Min	Max
V17		18	Numeric	275,663	0	-7.53e-16	0.85	-0.07	-25.16	9.25
V18		19	Numeric	275,663	0	4.33e-16	0.84	-3.64e-3	-9.50	5.04
V19		20	Numeric	275,663	0	9.05e-16	0.81	3.73e-3	-7.21	5.59
V20		21	Numeric	275,663	0	5.09e-16	0.77	-0.06	-54.50	39.42
V21		22	Numeric	275,663	0	1.54e-16	0.73	-0.03	-34.83	27.20
V22		23	Numeric	275,663	0	7.96e-16	0.73	6.78e-3	-10.93	10.50
V23		24	Numeric	275,663	0	5.37e-16	0.62	-0.01	-44.81	22.53
V24		25	Numeric	275,663	0	4.46e-15	0.61	0.04	-2.84	4.58
V25		26	Numeric	275,663	0	1.45e-15	0.52	0.02	-10.30	7.52
V26		27	Numeric	275,663	0	1.70e-15	0.48	-0.05	-2.60	3.52
V27		28	Numeric	275,663	0	-3.66e-16	0.40	1.34e-3	-22.57	31.61
V28		29	Numeric	275,663	0	-1.21e-16	0.33	0.01	-15.43	33.85
Amount		30	Numeric	32,767	0	88.35	250	22	0	25,691
Class	Use as Target	31	Numeric	2	0	1.73e-3	0.04	0	0	1

Workers: 000

1. Uploading Data (105.889 sec.)
2. Reading raw data (Quick) (29.39 sec.)
3. Exploratory Data Analysis : 31/31 Features (11.925 sec.)

Hide Advanced Options

Advanced Options

Partitioning | Smart Downsampling | Additional

Select partitioning method:

Random | Partition Feature | Group | Date/Time | Stratified

Rows are assigned in a way that ensures similar target distribution across each partition.

Run models using:

Cross-Validation | Train-Validation-Holdout

Number of cross-validation (CV) folds:

The number of CV folds defined between 2 - 50.

5

Holdout percentage:

The percentage of data allocated to the holdout set -- must be between 0% - 98%.

20

Explore creditcard.csv

Menu Search Feature List Informative Features View Raw Data 1-31 of 31

Feature Name	Index	Var Type	Unique	Missing	Mean	Std Dev	Median	Min	Max
V9	10	Numeric	275,663	0	-3.14e-15	1.10	-0.05	-13.43	15.59
V8	9	Numeric	275,663	0	-1.93e-16	1.19	0.02	-73.22	20
V7	8	Numeric	275,663	0	-1.69e-15	1.24	0.04	-43.56	121
V6	7	Numeric	275,663	0	2.01e-15	1.33	-0.27	-26.16	73.30

creditcard.csv
Total features: 31, Datapoints: 284.81k
Target: Class

Workers: 000

1. Uploading Data (105.889 sec.)
2. Reading raw data (Quick) (29.39 sec.)
3. Exploratory Data Analysis : 31/31 Features (11.925 sec.)

Hide Advanced Options

Advanced Options

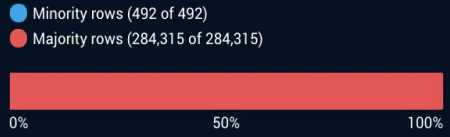
Partitioning | Smart Downsampling | Additional

Downsample Data

For classification or zero-inflated regression problems this will allow you to downsample the majority class in order to build faster models with similar accuracy.

Majority class downsampling percentage:
This must be between 1% - 100%.

100



Results of downsampling for 284,807 total nonmissing rows:

Minority rows 492 (of original 492)

Majority rows 284,315 (of original 284,315)

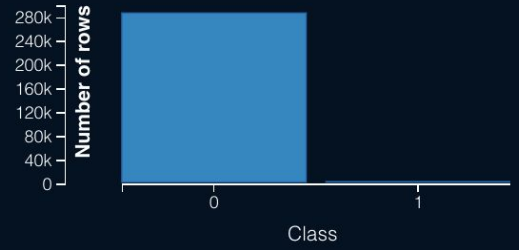
Explore creditcard.csv

Menu Search Feature List Informative Features View Raw Data 1-31 of 31

Feature Name	Index	Var Type	Unique	Missing	Mean	Std Dev	Median	Min	Max
V9	10	Numeric	275,663	0	-3.14e-15	1.10	-0.05	-13.43	15.59
V8	9	Numeric	275,663	0	-1.93e-16	1.19	0.02	-73.22	20
V7	8	Numeric	275,663	0	-1.69e-15	1.24	0.04	-43.56	121

What would you like to predict?

Class CLASSIFICATION



Start

Modeling Mode: Autopilot

Feature list: Informative Features

Optimization Metric: LogLoss

Time-Aware Modeling

This dataset cannot use time-aware modeling because it does not contain any date features.

Set up time-aware modeling

Menu | Search | Feature List All Features | View Raw Data | Create new feature list | CREATE | 1-31 of 31

Feature Name	Index	Var Type	Unique	Missing	Mean	Std Dev	Median	Min	Max
Class	31	Numeric	2	0					
Time	1	Numeric	124,592	0	94,814	47,488	84,692	0	172,792
V1	2	Numeric	275,663	0	3.92e-15	1.96	0.02	-56.41	2.45
V2	3	Numeric	275,663	0	5.69e-16	1.65	0.07	-72.72	22.06
V3	4	Numeric	275,663	0	-8.77e-15	1.52	0.18	-48.33	9.38
V4	5	Numeric	275,663	0	2.78e-15	1.42	-0.02	-5.68	16.88
V5	6	Numeric	275,663	0	-1.55e-15	1.38	-0.05	-113.74	34.80
V6	7	Numeric	275,663	0	2.01e-15	1.33	-0.27	-26.16	73.30
V7	8	Numeric	275,663	0	-1.69e-15	1.24	0.04	-43.56	121
V8	9	Numeric	275,663	0	-1.93e-16	1.19	0.02	-73.22	20

Workers: 004

1. Setting target feature (0.012 sec.)
2. Creating CV and Holdout partitions
40% Complete
3. Characterizing target variable
4. Loading dataset and preparing data
5. Saving target and partitioning information
6. Analyzing features
7. Calculating list of models

Menu | Search | Feature List All Features | View Raw Data | Create new feature list | CREATE | 1-31 of 31

Feature Name	Index	Importance	Var Type	Unique	Missing	Mean	Std Dev	Median	Min	Max
Class	31	Target	Numeric	2	0	1.73e-3	0.04	0	0	1
V17	18		Numeric	221,418	0	5.58e-4	0.84	-0.07	-23.24	9.25
V14	15		Numeric	221,418	0	-2.54e-4	0.96	0.05	-18.82	10.53
V12	13		Numeric	221,418	0	2.47e-4	1.00	0.14	-18.68	7.85
V10	11		Numeric	221,418	0	-4.42e-4	1.09	-0.09	-24.40	23.75
V21	22		Numeric	221,418	0	-4.55e-4	0.73	-0.03	-34.83	27.20
V16	17		Numeric	221,418	0	-2.35e-4	0.88	0.07	-13.30	17.32
V27	28		Numeric	221,418	0	-4.63e-6	0.40	1.36e-3	-22.57	31.61
V28	29		Numeric	221,418	0	-7.92e-4	0.33	0.01	-15.43	33.85
V6	7		Numeric	221,418	0	1.12e-3	1.33	-0.27	-26.16	73.30
V20	21		Numeric	221,418	0	-1.07e-5	0.76	-0.06	-54.50	39.42
V3	4		Numeric	221,418	0	2.86e-3	1.51	0.18	-48.33	9.38
V23	24		Numeric	221,418	0	1.45e-4	0.62	-0.01	-44.81	22.53
Amount	30		Numeric	29,390	0	88.26	249	22	0	25,691

Workers: 004

Processing (4)

TensorFlow Neural Network Classi...
16.00% sample , CV #1
0%
0.3 GB RAM
1.4 CPUs

RandomForest Classifier (Gini) (B...
16.00% sample , CV #1
0%
0.2 GB RAM
4.1 CPUs

Queue (22)

Logistic Regression (30)
16.00% sample , CV #1

Regularized Logistic Regression (...
16.00% sample , CV #1

Majority Class Classifier (32)
16.00% sample , CV #1

Gradient Boosted Trees Classifier...
16.00% sample , CV #1

Regularized Logistic Regression (...
16.00% sample , CV #1

Light Gradient Boosted Trees Cla...
16.00% sample , CV #1

eXtreme Gradient Boosted Trees ...
16.00% sample , CV #1

Elastic-Net Classifier (L2 / Binomi...

Leaderboard | Learning Curves | Speed vs Accuracy | Model Comparison

Menu | Search | + Add New Model

Metric LogLoss

Model Name & Description	Feature List & Sample Size	Validation	Cross Validation	Holdout
eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) Tree-based Algorithm Preprocessing v1 M12 BP38	Informative Features 16.0 %	0.0029	Run	
RandomForest Classifier (Gini) Missing Values Imputed | RandomForest Classifier (Gini) M3 BP27 REF	Informative Features 16.0 %	0.0032	Run	
Regularized Logistic Regression (L2) Constant Splines | Regularized Logistic Regression (L2) M10 BP36 β_i	Informative Features 16.0 %	0.0036	Run	
TensorFlow Neural Network Classifier Regularized Linear Model Preprocessing v4 M1 BP25	Informative Features 16.0 %	0.0043	Run	
Regularized Logistic Regression (L2) Missing Values Imputed | Standardize | Regularized Logistic Regression (L2) M7 BP31 β_i	Informative Features 16.0 %	0.0044	Run	
Naive Bayes combiner classifier Missing Values Imputed | Gaussian Naive Bayes classifier (scikit-learn) | Naive Bayes combiner classifier | Calibrate predictions M2 BP26 REF	Informative Features 16.0 %	0.0081	Run	
Logistic Regression Missing Values Imputed | Standardize | Logistic Regression M6 BP30 β_i	Informative Features 16.0 %	0.0094	Run	
Majority Class Classifier Majority Class Classifier M8 BP32	Informative Features 16.0 %	0.0127	Run	
Gradient Boosted Trees Classifier Missing Values Imputed | Gradient Boosted Trees Classifier M4 BP28 REF	Informative Features 16.0 %	0.0144	Run	

Workers: 004

Processing (4)

Gradient Boosted Trees Classifier ...
16.00% sample , CV #1
0% 0.0 GB
2.5 CPUs RAM

Light Gradient Boosted Trees Clas...
16.00% sample , CV #1
0% 0.3 GB
1.1 CPUs RAM

Elastic-Net Classifier (L2 / Binomi...
16.00% sample , CV #1
0% 0.3 GB
4.1 CPUs RAM

Queue (13)

RandomForest Classifier (Gini) (41)
16.00% sample , CV #1

Vowpal Wabbit Classifier (42)
16.00% sample , CV #1

Generalized Additive2 Model (43)
16.00% sample , CV #1

Regularized Logistic Regression (...
16.00% sample , CV #1

Elastic-Net Classifier (L2 / Binomi...
16.00% sample , CV #1

Elastic-Net Classifier (mixing alp...
16.00% sample , CV #1

Stochastic Gradient Descent Clas...
16.00% sample , CV #1

Balanced RandomForest Classifie...
16.00% sample , CV #1

Balanced ExtraTrees Classifier (G...
16.00% sample , CV #1

Leaderboard | Learning Curves | Speed vs Accuracy | Model Comparison

Menu | Search | + Add New Model | Metric LogLoss

Model Name & Description	Feature List & Sample Size	Validation	Cross Validation	Holdout
eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) Tree-based Algorithm Preprocessing v1 M29 BP38	Informative Features 32.0 %	0.0024	Run	
Light Gradient Boosted Trees Classifier with Early Stopping Tree-based Algorithm Preprocessing v1 M11 BP37	Informative Features 16.0 %	0.0029	Run	
eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) Tree-based Algorithm Preprocessing v1 M12 BP38	Informative Features 16.0 %	0.0029	Run	
eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) Tree-based Algorithm Preprocessing v20 M24 BP50	Informative Features 16.0 %	0.0030	Run	
eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) and Unsupervised Learning Features Tree-based Algorithm Preprocessing v22 with Unsupervised Learning Features M25 BP51	Informative Features 16.0 %	0.0031	Run	
Generalized Additive2 Model Missing Values Imputed | Generalized Additive2 Model | Text fit on Residuals (L2 / Binomial Deviance) M17 BP43	Informative Features 16.0 %	0.0032	Run	
RandomForest Classifier (Gini) Missing Values Imputed | RandomForest Classifier (Gini) M3 BP27 REF	Informative Features 16.0 %	0.0032	Run	
Nystroem Kernel SVM Classifier Regularized Linear Model Preprocessing v20 M14 BP40	Informative Features 16.0 %	0.0034	Run	
Elastic-Net Classifier (L2 / Binomial Deviance) with Binned numeric features Missing Values Imputed | Elastic-Net Classifier (L2 / Binomial Deviance) with Binned numeric features M19 BP45	Informative Features 16.0 %	0.0035	Run	

Workers: 004

Processing (4)

- Light Gradient Boosted Trees Clas... 32.00% sample , CV #1 — 0% — 0.3 GB RAM — 1.5 CPUs
- eXtreme Gradient Boosted Trees C... 32.00% sample , CV #1 — 0% — 0.3 GB RAM — 3.4 CPUs
- eXtreme Gradient Boosted Trees C... 32.00% sample , CV #1 — 0% — 0.4 GB RAM — 4.1 CPUs
- Generalized Additive2 Model (BP43) 32.00% sample , CV #1

Queue (11)

- RandomForest Classifier (Gini) (27) 32.00% sample , CV #1
- Nystroem Kernel SVM Classifier (... 32.00% sample , CV #1
- Elastic-Net Classifier (L2 / Binomi... 32.00% sample , CV #1
- Regularized Logistic Regression (... 32.00% sample , CV #1
- Vowpal Wabbit Classifier (42) 32.00% sample , CV #1
- Vowpal Wabbit Classifier (53) 32.00% sample , CV #1
- Regularized Logistic Regression (... 32.00% sample , CV #1
- Balanced ExtraTrees Classifier (G... 32.00% sample , CV #1
- Elastic-Net Classifier (L2 / Binomi...

Leaderboard | Learning Curves | Speed vs Accuracy | Model Comparison

Menu | Search | + Add New Model

Metric LogLoss

Workers: 004

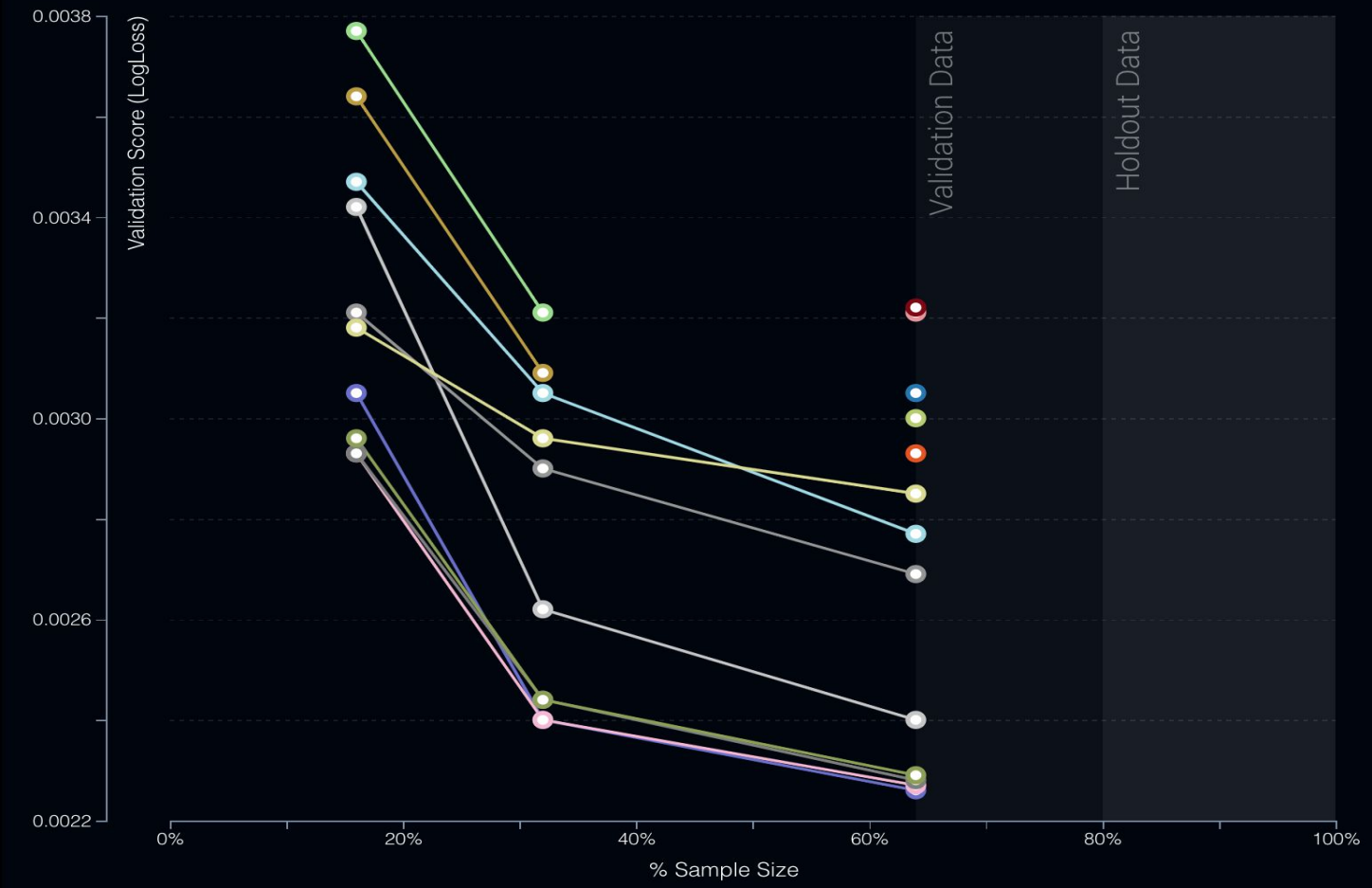
Autopilot has finished

Run Autopilot on a Different Feature List

Unlock Holdout

Model Name & Description	Feature List & Sample Size	Validation	Cross Validation	Holdout
eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) and Unsupervised Learning Features Tree-based Algorithm Preprocessing v22 with Unsupervised Learning Features M45 BP51	Informative Features 64.0 %	0.0023	Run	
Light Gradient Boosted Trees Classifier with Early Stopping Tree-based Algorithm Preprocessing v1 M44 BP37	Informative Features 64.0 %	0.0023	Run	
eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) Tree-based Algorithm Preprocessing v1 M47 BP38	Informative Features 64.0 %	0.0023	Run	
eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) Tree-based Algorithm Preprocessing v20 M46 BP50	Informative Features 64.0 %	0.0023	Run	
Light Gradient Boosted Trees Classifier with Early Stopping Tree-based Algorithm Preprocessing v1 M28 BP37	Informative Features 32.0 %	0.0024	Run	
Nystroem Kernel SVM Classifier Regularized Linear Model Preprocessing v20	Informative Features 64.0 %	0.0024	Run	

Leaderboard | Learning Curves | Speed vs Accuracy | Model Comparison



Feature List All Features

- eXtreme Gradient Boosted Trees Classi...
  BP51
  Informative Features
- Light Gradient Boosted Trees Classifi...
  BP37
  Informative Features
- eXtreme Gradient Boosted Trees Classi...
  BP38
  Informative Features
- eXtreme Gradient Boosted Trees Classi...
  BP50
  Informative Features
- Nystroem Kernel SVM Classifier
  BP40
  Informative Features
- RandomForest Classifier (Gini)
  BP27
  Informative Features
- Elastic-Net Classifier (L2 / Binomial...
  BP45
  Informative Features
- Generalized Additive2 Model
  BP43
  Informative Features
- AVG Blender
  M44+47+45
  Informative Features
- GLM Blender
  M44+47+45
  Informative Features
- Advanced AVG Blender
  M49+44+47+48+50+...
  Informative Features
- Balanced ExtraTrees Classifier (Gini)
  BP49
  Informative Features
- Elastic-Net Classifier (L2 / Binomial...
  BP39
  Informative Features

Leaderboard | Learning Curves | Speed vs Accuracy | Model Comparison

Menu | Search | + Add New Model | Metric LogLoss

Model Name & Description	Feature List & Sample Size	Validation	Cross Validation	Holdout
eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) and Unsupervised Learning Features Tree-based Algorithm Preprocessing v22 with Unsupervised Learning Features M45 BP51	Informative Features 64.0 %	0.0023	In Progress	
Light Gradient Boosted Trees Classifier with Early Stopping Tree-based Algorithm Preprocessing v1 M44 BP37	Informative Features 64.0 %	0.0023	Run	
eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) Tree-based Algorithm Preprocessing v1 M47 BP38	Informative Features 64.0 %	0.0023	Run	
eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) Tree-based Algorithm Preprocessing v20 M46 BP50	Informative Features 64.0 %	0.0023	Run	
Light Gradient Boosted Trees Classifier with Early Stopping Tree-based Algorithm Preprocessing v1 M28 BP37	Informative Features 32.0 %	0.0024	Run	
Nystroem Kernel SVM Classifier Regularized Linear Model Preprocessing v20 M48 BP40	Informative Features 64.0 %	0.0024	Run	

Workers: 004

Run Autopilot on a Different Feature List

Unlock Holdout

Processing (4)

eXtreme Gradient Boosted Trees C... ×
64.00% sample , CV #2
0%
0.3 GB RAM
3.3 CPUs

eXtreme Gradient Boosted Trees C... ×
64.00% sample , CV #3
0%
0.3 GB RAM
0.8 CPUs

eXtreme Gradient Boosted Trees C... ×
64.00% sample , CV #4
0%
0.4 GB RAM
3.2 CPUs

eXtreme Gradient Boosted Trees C... ×
64.00% sample , CV #5
0%
0.4 GB RAM
3.3 CPUs

eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) and Unsupervised Learning Features

BP51 M45
64% Sample Size, Informative Features

Change Model

vs

Dual Lift Lift Roc Curve

RandomForest Classifier (Gini)

BP27 M49
64% Sample Size, Informative Features

Change Model

Number of Bins : 15

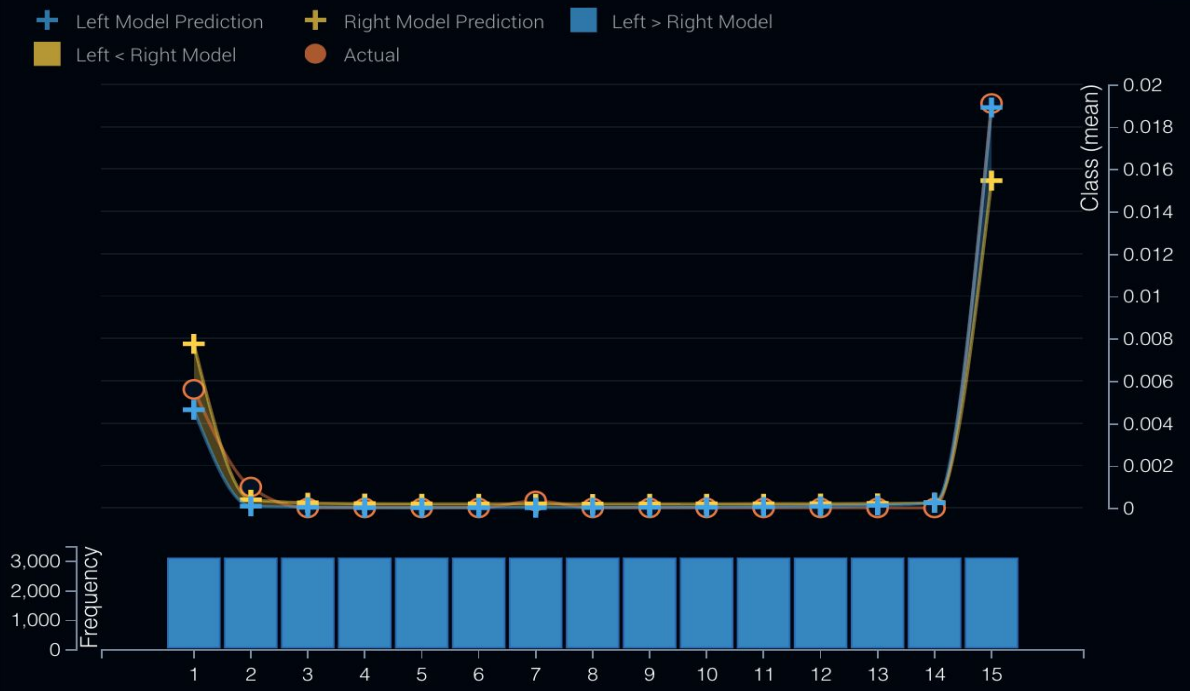
LogLoss (Validation) :
2.2600e-3

LogLoss (Cross Validation) :
2.5560e-3

Gini Norm (Validation) :
0.9642

Gini Norm (Cross Validation) :
0.9678

Prediction Time:
4984.05 ms



LogLoss (Validation) :
2.6900e-3

LogLoss (Cross Validation) :
NA

Gini Norm (Validation) :
0.9707

Gini Norm (Cross Validation) :
NA

Prediction Time:
4514.50 ms

eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) and Unsupervised Learning Features

BP51 M45
64% Sample Size, Informative Features

Change Model

LogLoss (Validation) :
2.2600e-3

LogLoss (Cross Validation) :
2.5560e-3

Gini Norm (Validation) :
0.9642

Gini Norm (Cross Validation) :
0.9678

Prediction Time:
4984.05 ms

VS

Dual Lift Lift Roc Curve

RandomForest Classifier (Gini)

BP27 M49
64% Sample Size, Informative Features

Change Model

LogLoss (Validation) :
2.6900e-3

LogLoss (Cross Validation) :
NA

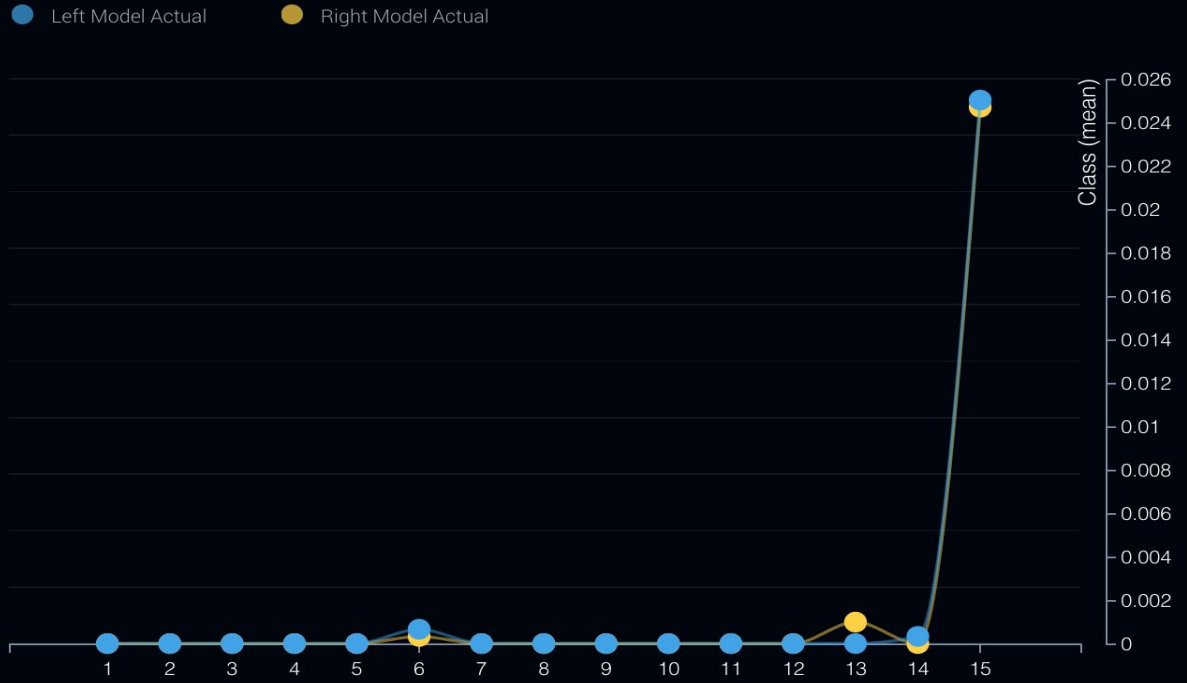
Gini Norm (Validation) :
0.9707

Gini Norm (Cross Validation) :
NA

Prediction Time:
4514.50 ms

Lift Data Source: Validation

Number of Bins : 15



eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) and Unsupervised Learning Features

BP51 M45
64% Sample Size, Informative Features

Change Model

LogLoss (Validation) :
2.2600e-3

LogLoss (Cross Validation) :
2.5560e-3

Gini Norm (Validation) :
0.9642

Gini Norm (Cross Validation) :
0.9678

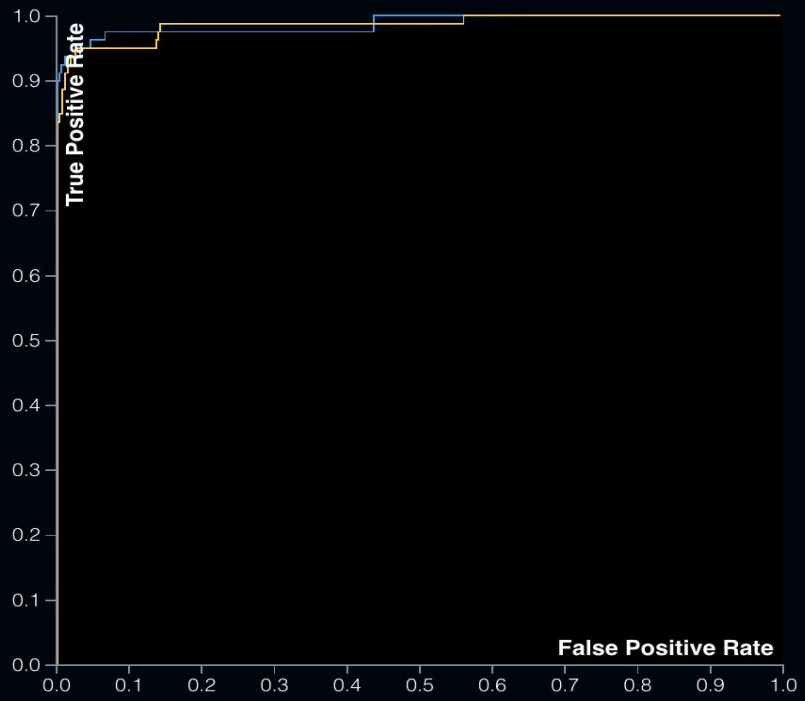
AUC (Validation) :
0.9821

Prediction Time:
4984.05 ms

vs

Dual Lift Lift Roc Curve

Roc Curve Data Source: Validation



RandomForest Classifier (Gini)

BP27 M49
64% Sample Size, Informative Features

Change Model

LogLoss (Validation) :
2.6900e-3

LogLoss (Cross Validation) :
NA

Gini Norm (Validation) :
0.9707

Gini Norm (Cross Validation) :
NA

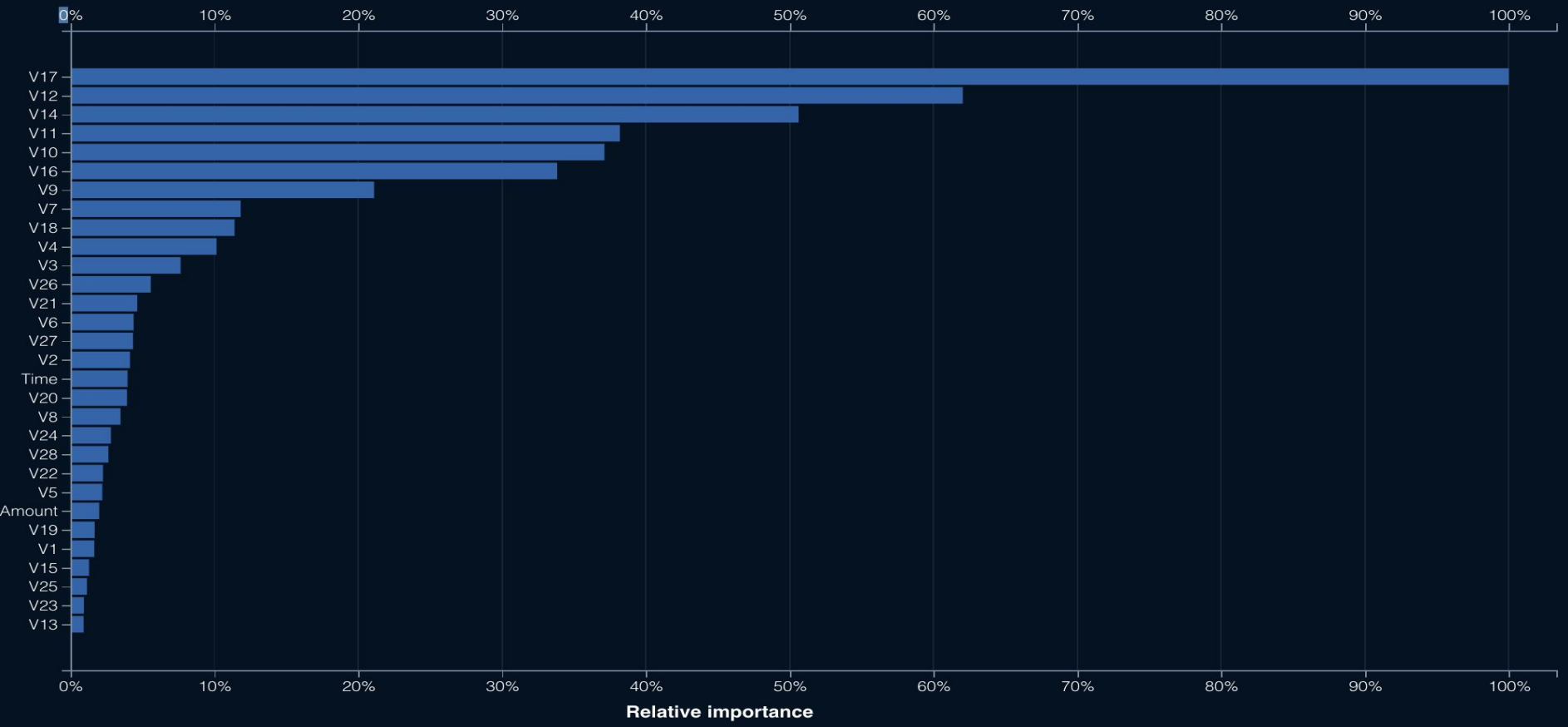
AUC (Validation) :
0.9854

Prediction Time:
4514.50 ms

RandomForest Classifier (Gini)

M49 BP27 | 64% Sample Size | Informative Features

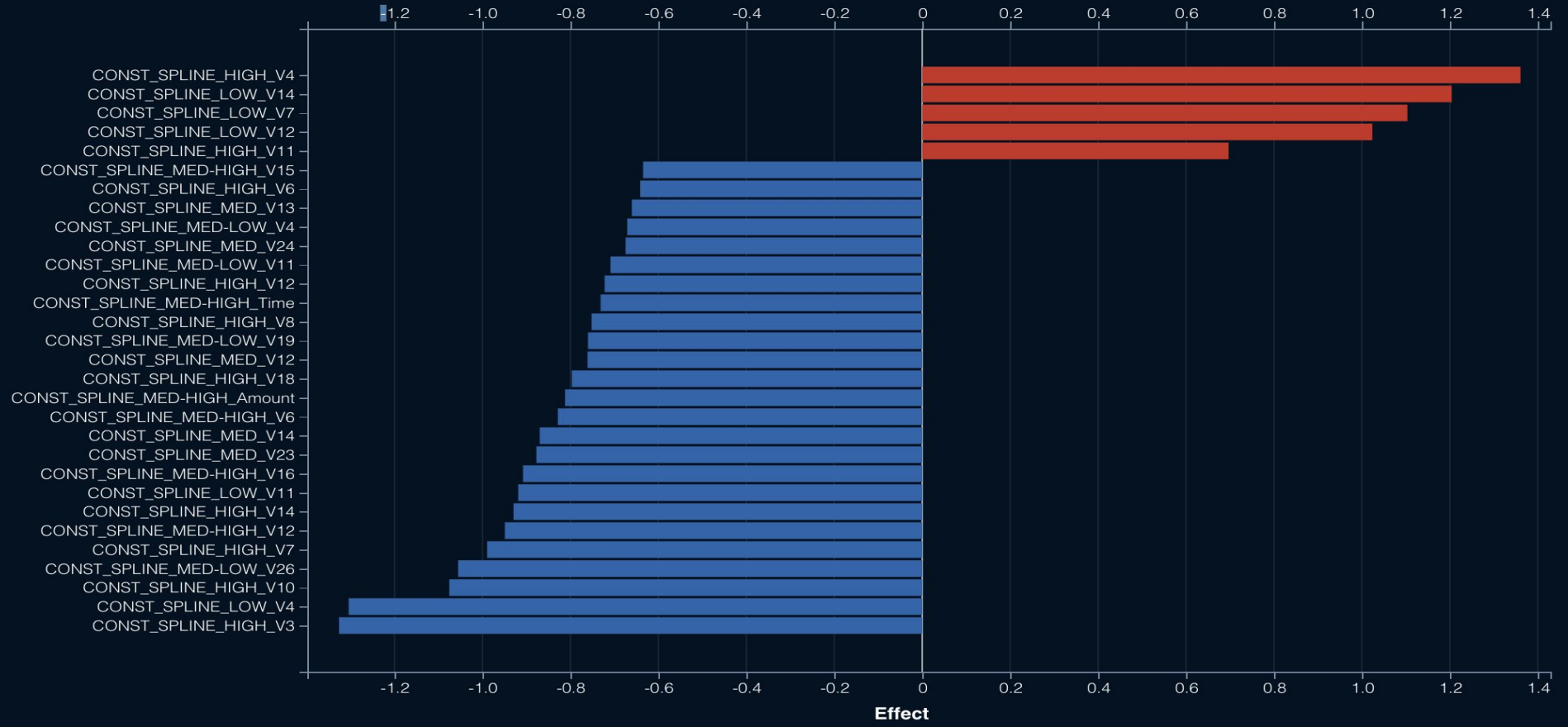
Sort by Feature Importance Export



Regularized Logistic Regression (L2)

M39 BP36 | 32% Sample Size | Informative Features

Sort by Feature Effect Export



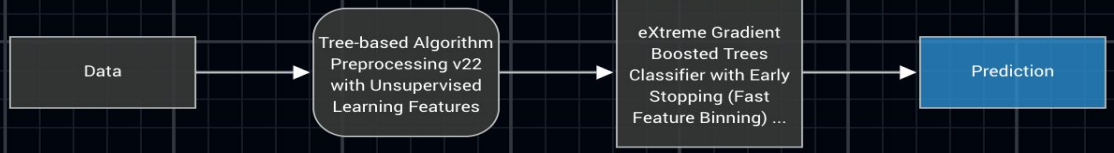
Leaderboard | Learning Curves | Speed vs Accuracy | Model Comparison

Menu | Search | + Add New Model | Metric LogLoss

Model Name & Description	Feature List & Sample Size	Validation	Cross Validation	Holdout
XG Boost eXtreme Gradient Boosted Trees Classifier with Early Stopping (Fast Feature Binning) and Unsupervised Learning Features Tree-based Algorithm Preprocessing v22 with Unsupervised Learning Features M45 BP51	Informative Features 64.0 %	0.0023	0.0026	

Blueprint | Lift Chart | Model X-Ray | Feature Impact | Model Info | Model Log | ROC Curve | Advanced Tuning | Deploy Model | Predict | Reason Codes | Download

1.0x



Model Name & Description | Feature List & Sample Size | Validation | Cross Validation | Holdout

Selection Summary

Export

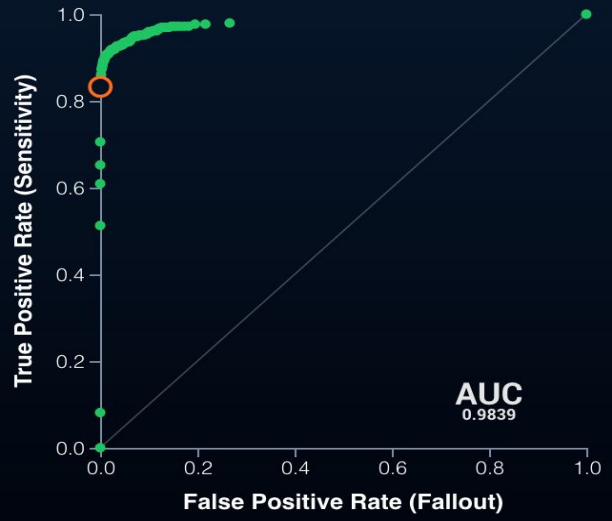
Maximizes F1 Score − NOT(1) + 1

F1 Score	True Positive Rate (Sensitivity)	False Positive Rate (Fallout)	True Negative Rate (Specificity)	Positive Predictive Value (Precision)	Negative Predictive Value	Accuracy	Matthews Correlation Coefficient
0.821	0.8325	0.0003	0.9997	0.8099	0.9997	0.9994	0.8208

		Predicted −	Predicted +	
Actual	−	227374 (TN)	77 (FP)	227451
	+	66 (FN)	328 (TP)	394
		227440	405	227845

ROC Curve

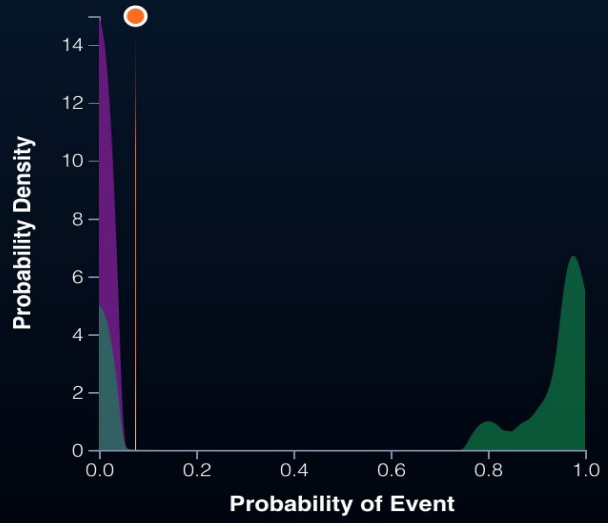
Data Source: Cross Validation



Prediction Distribution

Threshold (0-1): 0.0738

Density



Lending Club Analysis

Measures of Fit for Defaulted

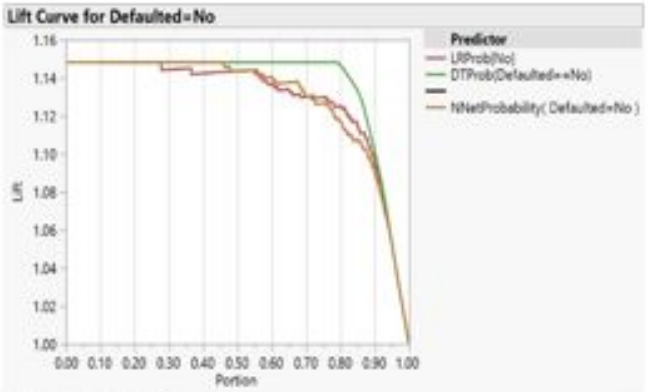
Creator	.2 .4 .6 .8	Entropy RSquare	Generalized RSquare	Mean -Log p	RMSE	Mean Abs Dev	Misclassification Rate	N	AUC
Fit Nominal Logistic		0.5913	0.6810	0.1571	0.2058	0.0894	0.0530	1000	0.9535
Partition		0.7784	0.8395	0.0852	0.1642	0.0599	0.0390	1000	0.9903
K Nearest Neighbors		.	.	.	.	.	0.1200	1000	.
Neural		0.5262	0.6203	0.1822	0.2267	0.1221	0.0700	1000	0.9507

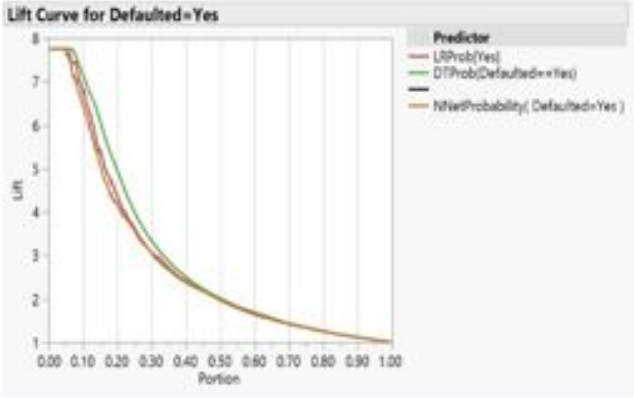
Receiver Operating Characteristics (ROC)

Measure	Logistic Regression	Decision Tree	Neural Networks	K-Nearest Neighbor
AUC = No	.95	.99	.95	0
AUC = YES	.95	.99	.95	0

Confusion Matrix

Parameter	Logistic Regression	Decision Tree	Neural Networks	K-Nearest Neighbor
Accuracy	.95	.96	.96	.88
Sensitivity	.69	.85	.50	.07
Specificity	.99	.98	.99	.99





Lending Club Loan History

Performance by Year (36-month loans)

	ROI	Avg Rate	Loss	Principal	Count	Completed	Total Loss
2007	-3.44%	12.21%	14.81%	$4,791,550.00	603	100.00%	$904,605.25
2008	-0.27%	12.37%	11.82%	$19,980,228.00	2,395	100.00%	$3,090,375.20
2009	4.56%	12.81%	7.41%	$51,809,560.00	5,279	100.00%	$5,004,401.00
2010	5.86%	11.68%	5.00%	$87,524,488.00	9,156	100.00%	$5,815,202.00
2011	5.81%	10.79%	4.18%	$132,530,984.00	14,101	64.19%	$7,355,289.00
2012	6.38%	12.98%	5.84%	$507,669,440.00	43,470	0.00%	$36,866,752.00
2013	8.19%	12.95%	4.10%	$1,271,489,020.00	100,380	0.00%	$47,034,752.00
2014	9.61%	12.48%	2.38%	$912,224,700.00	71,779	0.00%	$9,564,484.00

Lending Club 2013 - 2014

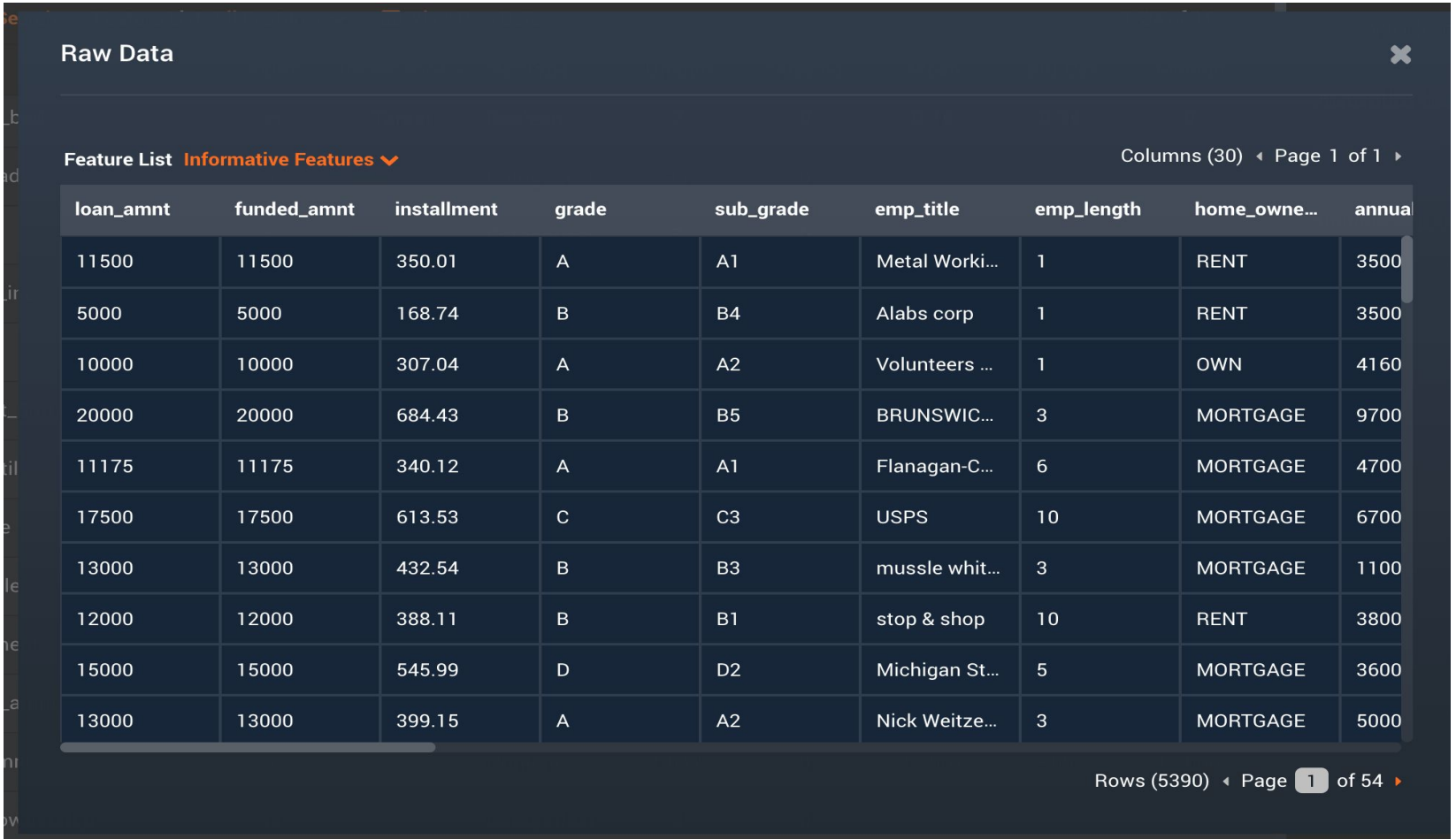
	ROI	Avg Rate	Loss	Principal	Count	Avg Age (Months)	Total Loss
2013	8.19%	12.95%	4.10%	$1,271,489,020.00	100,380	14.93	$47,034,752.00
2014	9.61%	12.48%	2.38%	$912,224,700.00	71,779	6.53	$9,564,484.00

Prosper 2013 - 2014

	ROI	Avg Rate	Loss	Principal	Count	Avg Age (Months)	Total Loss
2013	10.09%	16.24%	5.24%	$188,368,000.00	20,427	13.80	$8,206,932.50
2014	9.69%	13.11%	2.49%	$685,486,140.00	59,641	3.91	$4,289,422.00

http://www.lendingmemo.com/lending-club-prosper-default-rates/

Lending Club Analysis with Data Robot



Raw Data

Feature List Informative Features

Columns (30) Page 1 of 1

loan_amnt	funded_amnt	installment	grade	sub_grade	emp_title	emp_length	home_owne...	annual
11500	11500	350.01	A	A1	Metal Worki...	1	RENT	3500
5000	5000	168.74	B	B4	Alabs corp	1	RENT	3500
10000	10000	307.04	A	A2	Volunteers ...	1	OWN	4160
20000	20000	684.43	B	B5	BRUNSWIC...	3	MORTGAGE	9700
11175	11175	340.12	A	A1	Flanagan-C...	6	MORTGAGE	4700
17500	17500	613.53	C	C3	USPS	10	MORTGAGE	6700
13000	13000	432.54	B	B3	mussle whit...	3	MORTGAGE	1100
12000	12000	388.11	B	B1	stop & shop	10	RENT	3800
15000	15000	545.99	D	D2	Michigan St...	5	MORTGAGE	3600
13000	13000	399.15	A	A2	Nick Weitze...	3	MORTGAGE	5000

Rows (5390) Page 1 of 54

Menu Search Feature List All Features View Raw Data 1-34 of 34

Feature Name	Index	Var Type	Unique	Missing	Mean	Std Dev	Median
[Reference ID] id	1	Numeric	50,000	0	1,918,444	638,983	1,587,759
[Reference ID] member_id	2	Numeric	50,000	0	2,283,786	802,482	1,857,296
loan_amnt	3	Numeric	1,080	0	13,901	8,086	12,000
funded_amnt	4	Numeric	1,086	0	13,896	8,081	12,000
installment	5	Numeric	10,875	0	437	246	399
grade	6	Categorical	7	0			
sub_grade	7	Categorical	35	0			
emp_title	8	Text	37,228	2,833			
emp_length	9	Numeric	10	1,802	5.99	3.43	6
home_ownership	10	Categorical	5	0			
annual_inc	11	Numeric	5,100	0	71,317	67,474	60,000
purpose	12	Categorical	13	0			
title Use as Target	13	Categorical	15,531	2			

early_2012_2013_train.csv
Total features: 34, Datapoints: 50k

Workers: 000

1. Uploading Data
(7.435 sec.)

2. Reading raw data (Quick)
(12.374 sec.)

3. Exploratory Data Analysis : 34/34 Features
(2.961 sec.)

Select target to continue

What would you like to predict?

Enter your target

Start

Modeling Mode: Autopilot

Feature list: Informative Features

Optimization Metric:

Time-Aware Modeling

This dataset cannot use time-aware modeling because it does not contain any date features.

Set up time-aware modeling

Explore early_2012_2013_train.csv

Workers: 000

1. Uploading Data
(7.435 sec.)

2. Reading raw data (Quick)
(12.374 sec.)

3. Exploratory Data Analysis : 34/34 Features
(2.961 sec.)

Select target to continue

early_2012_2013_train.csv
Total features: 34, Datapoints: 50k

DataRobot Data Models Insights LC_San

Menu Search Feature List All Features View Raw Data

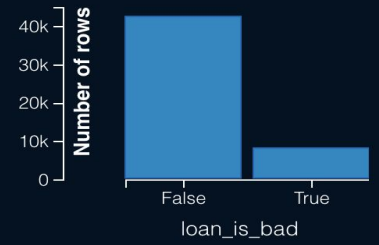
Feature Name	Index	Importance	Var Type	Unique	Missing	Mean	Std Dev	Me
loan_is_bad	34	Target	Boolean	2	0	0.16	0.36	
sub_grade	7		Categorical	35	0			
grade	6		Categorical	7	0			
annual_inc	11		Numeric	4,261	0	71,249	62,497	60,
dti	16		Numeric	3,468	0	17.38	7.77	17.
inq_last_6mths	19		Numeric	9	0	0.84	1.02	
revol_util	25		Percentage	1,011	23	0.59	0.23	0.
purpose	12		Categorical	13	0			
emp_title	8		Text	30,376	2,247			
installment	5		Numeric	9,627	0	437	246	39
funded_amnt	4		Numeric	1,046	0	13,903	8,088	12,

early_2012_2013_train.csv.zip
Total features: 34, Datapoints: 50k
Target: loan_is_bad, Metric: LogLoss

Current Feature List: All Features
Showing 1 - 34 of 34 Active Features « Previous Next » *

What would you like to predict?

loan_is_bad CLASSIFICATION



Start

Modeling Mode: Autopilot

Feature list: Informative Features

Optimization Metric: LogLoss

Time-Aware Modeling

This dataset cannot use time-aware modeling because it does not contain any date features.

Set up time-aware modeling

Show Advanced Options

Explore early_2012_2013_train.csv

Workers: 000

1. Uploading Data
(7.435 sec.)

2. Reading raw data (Quick)
(12.374 sec.)

3. Exploratory Data Analysis : 34/34 Features
(2.961 sec.)

early_2012_2013_train.csv
Total features: 34, Datapoints: 50k
Target: loan_is_bad

Menu Search Feature List All Features View Raw Data 1-34 of 34

Feature Name	Index	Var Type	Unique	Missing	Mean	Std Dev	Median	Min	N
loan_is_bad	34	Boolean	2	0					
emp_title	8	Text	37,228	0					
[Reference ID] id	1	Numeric	50,000	0	1,918,444	638,983	1,587,759	58,524	3,30
[Reference ID] member_id	2	Numeric	50,000	0	2,283,786	802,482	1,857,296	149,512	4,07
loan_amnt	3	Numeric	1,080	0	13,901	8,086	12,000	1,000	35
funded_amnt	4	Numeric	1,086	0	13,896	8,081	12,000	1,000	35
installment	5	Numeric	10,875	0	437	246	399	25.81	1,
grade	6	Categorical	7	0					
sub_grade	7	Categorical	35	0					
emp_length	9	Numeric	10	1,802	5.99	3.43	6	1	
home_ownership	10	Categorical	5	0					
annual_inc	11	Numeric	5,100	0	71,317	67,474	60,000	5,000	7,14
purpose	12	Categorical	13	0					

early_2012_2013_train.csv
Total features: 34, Datapoints: 50k
Target: loan_is_bad, Metric: LogLoss

Workers: 002

1. Setting target feature
(0.007 sec.)

2. Creating CV and Holdout partitions
100% Complete

3. Characterizing target variable

4. Loading dataset and preparing data

5. Saving target and partitioning information

6. Analyzing features

7. Calculating list of models

Menu Search Feature List All Features View Raw Data 1-34 of 34

Feature Name	Index	Importance	Var Type	Unique	Missing	Mean	Std Dev	Median
loan_is_bad	34	Target	Boolean	2	0	0.16	0.36	0
sub_grade	7		Categorical	35	0			
grade	6		Categorical	7	0			
annual_inc	11		Numeric	4,261	0	71,249	62,497	60,120
dti	16		Numeric	3,468	0	17.38	7.77	17.17
inq_last_6mths	19		Numeric	9	0	0.84	1.02	1
revol_util	25		Percentage	1,011	23	0.59	0.23	0.62
purpose	12		Categorical	13	0			
installment	5		Numeric	9,627	0	437	246	399
funded_amnt	4		Numeric	1,046	0	13,903	8,088	12,000
loan_amnt	3		Numeric	1,042	0	13,907	8,093	12,000
home_ownership	10		Categorical	5	0			
tot_cur_bal	33		Numeric	26,455	11,654	133,542	158,110	71,824

early_2012_2013_train.csv
Total features: 34, Datapoints: 50k
Target: loan_is_bad, Metric: LogLoss

Workers: 004

Processing (4)

TensorFlow Neural Network Classi...
16.00% sample , CV #1
0%
0.3 GB RAM
4.2 CPUs

Queue (29)

Logistic Regression (37)
16.00% sample , CV #1

Regularized Logistic Regression (...
16.00% sample , CV #1

Majority Class Classifier (39)
16.00% sample , CV #1

Gradient Boosted Trees Classifier...
16.00% sample , CV #1

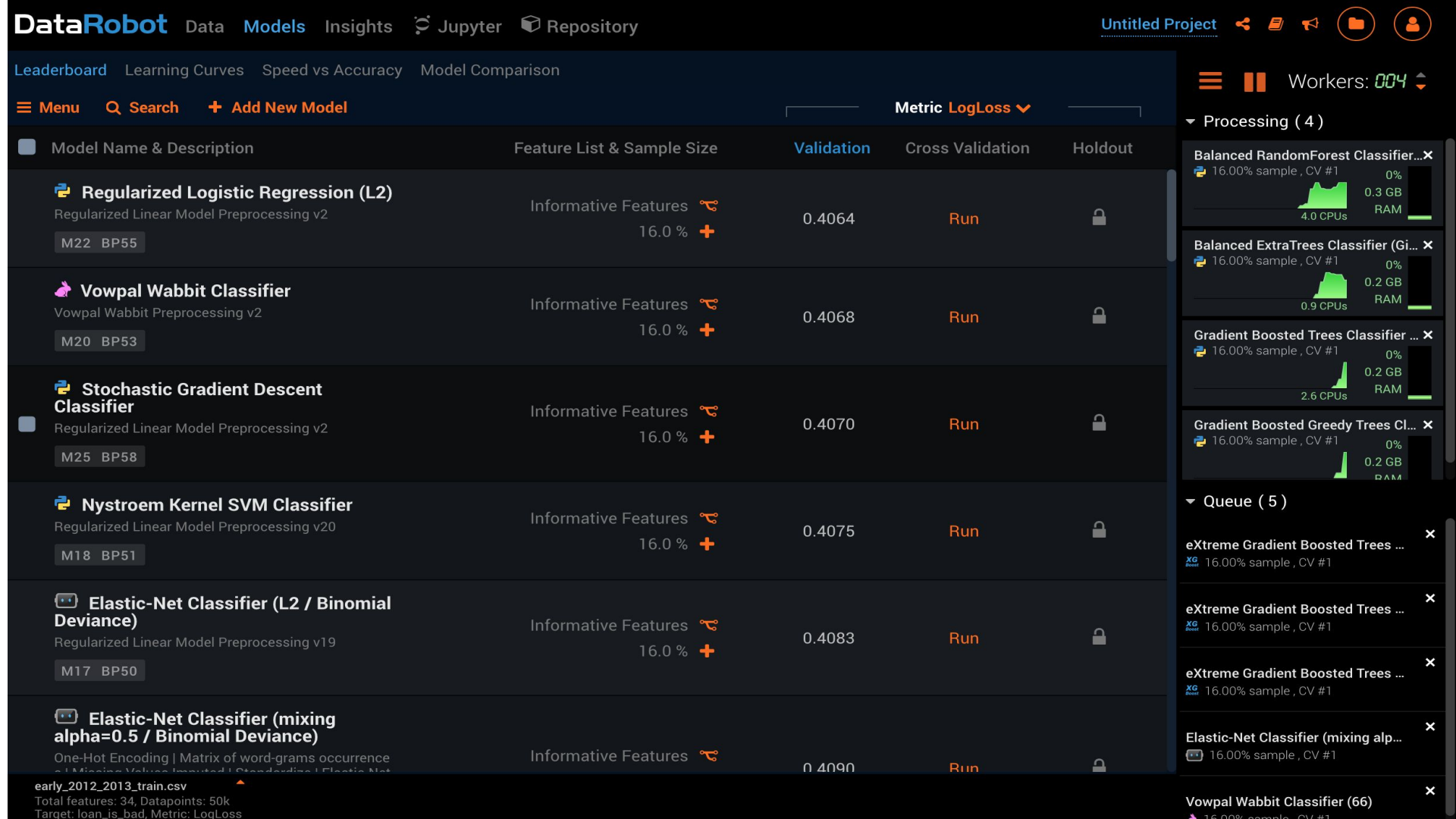
Breiman and Cutler Random Fore...
16.00% sample , CV #1

Auto-Tuned Word N-Gram Text M...
16.00% sample , CV #1

RuleFit Classifier (45)
16.00% sample , CV #1

Elastic-Net Classifier (L2 / Binomi...
16.00% sample , CV #1

Regularized Logistic Regression (...


DataRobot
Data
Models
Insights
Jupyter
Repository
Untitled Project
Leaderboard
Learning Curves
Speed vs Accuracy
Model Comparison
Menu
Search
Add New Model
Metric LogLoss
Model Name & Description
Feature List & Sample Size
Validation
Cross Validation
Holdout
Regularized Logistic Regression (L2)
Regularized Linear Model Preprocessing v2
M22 BP55
Informative Features
16.0 %
0.4064
Run
Vowpal Wabbit Classifier
Vowpal Wabbit Preprocessing v2
M20 BP53
Informative Features
16.0 %
0.4068
Run
Stochastic Gradient Descent Classifier
Regularized Linear Model Preprocessing v2
M25 BP58
Informative Features
16.0 %
0.4070
Run
Nystroem Kernel SVM Classifier
Regularized Linear Model Preprocessing v20
M18 BP51
Informative Features
16.0 %
0.4075
Run
Elastic-Net Classifier (L2 / Binomial Deviance)
Regularized Linear Model Preprocessing v19
M17 BP50
Informative Features
16.0 %
0.4083
Run
Elastic-Net Classifier (mixing alpha=0.5 / Binomial Deviance)
One-Hot Encoding | Matrix of word-grams occurrence
Informative Features
0.4090
Run
Workers: 004
Processing (4)
Balanced RandomForest Classifier...
16.00% sample , CV #1
0%
0.3 GB
RAM
4.0 CPUs
Balanced ExtraTrees Classifier (Gi...
16.00% sample , CV #1
0%
0.2 GB
RAM
0.9 CPUs
Gradient Boosted Trees Classifier ...
16.00% sample , CV #1
0%
0.2 GB
RAM
2.6 CPUs
Gradient Boosted Greedy Trees Cl...
16.00% sample , CV #1
0%
0.2 GB
Queue (5)
eXtreme Gradient Boosted Trees ...
16.00% sample , CV #1
eXtreme Gradient Boosted Trees ...
16.00% sample , CV #1
eXtreme Gradient Boosted Trees ...
16.00% sample , CV #1
Elastic-Net Classifier (mixing alp...
16.00% sample , CV #1
Vowpal Wabbit Classifier (66)
early_2012_2013_train.csv
Total features: 34, Datapoints: 50k
Target: loan_is_bad, Metric: LogLoss

Leaderboard Learning Curves Speed vs Accuracy Model Comparison

Menu Search + Add New Model Metric LogLoss

Model Name & Description	Feature List & Sample Size	Validation	Cross Validation	Holdout
ENET Blender M62 M55+51+52	Informative Features 64.0 %	0.4032	Run	
AVG Blender M59 M55+51+52	Informative Features 64.0 %	0.4032	Run	
ENET Blender M61 M57+55+51+56+52+...	Informative Features 64.0 %	0.4033	Run	
Advanced AVG Blender M60 M57+55+51+56+52+...	Informative Features 64.0 %	0.4033	Run	
Nystroem Kernel SVM Classifier Regularized Linear Model Preprocessing v20 M51 BP51	Informative Features 64.0 %	0.4033	Run	
Regularized Logistic Regression (L2) Regularized Linear Model Preprocessing v2 M52 BP55	Informative Features 64.0 %	0.4035	Run	
Elastic-Net Classifier (L2 / Binomial Deviance) Regularized Linear Model Preprocessing v19 M55 BP50	Informative Features 64.0 %	0.4039	Run	

Workers: 004

Autopilot has finished

Run Autopilot on a Different Feature List

Unlock Holdout

early_2012_2013_train.csv
Total features: 34, Datapoints: 50k
Target: loan_is_bad, Metric: LogLoss

DataRobot Data Models Insights LC_San

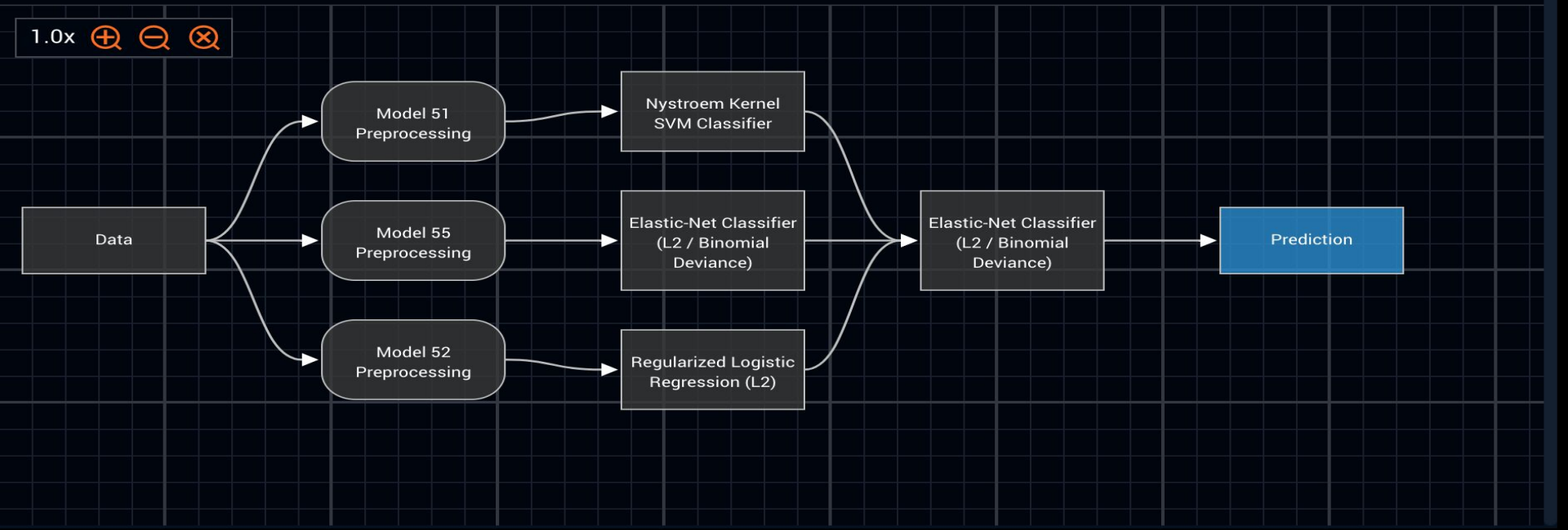
Leaderboard Learning Curves Speed vs Accuracy Model Comparison

Menu Search + Add New Model Metric LogLoss

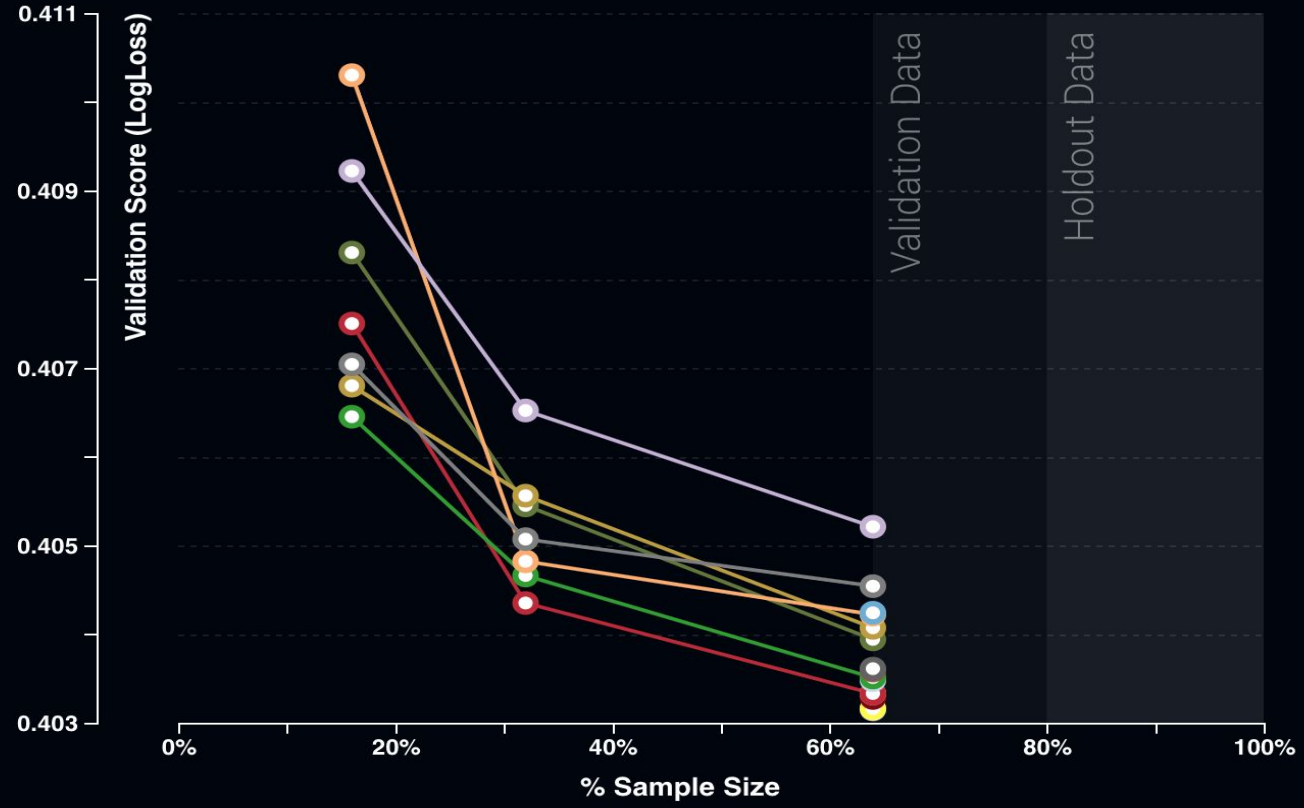
Model Name & Description

Holdout : 0.4085

Blueprint Lift Chart Model X-Ray Feature Impact Model Info Model Log ROC Curve Deploy Model Predict Reason Codes Download

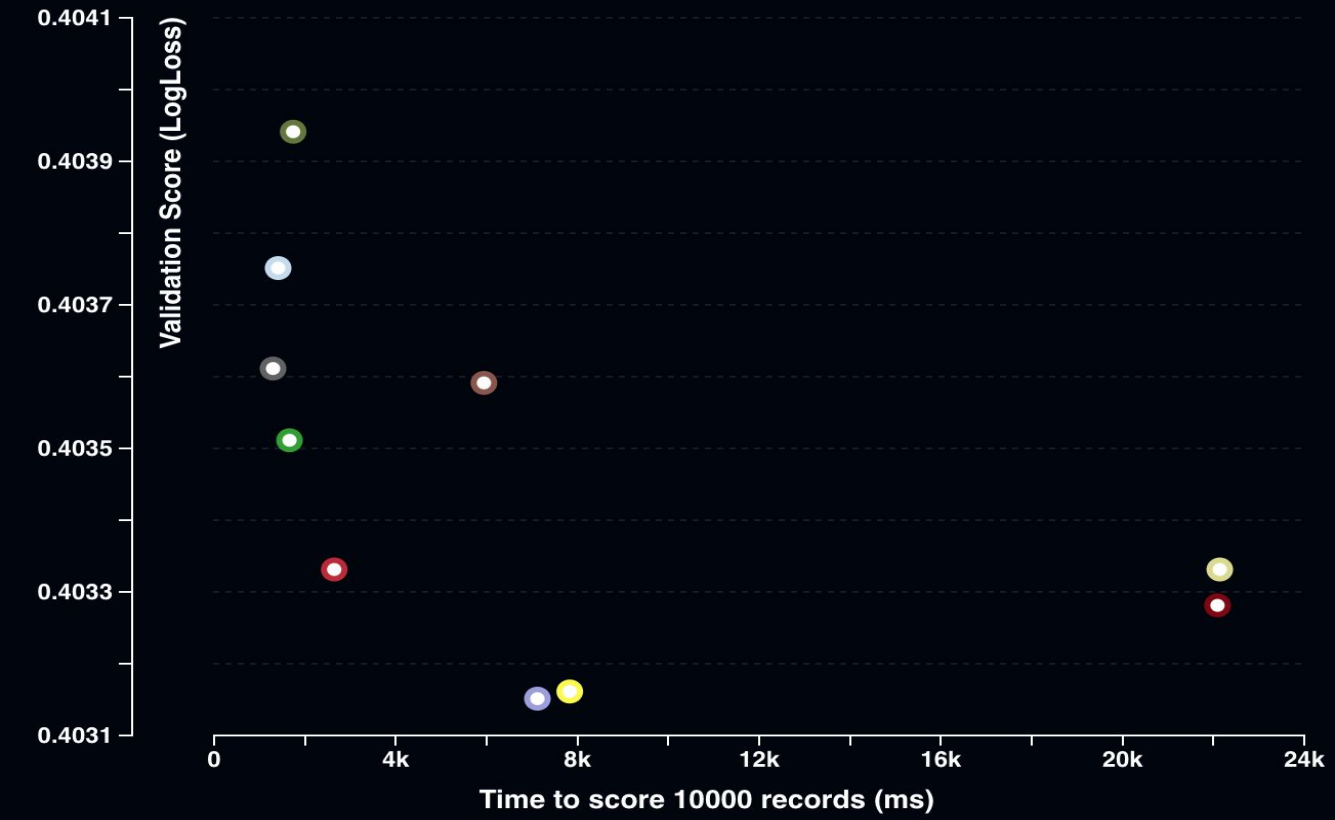


early_2012_2013_train.csv.zip
Total features: 34, Datapoints: 50k
Target: loan_is_bad, Metric: LogLoss



Feature List All Features

- ENET Blender
  M55+51+52
  Informative Features
- AVG Blender
  M55+51+52
  Informative Features
- ENET Blender
  M57+55+51+56+52+...
  Informative Features
- Advanced AVG Blender
  M57+55+51+56+52+...
  Informative Features
- Nystroem Kernel SVM Classifier
  BP44
  Informative Features
- Regularized Logistic Regression (L2)
  BP47
  M52Top25
- Regularized Logistic Regression (L2)
  BP47
  Informative Features
- MED Blender
  M55+52
  Informative Features
- Regularized Logistic Regression (L2)
  BP47
  M52Top20



Feature List All Features

- ENET Blender
  M55+51+52, M62
  64% Sample Size, Informative Features
- AVG Blender
  M55+51+52, M59
  64% Sample Size, Informative Features
- ENET Blender
  M57+55+51+56+52+..., M61
  64% Sample Size, Informative Features
- Advanced AVG Blender
  M57+55+51+56+52+..., M60
  64% Sample Size, Informative Features
- Nystroem Kernel SVM Classifier
  BP44, M51
  64% Sample Size, Informative Features
- Regularized Logistic Regression (L2)
  BP47, M52
  64% Sample Size, Informative Features
- MED Blender
  M55+52, M69
  64% Sample Size, Informative Features
- Regularized Logistic Regression (L2)
  BP47, M73
  64% Sample Size, M52Top20
- Regularized Logistic Regression (L2)
  BP47, M79
  100% Sample Size, M52Top25

ENET Blender

M55+51+52 M62

64% Sample Size, Informative Features

Change Model

LogLoss (Validation) :
0.4032

Gini Norm (Validation) :
0.3856

Prediction Time:
8368.36 ms

VS

Dual Lift Lift Roc Curve

Regularized Logistic Regression (L2)

BP38 M45

32% Sample Size, Informative Features

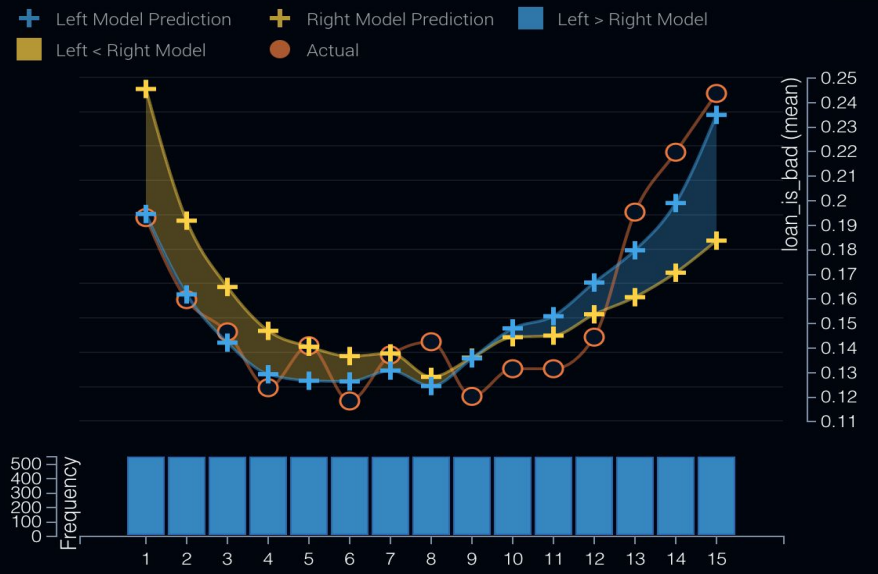
Change Model

LogLoss (Validation) :
0.4071

Gini Norm (Validation) :
0.3666

Prediction Time:
991.91 ms

Number of Bins : 15



Workers: 004

Autopilot has finished

Run Autopilot on a Different Feature List

Unlock Holdout

ENET Blender

M55+51+52 M62

64% Sample Size, Informative Features

Change Model

LogLoss (Validation) :
0.4032

LogLoss (Holdout) :
0.4085

Gini Norm (Validation) :
0.3856

Gini Norm (Holdout) :
0.3421

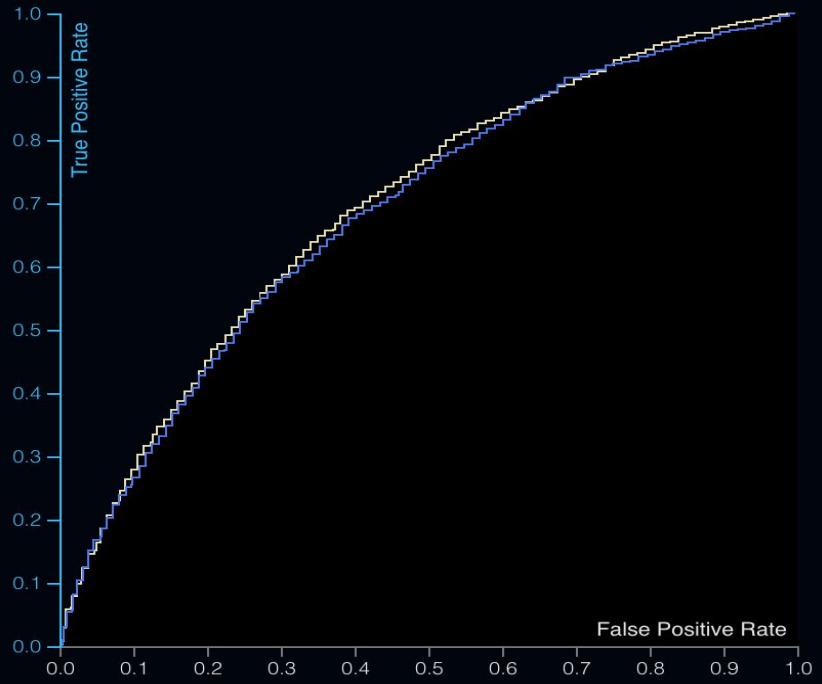
AUC (Validation) :
0.6928

Prediction Time:
7140.19 ms

VS

Dual Lift Lift Roc Curve

Roc Curve Data Source: Validation Holdout



Regularized Logistic Regression (L2)

BP34 M44

32% Sample Size, Informative Features

Change Model

LogLoss (Validation) :
0.4071

LogLoss (Holdout) :
0.4109

Gini Norm (Validation) :
0.3666

Gini Norm (Holdout) :
0.3297

AUC (Validation) :
0.6833

Prediction Time:
1055.62 ms

Model Name & Description	Feature List & Sample Size	Validation	Cross Validation	Holdout
ENET Blender M62 M55+51+52	Informative Features 64.0 %	0.4032	Run	

Autopilot has finished

Run Autopilot on a Different Feature List

Unlock Holdout

Blueprint Lift Chart Model X-Ray Feature Impact Model Info Model Log ROC Curve Coefficients Rating Tables Word Cloud Deploy Model Predict Reason Codes Download

Selection Summary

Export

Maximizes F1 Score − NOT(1) + 1

F1 Score	True Positive Rate (Sensitivity)	False Positive Rate (Fallout)	True Negative Rate (Specificity)	Positive Predictive Value (Precision)	Negative Predictive Value	Accuracy	Matthews Correlation Coefficient
0.3643	0.5588	0.2797	0.7203	0.2702	0.898	0.695	0.2166

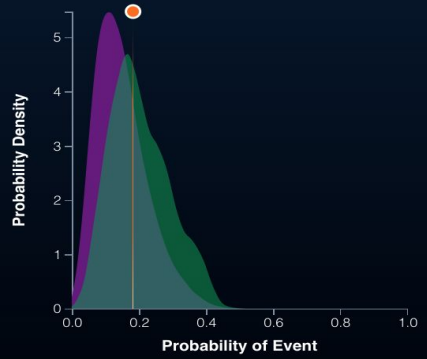
		Predicted −	Predicted +	
Actual	−	4861 (TN)	1888 (FP)	6749
	+	552 (FN)	699 (TP)	1251
		5413	2587	8000

ROC Curve

Data Source: Validation

Prediction Distribution

Threshold (0-1): 0.1814 Density



Model Name & Description	Feature List & Sample Size	Validation	Cross Validation	Holdout
AVG Blender M59 M55+51+52	Informative Features 64.0 %	0.4032	Run	

ENET Blender

M55+51+52 M62

64% Sample Size, Informative Features

Change Model

LogLoss (Validation) :
0.4032

LogLoss (Holdout) :
0.4085

Gini Norm (Validation) :
0.3856

Gini Norm (Holdout) :
0.3421

Prediction Time:
7140.19 ms

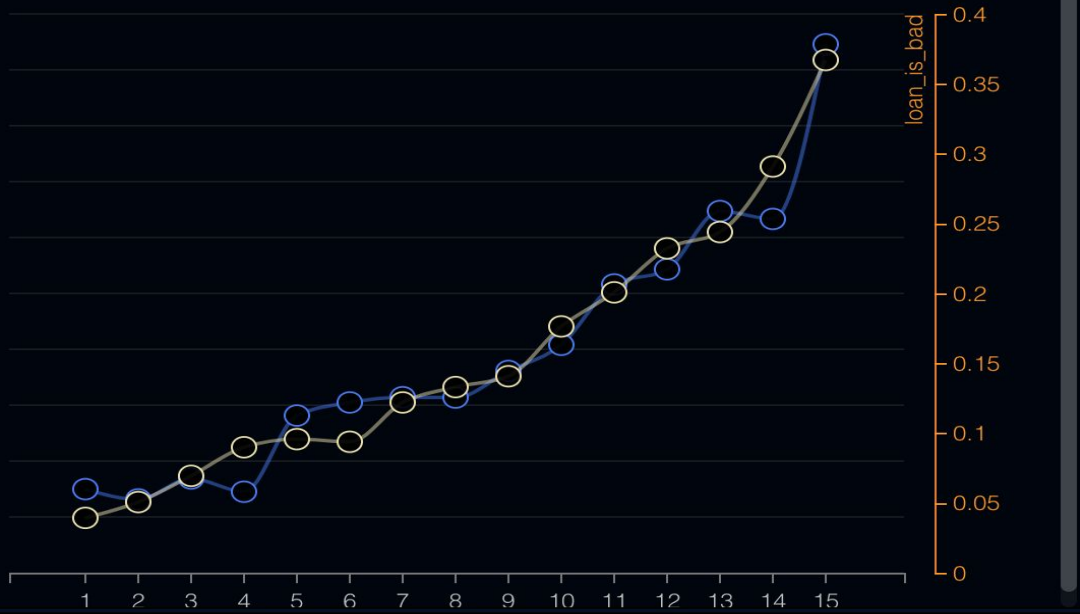
VS

Dual Lift Lift Roc Curve

Lift Data Source: Validation Holdout

Number of Bins : 15

Left Model Actual Right Model Actual



Regularized Logistic Regression (L2)

BP34 M44

32% Sample Size, Informative Features

Change Model

LogLoss (Validation) :
0.4071

LogLoss (Holdout) :
0.4109

Gini Norm (Validation) :
0.3666

Gini Norm (Holdout) :
0.3297

Prediction Time:
1055.62 ms

DataRobot Data Models Insights

Leaderboard Learning Curves Speed vs Accuracy Model Comparison

Menu Search + Add New Model Metric LogLoss

Model Name & Description

Blueprint Lift Chart Model X-Ray Feature Impact Model Info Model Log ROC Curve Deploy Model Predict Reason Codes Download

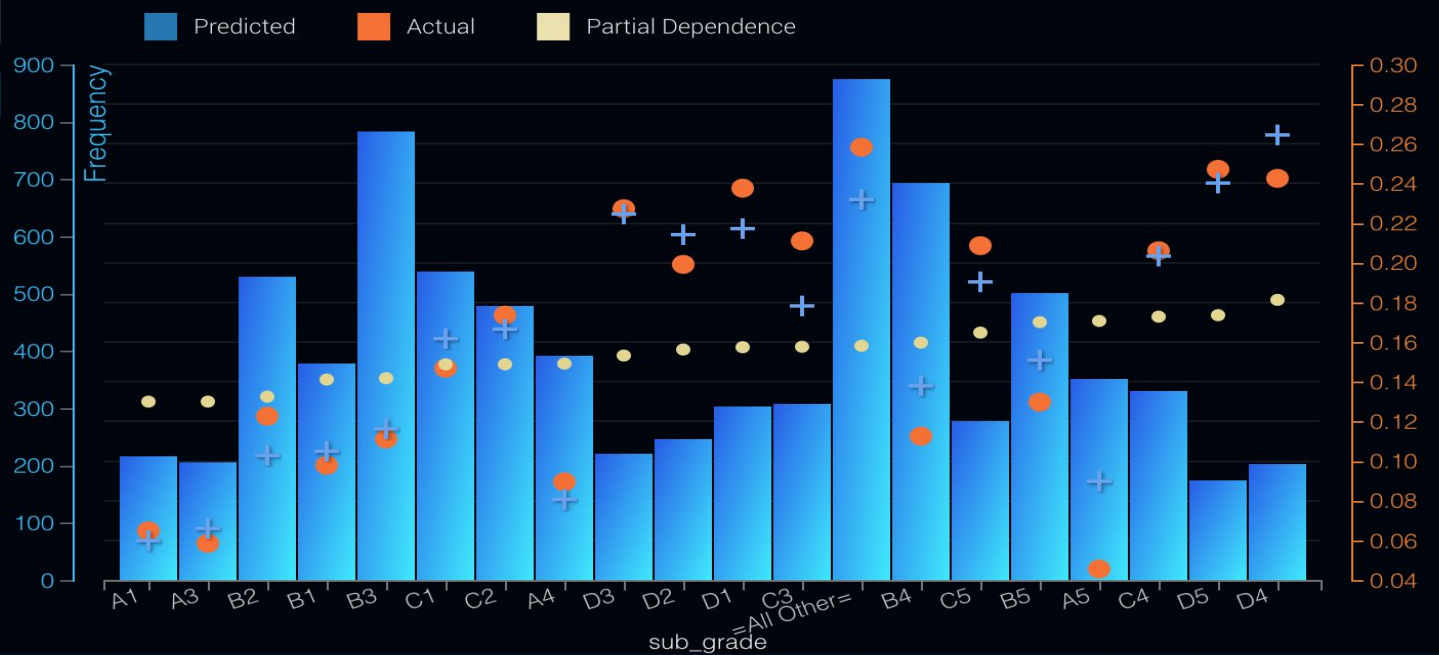
Features

Search for features

sub_grade

grade

Sort: Effect Size Export



DataRobot Data Models Insights LC_Sanjiv

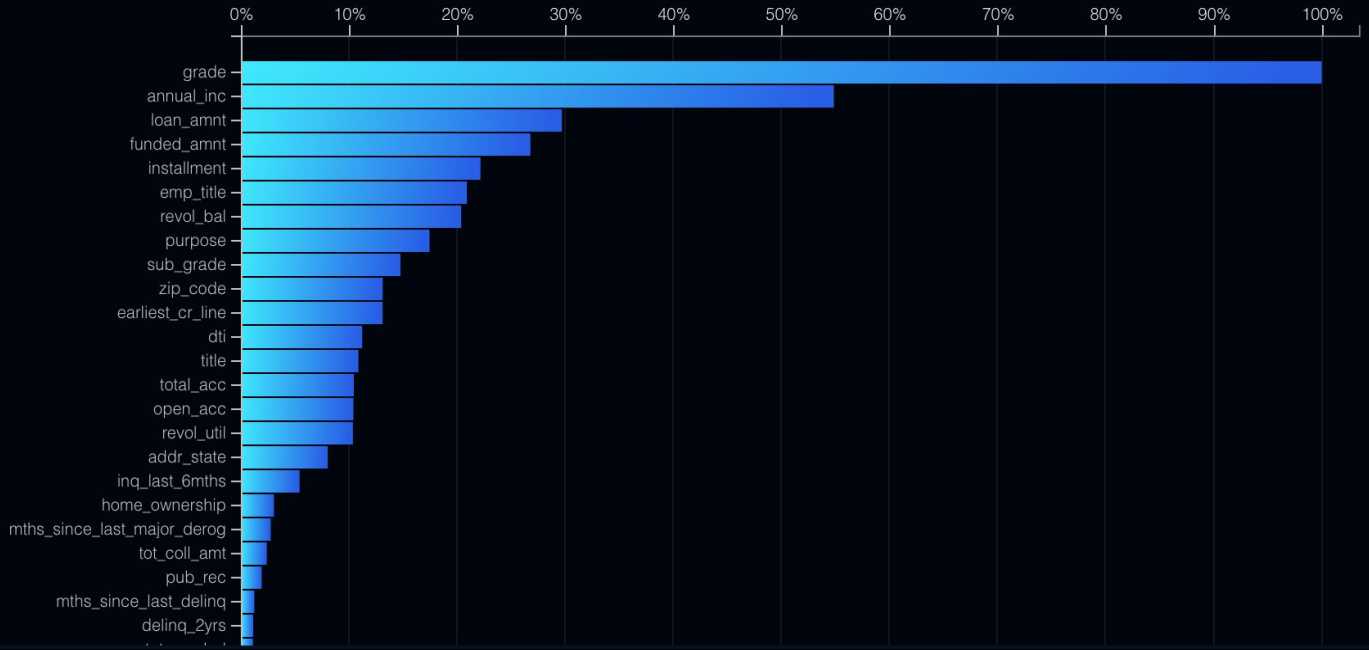
Leaderboard Learning Curves Speed vs Accuracy Model Comparison

Menu Search + Add New Model Metric LogLoss

Model Name & Description

Blueprint Lift Chart Model X-Ray Feature Impact Model Info Model Log ROC Curve Deploy Model Predict Reason Codes Download

Sort by Feature Impact Export Create new feature list with top 0 features CREATE



Workers: 004

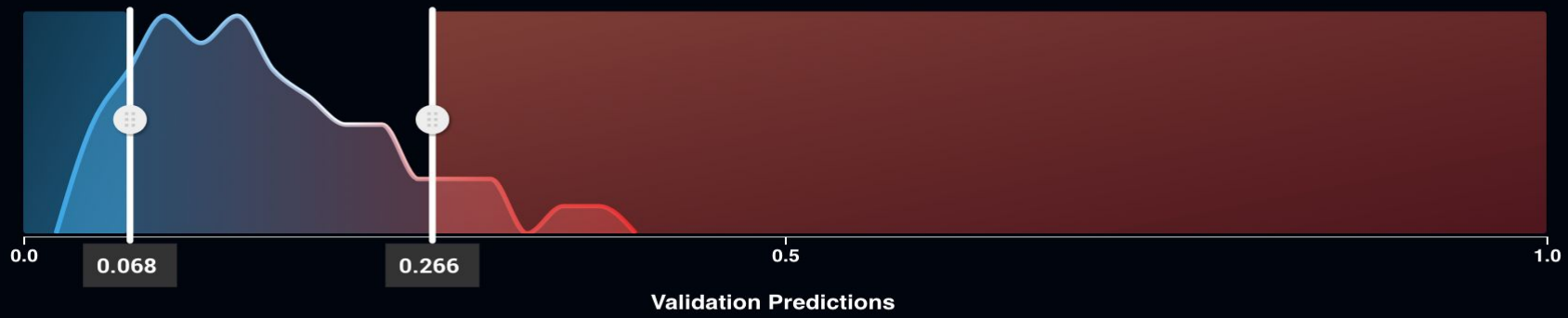
Run Autopilot on a Different Feature List

Holdout is Unlocked

Processing (1)

Model X-Ray M59 (M55+51+52)
64.00% sample
5%
3.4 GB RAM
1.0 CPUs

Get top 3 reasons for low high predictions: COMPUTE & DOWNLOAD



Preview of reasons for predictions from validation data within highlighted area(s):

ID	PREDICTION	REASONS		
40078	0.542	++ purpose = "small_business"	++ grade = "D"	++ annual_inc = 39000
21989	0.492	+++ purpose = "small_business"	++ grade = "D"	++ annual_inc = 36000
46230	0.465	+++ grade = "E"	++ annual_inc = 33763	++ inq_last_6mths = 3
16965	0.024	--- grade = "A"	-- annual_inc = 138000	-- sub_grade = "A1"
33460	0.023	--- grade = "A"	-- annual_inc = 88000	-- revol_bal = 52720
19155	0.023	--- grade = "A"	-- annual_inc = 128000	-- sub_grade = "A3"