

Automation, Robots, AI, Credit Analytics

Sanjiv R. Das

ISB | FinTech | 2018

Recap: 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

Payments & Transfers



Lending & Financing



Retail Banking



Financial Management



Insurance

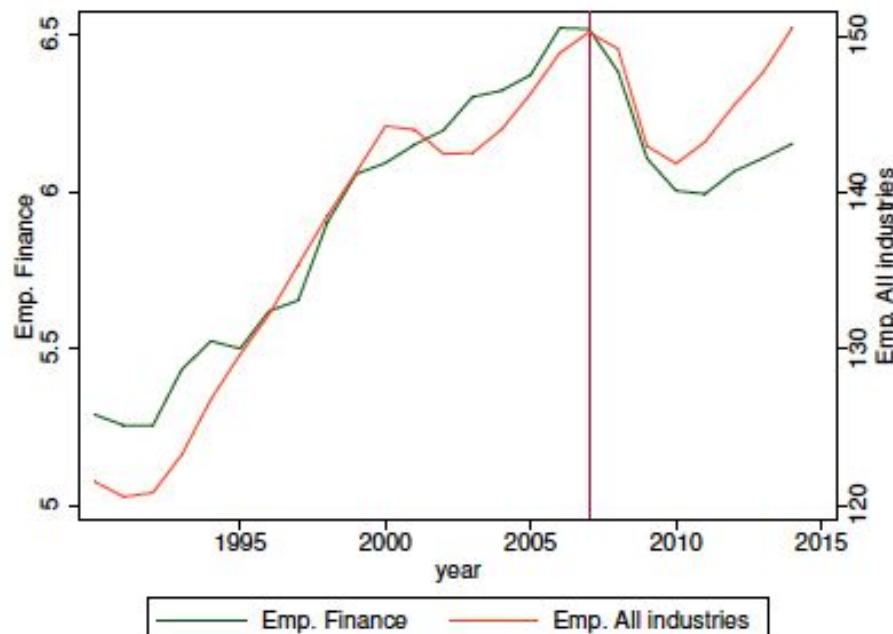
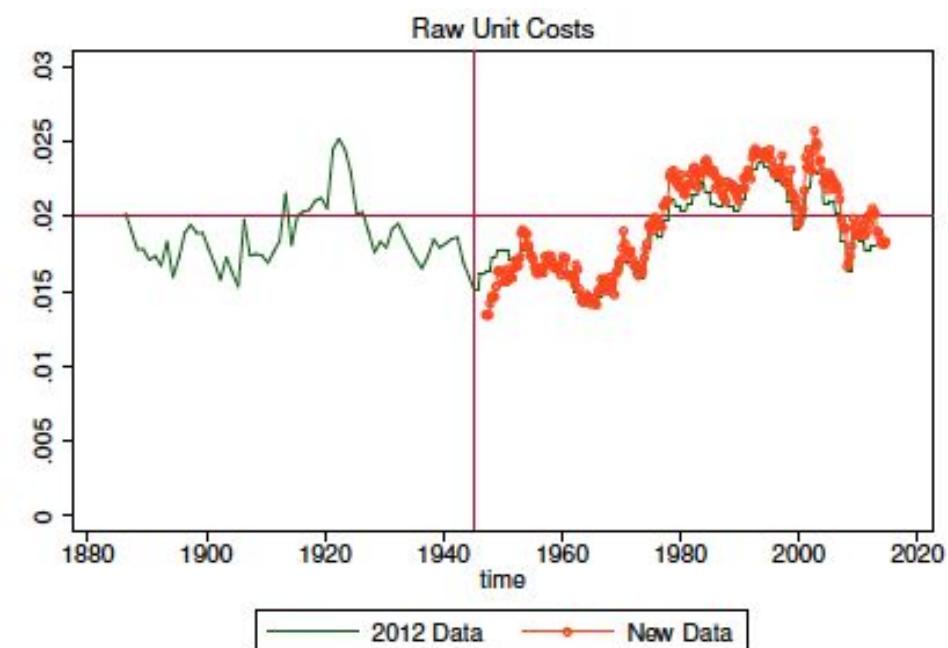


Markets & Exchanges



BI INTELLIGENCE

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.



THE AI IN FINTECH MARKET MAP

CREDIT SCORING / DIRECT LENDING

- Affirm
- AVANT
- zest finance
- argon
- ADF
- habitot
- CREAM FINANCE
- aire
- float
- naborly
- Upstart
- creditvidya
- james
- WeCASH 闪电贷

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
- WAVE
- fount
- domeyard
- Alpaca
- AIM
- ForwardLane
- TRUMID
- ALGORIZM
- bit.ai
- AIDYIA
- BINATIX

REGULATORY, COMPLIANCE, & FRAUD DETECTION

- TRIFACTA
- Digital Reasoning
- WorkFusion
- Paxata
- DataRobot
- sift science
- Onfido
- feedzai
- SKYTREE
- SOCURE
- bigstream
- trooly
- BehavioSec
- ComplyAdvantage
- cortical.io
- Dimebox
- Fraugster
- neurensic
- PredicSis
- Jewel Paymentech
- CheckRecipient
- Fortia



GENERAL PURPOSE / PREDICTIVE ANALYTICS

- OPERA
- AYASDI
- KENSIC
- NarrativeScience
- cocontext relevant
- H2O AI
- smartzip
- CognitiveScale
- Anodot
- Lucena
- Numenta

BUSINESS FINANCE & EXPENSE REPORTING

- fyle
- NetChain²
- AppZen
- ZEITGOLD.
- WAVE

INSURANCE

- CAPE
- Captricity
- TRACTABLE
- riskgenius
- CYENCE
- Shift Technology
- Zendrive
- Lemonade
- UNDERSTORY

MARKET RESEARCH / SENTIMENT ANALYSIS

- Dataminr
- alphasense
- Orbital Insight
- Descartes Labs
- iSENTIUM
- indico
- acuity
- SIGNAL
- FeedStock

DEBT COLLECTION

- TrueAccord
- collectAI

Investing in AI



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



Certified Pre-Owned
Mercedes-Benz E-Class.
Visit your local dealership.

[View Inventory](#)



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](#), [Zoloz](#), [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

kaggle

Search kaggle



Competitions

Datasets

Kernels

Discussion

Jobs

...

Sign In

Featured Dataset

701

Credit Card Fraud Detection

Anonymized credit card transactions labeled as fraudulent or genuine



Andrea • last updated a year ago

Overview

Data

Kernels

Discussion

Activity

Download (68 MB)

New Kernel

Tags

finance

crime

medium

featured

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

Credit Card Fraud Detection Analysis

(See Jupyter NB = CC_Fraud_RF.ipynb)

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

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 results 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.

Metrics: Confusion Matrix

- Given K categories, the matrix is of dimension $K \times K$. By convention, the columns relate to the category assigned by the classifier algorithm and the rows refer to the actual category in which the text resides.
- If an algorithm has no classification ability, then the rows and columns of the matrix will be independent of each other. Under this null hypothesis, the statistic that is examined for rejection is as follows:

$$\chi^2[dof = (K - 1)^2] = \sum_{i=1}^K \sum_{j=1}^K \frac{[O(i,j) - E(i,j)]^2}{E(i,j)}$$

where $O(i,j)$ are the actual numbers observed in the confusion matrix, and $E(i,j)$ are the expected numbers, assuming no classification ability under the null hypothesis.

- If $M(i)$ represents the total across row i of the confusion matrix, and $M(j)$ the column total, then

$$E(i,j) = \frac{M(i) \times M(j)}{\sum_{i=1}^K M(i)} \equiv \frac{M(i) \times M(j)}{\sum_{j=1}^K M(j)}$$

The degrees of freedom of the χ^2 statistic is $(K - 1)^2$. This statistic is very easy to implement and may be applied to models for any K . A highly significant statistic is evidence of classification ability.

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>

© 2015 INFORMS

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

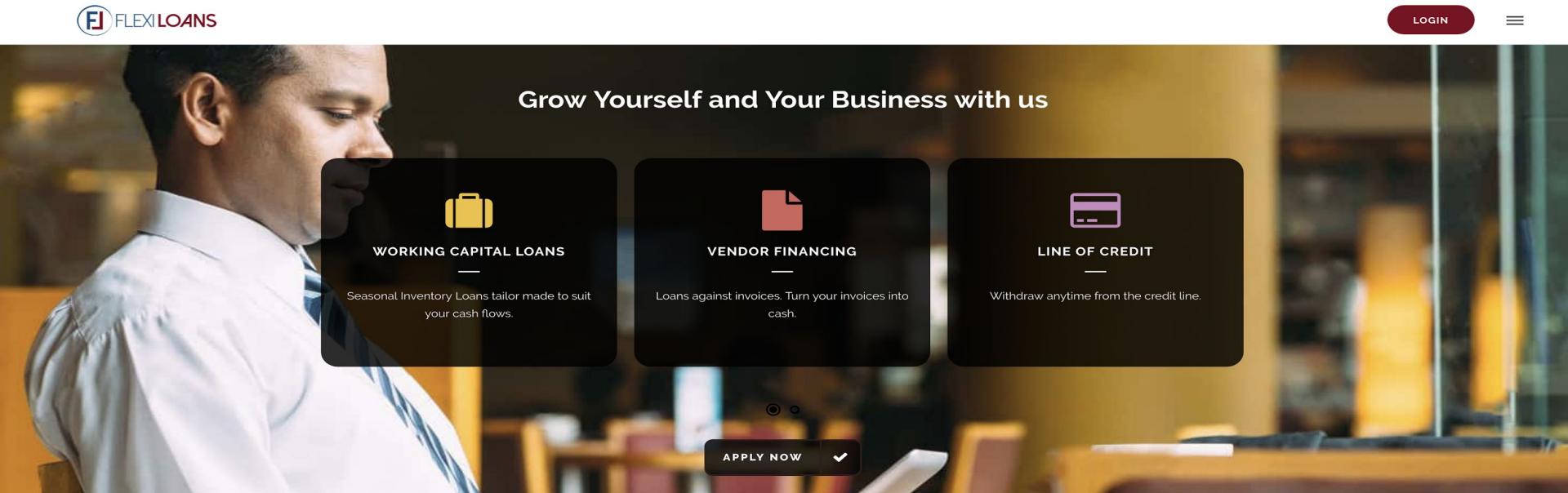
Marketing Department, The Wharton School, University of Pennsylvania, Philadelphia, Pennsylvania 19104
[{pyild@wharton.upenn.edu, vdbulte@wharton.upenn.edu}](mailto:{pyild@wharton.upenn.edu,vdbulte@wharton.upenn.edu})

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



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

Python For Data Science Cheat Sheet

Scikit-Learn

Learn Python for data science interactively at www.DataCamp.com



Scikit-learn

Scikit-learn is an open source Python library that implements a range of machine learning, preprocessing, cross-validation and visualization algorithms using a unified interface.



A Basic Example

```
>>> from sklearn import neighbors, datasets, preprocessing
>>> from sklearn.model_selection import train_test_split
>>> from sklearn.metrics import accuracy_score
>>> iris = datasets.load_iris()
>>> X, y = iris.data[:, :2], iris.target
>>> X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
>>> scaler = preprocessing.StandardScaler().fit(X_train)
>>> X_train = scaler.transform(X_train)
>>> X_test = scaler.transform(X_test)
>>> knn = neighbors.KNeighborsClassifier(n_neighbors=5)
>>> knn.fit(X_train, y_train)
>>> y_pred = knn.predict(X_test)
>>> accuracy_score(y_test, y_pred)
```

Loading The Data

Also see NumPy & Pandas

Your data needs to be numeric and stored as NumPy arrays or SciPy sparse matrices. Other types that are convertible to numeric arrays, such as Pandas DataFrame, are also acceptable.

```
>>> import numpy as np
>>> X = np.random.random((10,5))
>>> y = np.array(['M','M','F','F','M','F','M','F','F','F'])
>>> X[X < 0.7] = 0
```

Training And Test Data

```
>>> from sklearn.model_selection import train_test_split
>>> X_train, X_test, y_train, y_test = train_test_split(X,
...                                                    y,
...                                                    random_state=0)
```

Preprocessing The Data

Standardization

```
>>> from sklearn.preprocessing import StandardScaler
>>> scaler = StandardScaler().fit(X_train)
>>> standardized_X_train = scaler.transform(X_train)
>>> standardized_X_test = scaler.transform(X_test)
```

Normalization

```
>>> from sklearn.preprocessing import Normalizer
>>> scaler = Normalizer().fit(X_train)
>>> normalized_X_train = scaler.transform(X_train)
>>> normalized_X_test = scaler.transform(X_test)
```

Binarization

```
>>> from sklearn.preprocessing import Binarizer
>>> binarizer = Binarizer(threshold=0.0).fit(X)
>>> binary_X = binarizer.transform(X)
```

Create Your Model

Supervised Learning Estimators

Linear Regression

```
>>> from sklearn.linear_model import LinearRegression
>>> lr = LinearRegression(normalize=True)
```

Support Vector Machines (SVM)

```
>>> from sklearn.svm import SVC
>>> svc = SVC(kernel='linear')
```

Naïve Bayes

```
>>> from sklearn.naive_bayes import GaussianNB
>>> gnb = GaussianNB()
```

KNN

```
>>> from sklearn import neighbors
>>> knn = neighbors.KNeighborsClassifier(n_neighbors=5)
```

Unsupervised Learning Estimators

Principal Component Analysis (PCA)

```
>>> from sklearn.decomposition import PCA
>>> pca = PCA(n_components=0.95)
```

K Means

```
>>> from sklearn.cluster import KMeans
>>> k_means = KMeans(n_clusters=3, random_state=0)
```

Model Fitting

Supervised learning

```
>>> lr.fit(X, y)
>>> knn.fit(X_train, y_train)
>>> svc.fit(X_train, y_train)
```

Unsupervised Learning

```
>>> k_means.fit(X_train)
>>> pca_model = pca.fit_transform(X_train)
```

Fit the model to the data

Fit the model to the data
Fit to data, then transform it

Prediction

Supervised Estimators

```
>>> y_pred = svc.predict(np.random.random((2,5)))
>>> y_pred = lr.predict(X_test)
>>> y_pred = knn.predict_proba(X_test)
```

Unsupervised Estimators

```
>>> y_pred = k_means.predict(X_test)
```

Predict labels
Predict labels
Estimate probability of a label
Predict labels in clustering algos

Encoding Categorical Features

```
>>> from sklearn.preprocessing import LabelEncoder
>>> enc = LabelEncoder()
>>> y = enc.fit_transform(y)
```

Imputing Missing Values

```
>>> from sklearn.preprocessing import Imputer
>>> imp = Imputer(missing_values=0, strategy='mean', axis=0)
>>> imp.fit_transform(X_train)
```

Generating Polynomial Features

```
>>> from sklearn.preprocessing import PolynomialFeatures
>>> poly = PolynomialFeatures(5)
>>> poly.fit_transform(X)
```

Evaluate Your Model's Performance

Classification Metrics

Accuracy Score

```
>>> knn.score(X_test, y_test)
>>> from sklearn.metrics import accuracy_score
>>> accuracy_score(y_test, y_pred)
```

Estimator score method
Metric scoring functions

Classification Report

```
>>> from sklearn.metrics import classification_report
>>> print(classification_report(y_test, y_pred))
```

Precision, recall, f1-score
and support

Confusion Matrix

```
>>> from sklearn.metrics import confusion_matrix
>>> print(confusion_matrix(y_test, y_pred))
```

Regression Metrics

Mean Absolute Error

```
>>> from sklearn.metrics import mean_absolute_error
>>> y_true = [13, -0.5, 2]
>>> mean_absolute_error(y_true, y_pred)
```

Mean Squared Error

```
>>> from sklearn.metrics import mean_squared_error
>>> mean_squared_error(y_test, y_pred)
```

R² Score

```
>>> from sklearn.metrics import r2_score
```

Clustering Metrics

Adjusted Rand Index

```
>>> from sklearn.metrics import adjusted_rand_score
>>> adjusted_rand_score(y_true, y_pred)
```

Homogeneity

```
>>> from sklearn.metrics import homogeneity_score
>>> homogeneity_score(y_true, y_pred)
```

V-measure

```
>>> from sklearn.metrics import v_measure_score
>>> metrics.v_measure_score(y_true, y_pred)
```

Cross-Validation

```
>>> from sklearn.cross_validation import cross_val_score
>>> print(cross_val_score(knn, X_train, y_train, cv=4))
>>> print(cross_val_score(lr, X, y, cv=2))
```

Tune Your Model

Grid Search

```
>>> from sklearn.grid_search import GridSearchCV
>>> params = {"n_neighbors": np.arange(1,3),
...            "metric": ["euclidean", "cityblock"]}
>>> grid = GridSearchCV(knn, params)
>>> grid.fit(X_train, y_train)
>>> print(grid.best_score_)
>>> print(grid.best_estimator_.n_neighbors)
```

Randomized Parameter Optimization

```
>>> from sklearn.grid_search import RandomizedSearchCV
>>> params = {"n_neighbors": np.arange(1,5),
...            "weights": ["uniform", "distance"]}
>>> rsearch = RandomizedSearchCV(knn, params,
...                                param_distributions=params,
...                                n_iter=8,
...                                random_state=5)
>>> rsearch.fit(X_train, y_train)
>>> print(rsearch.best_score_)
```



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

DataRobot Data Models Insights Jupyter Repository Untitled Project Workers: 000

☰ Menu Q Search Feature List All Features View Raw Data 1-31 of 31 < >

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	31	Numeric	2	0	1.73e-3	0.04	0	0	1

creditcard.csv Total features: 31, Datapoints: 284,811 Target: Class

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

- 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.)

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



CV Folds Holdout



Holdout percentage:

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

20



Explore creditcard.csv

☰ Menu Q. 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	1.22	-0.27	-26.16	73.30

✗ 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



0%

50%

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 Q Search Feature List Informative Features ▾ View Raw Data

< 1-31 of 31 >

What would you like to predict?

Class

CLASSIFICATION



Start



Time-Aware Modeling

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

Set up time-aware modeling

Modeling Mode: Autopilot ▾

Feature list: Informative Features

Optimization Metric: LogLoss

☰ Menu Q Search Feature List All Features ▾ View Raw Data

Create new feature list

CREATE

◀ 1-31 of 31 ▶

Workers: 004 ▾

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

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

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	<div style="width: 100%;"> </div>	Numeric	221,418	0	5.58e-4	0.84	-0.07	-23.24	9.25				
V14	15	<div style="width: 100%;"> </div>	Numeric	221,418	0	-2.54e-4	0.96	0.05	-18.82	10.53				
V12	13	<div style="width: 100%;"> </div>	Numeric	221,418	0	2.47e-4	1.00	0.14	-18.68	7.85				
V10	11	<div style="width: 100%;"> </div>	Numeric	221,418	0	-4.42e-4	1.09	-0.09	-24.40	23.75				
V21	22	<div style="width: 100%;"> </div>	Numeric	221,418	0	-4.55e-4	0.73	-0.03	-34.83	27.20				
V16	17	<div style="width: 100%;"> </div>	Numeric	221,418	0	-2.35e-4	0.88	0.07	-13.30	17.32				
V27	28	<div style="width: 100%;"> </div>	Numeric	221,418	0	-4.63e-6	0.40	1.36e-3	-22.57	31.61				
V28	29	<div style="width: 100%;"> </div>	Numeric	221,418	0	-7.92e-4	0.33	0.01	-15.43	33.85				
V6	7	<div style="width: 100%;"> </div>	Numeric	221,418	0	1.12e-3	1.33	-0.27	-26.16	73.30				
V20	21	<div style="width: 100%;"> </div>	Numeric	221,418	0	-1.07e-5	0.76	-0.06	-54.50	39.42				
V3	4	<div style="width: 100%;"> </div>	Numeric	221,418	0	2.86e-3	1.51	0.18	-48.33	9.38				
V23	24	<div style="width: 100%;"> </div>	Numeric	221,418	0	1.45e-4	0.62	-0.01	-44.81	22.53				
Amount	30	<div style="width: 100%;"> </div>	Numeric	29,390	0	88.26	249	22	0	25,691				

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

▼ Processing (4)

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

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	

▼ 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 0%

▼ 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... 32.00% sample, CV #1

☰ 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

Run



TK DM Light Gradient Boosted Trees Classifier with Early Stopping

Tree-based Algorithm Preprocessing v1

M44 BP37

Informative Features

64.0 %

0.0023

Run



XG Boost 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



XG Boost 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



TK DM Light Gradient Boosted Trees Classifier with Early Stopping

Tree-based Algorithm Preprocessing v1

M28 BP37

Informative Features

32.0 %

0.0024

Run



Nystrom Kernel SVM Classifier

Regularized Linear Model Preprocessing v20

Informative Features

64.0 %

0.0024

Run

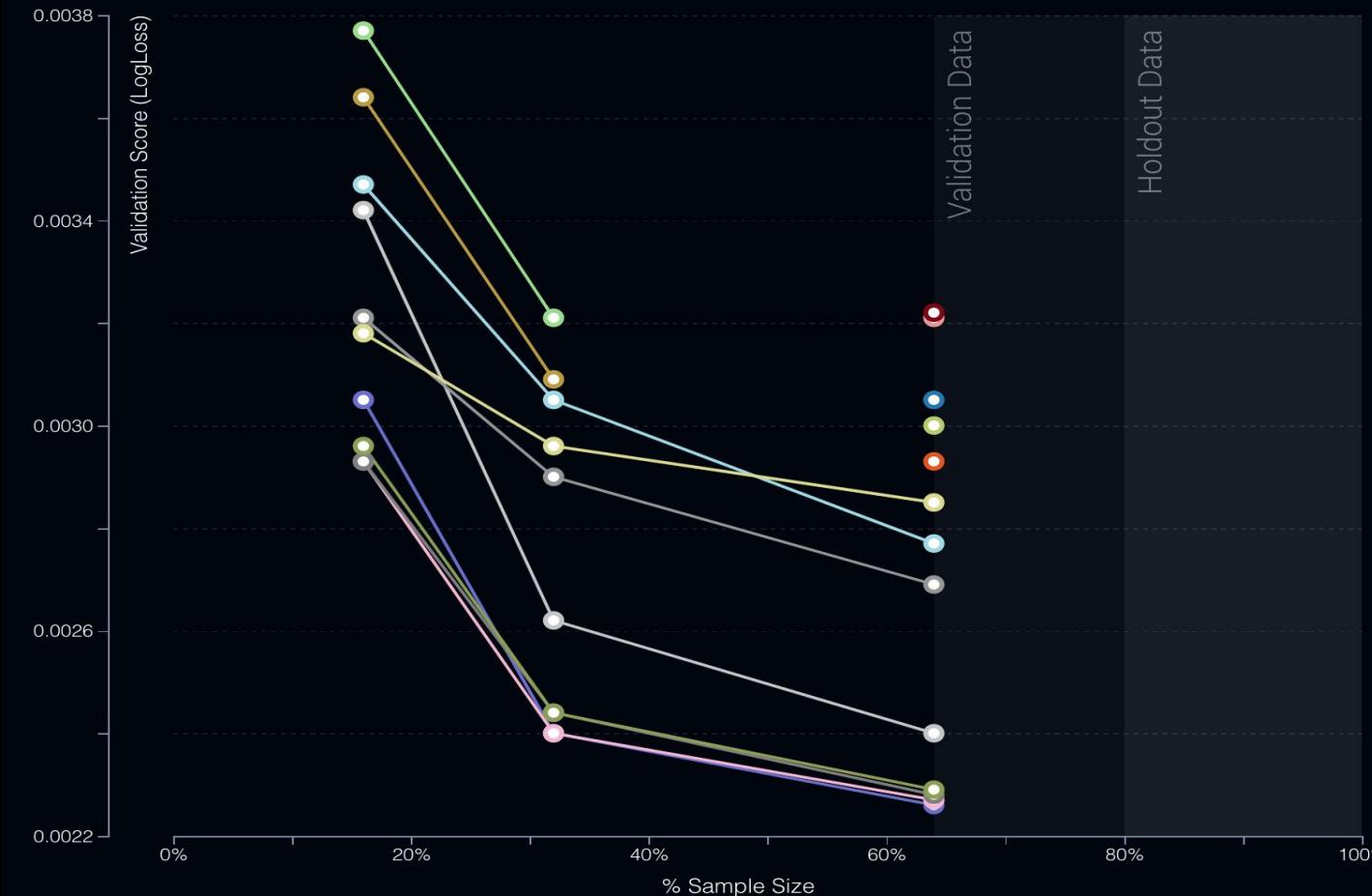


Autopilot has finished

Run Autopilot on a Different Feature List

Unlock Holdout

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

[Advanced AVG Blender](#)

☰ Menu Q 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	In Progress	
DM TK Light Gradient Boosted Trees Classifier with Early Stopping Tree-based Algorithm Preprocessing v1 M44 BP37	Informative Features 64.0 %	0.0023	Run	
XG Boost 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	
XG Boost 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	
DM TK 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	

Run Autopilot on a Different Feature List

Unlock Holdout

▼ Processing (4)



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

+ Left Model Prediction + Right Model Prediction □ Left > Right Model
█ Left < Right Model ● Actual



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

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

BP51 M45
64% Sample Size, Informative Features

VS

Dual Lift Lift Roc Curve

RandomForest Classifier (Gini)

BP27 M49
64% Sample Size, Informative Features

Change Model ↗

Change Model ↗

Lift Data Source: Validation

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

● Left Model Actual ● Right Model Actual

Number of Bins : **15**

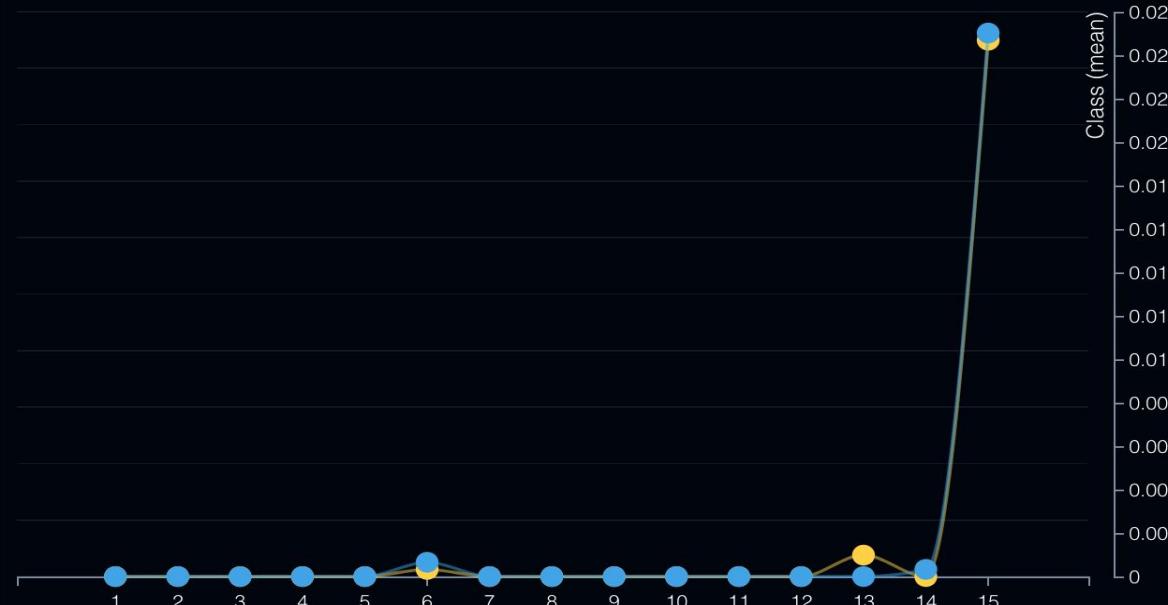
LogLoss (Validation) :
2.6900e-3

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

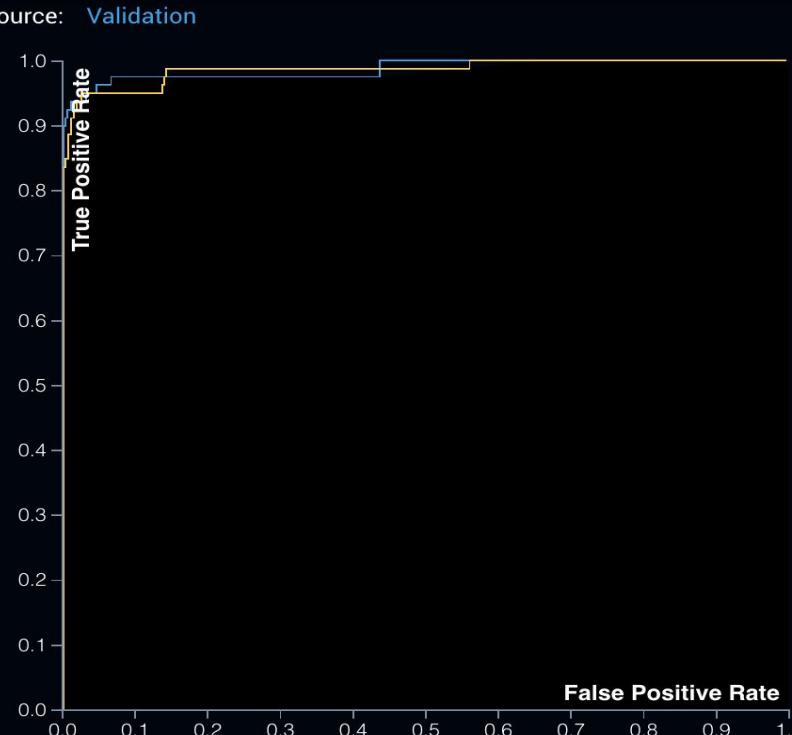
Change Model LogLoss (Validation) :
2.2600e-3LogLoss (Cross Validation) :
2.5560e-3Gini Norm (Validation) :
0.9642Gini Norm (Cross Validation) :
0.9678AUC (Validation) :
0.9821Prediction Time:
4984.05 ms**VS**

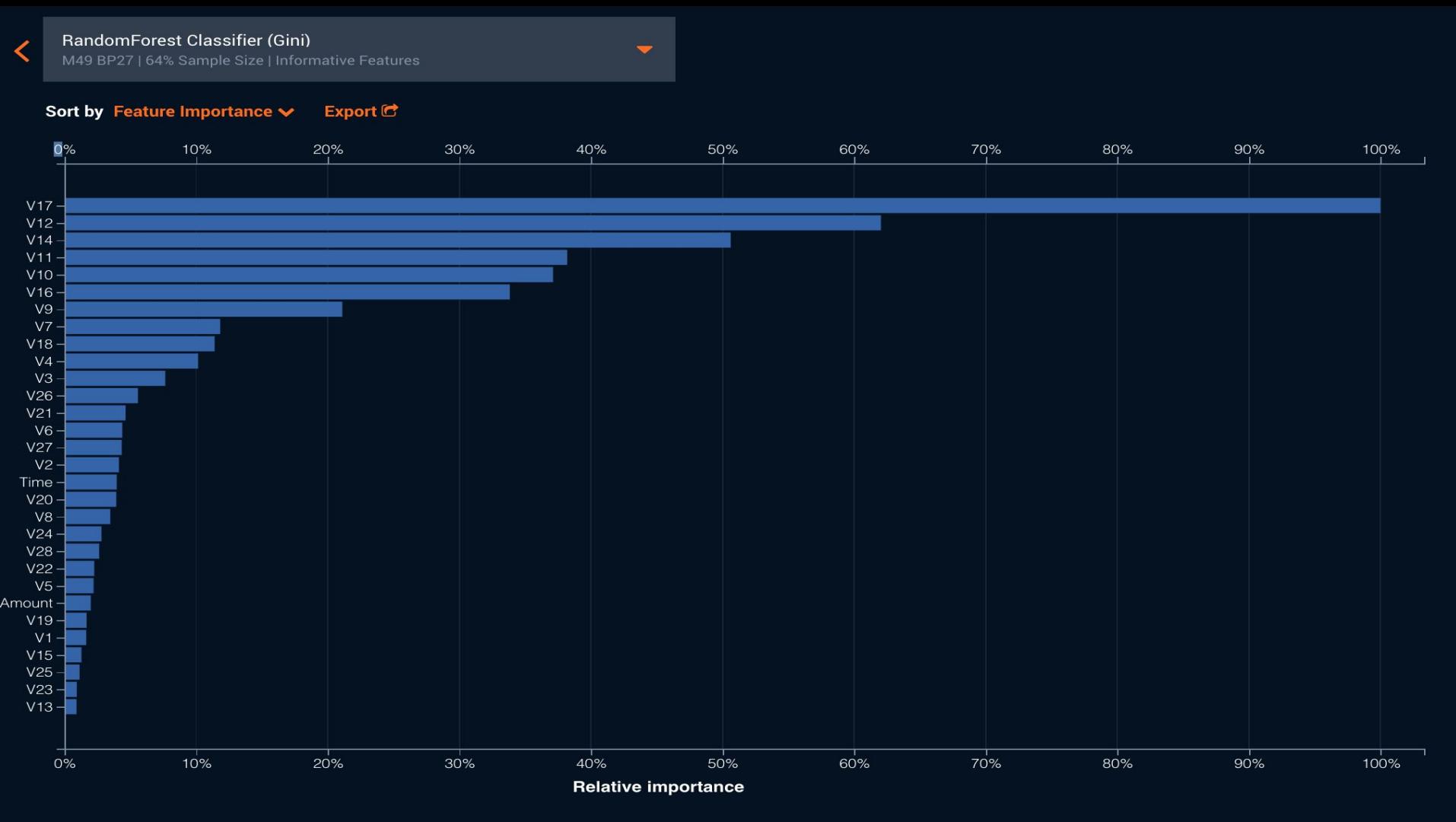
Dual Lift Lift Roc Curve

RandomForest Classifier (Gini)

BP27 M49

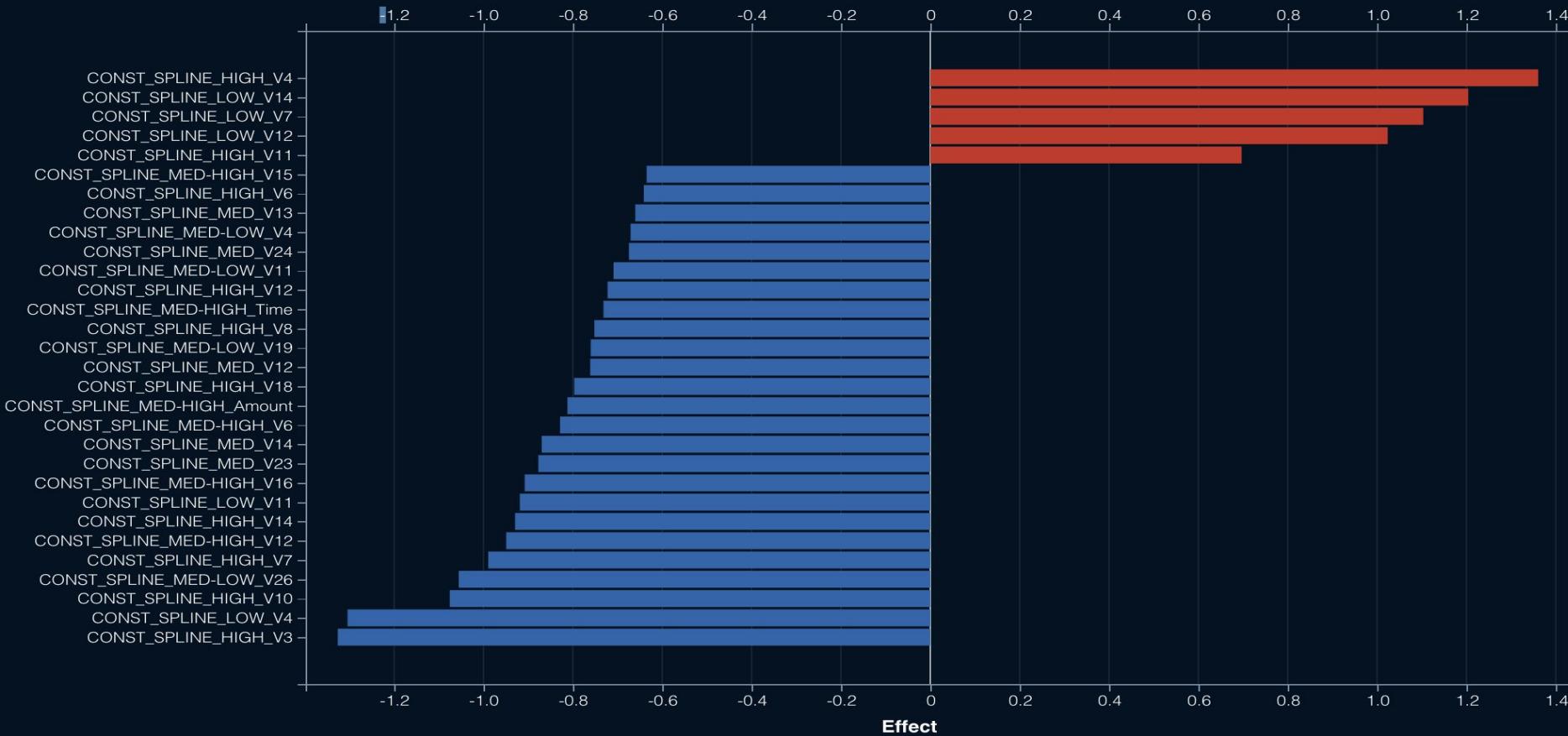
64% Sample Size, Informative Features

Change Model LogLoss (Validation) :
2.6900e-3LogLoss (Cross Validation) :
NAGini Norm (Validation) :
0.9707Gini Norm (Cross Validation) :
NAAUC (Validation) :
0.9854Prediction Time:
4514.50 ms





Sort by Feature Effect ▾ Export ↗



Model Name & Description

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

Feature List & Sample Size

Informative Features 📈

64.0 % +

Validation

0.0023

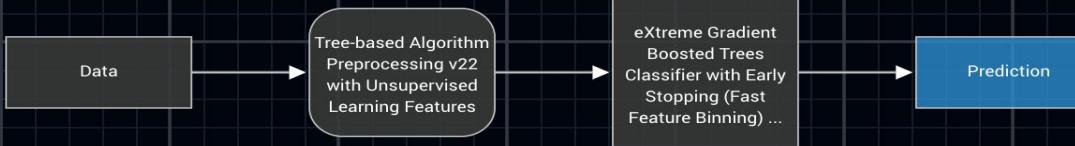
Cross Validation

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



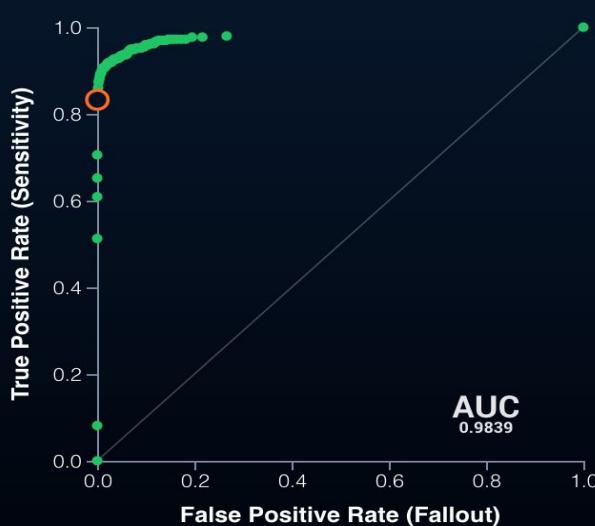
Selection Summary [Export](#)

Maximizes F1 Score - NOT(1) + 1

	True Positive Rate (Sensitivity)	False Positive Rate (Fallout)	True Negative Rate (Specificity)	Positive Predictive Value (Precision)	Negative Predictive Value	Accuracy	Matthews Correlation Coefficient	Predicted		227451		
								Actual	-			
F1 Score	0.821	0.8325	0.0003	0.9997	0.8099	0.9997	0.9994	0.8208	+	66 (FN)	328 (TP)	394
										227440	405	227845

ROC Curve

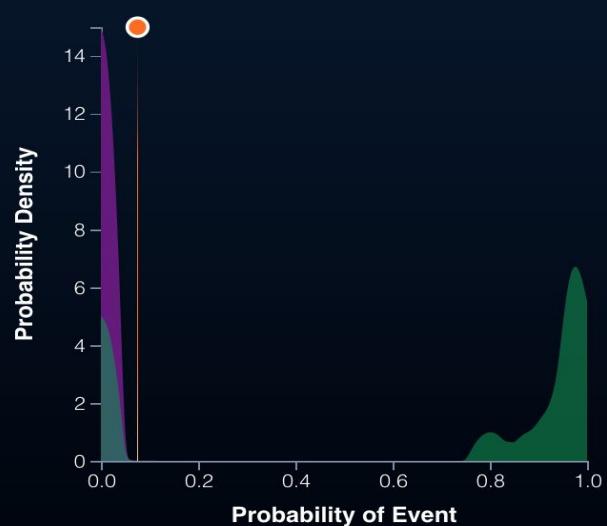
Data Source: Cross Validation



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	Abs Dev	Mean 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

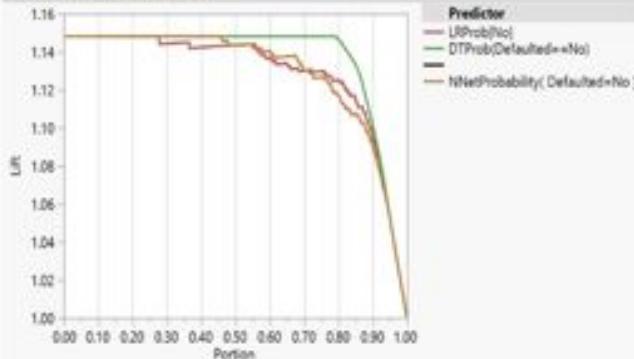
Receiver Operating Characteristics (ROC)

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

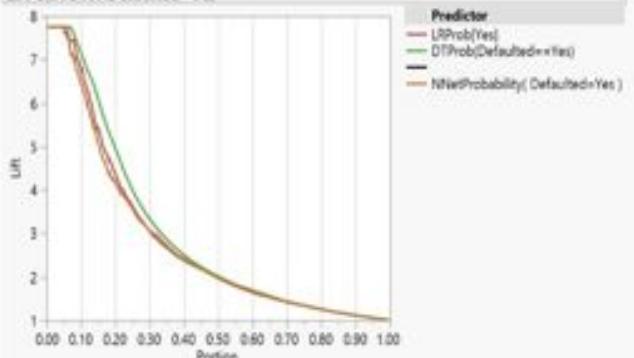
Confusion Matrix

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

Lift Curve for Defaulted=No



Lift Curve for Defaulted=Yes



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

Lending Club Analysis with Data Robot

Raw Data X

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 ▶

Workers: 000 ▾

Feature Name	Index	Var Type	Unique	Missing	Mean	Std Dev	Median	1-34 of 34
[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	13	Categorical	15,531	2				

early_2012_2013_train.csv ▾

Total features: 34, Datapoints: 50k ➤

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

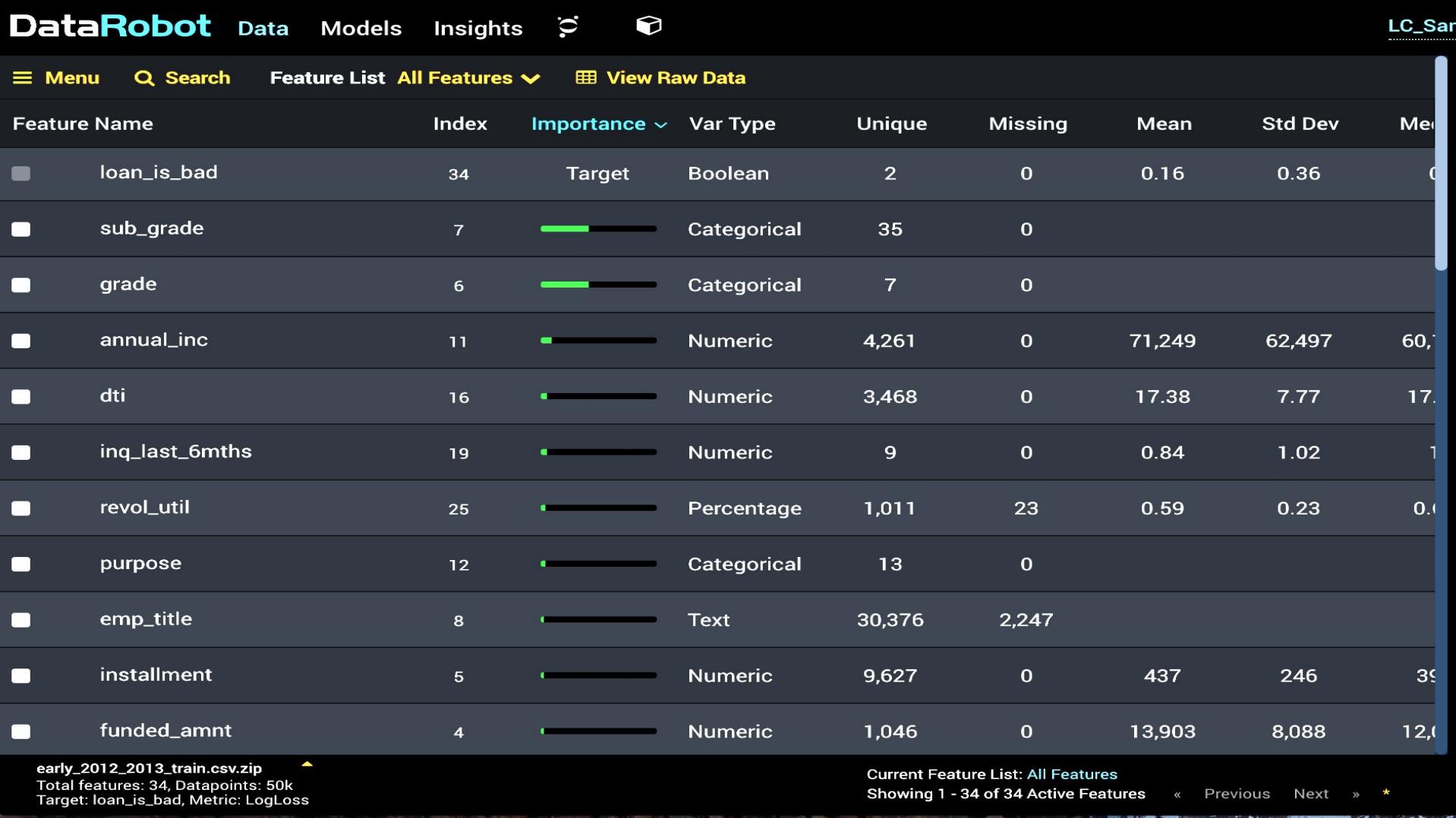
Workers: 000 ▲

1. Uploading Data
(7.435 sec.)

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

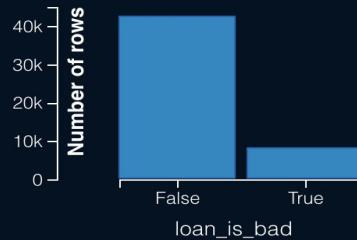
3. Exploratory Data Analysis:
34/34 Features
(2.961 sec.)

⌚ Select target to continue



What would you like to predict?

loan_is_bad CLASSIFICATION



Modeling Mode: Autopilot ▾

Feature list: **Informative Features**

Optimization Metric: **LogLoss**

Show Advanced Options

Explore early_2012_2013_train.csv

1. Uploading Data
(7.435 sec.)

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

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



Time-Aware Modeling

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

Set up time-aware modeling

Feature Name	Index	Var Type	Unique	Missing	Mean	Std Dev	Median	Min	Max
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	3,300
[Reference ID] member_id	2	Numeric	50,000	0	2,283,786	802,482	1,857,296	149,512	4,070
loan_amnt	3	Numeric	1,080	0	13,901	8,086	12,000	1,000	35,000
funded_amnt	4	Numeric	1,086	0	13,896	8,081	12,000	1,000	35,000
installment	5	Numeric	10,875	0	437	246	399	25.81	1,000
grade	6	Categorical	7	0					
sub_grade	7	Categorical	35	0					
emp_length	9	Numeric	10	1,802	5.99	3.43	6	1	30
home_ownership	10	Categorical	5	0					
annual_inc	11	Numeric	5,100	0	71,317	67,474	60,000	5,000	714,000
purpose	12	Categorical	13	0					

Workers: 002 ▾

1. Setting target feature
(0.007 sec.)2. Creating CV and Holdout partitions
100% Complete

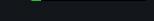
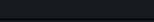
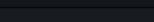
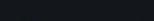
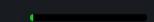
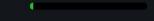
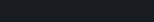
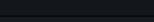
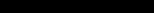
3. Characterizing target variable

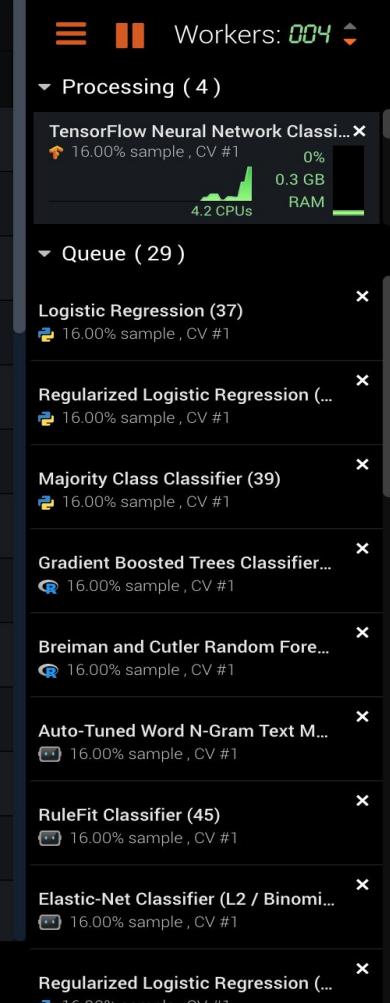
4. Loading dataset and preparing data

5. Saving target and partitioning information

6. Analyzing features

7. Calculating list of models

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



Leaderboard Learning Curves Speed vs Accuracy Model Comparison

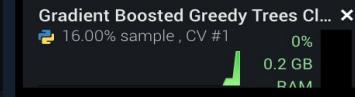
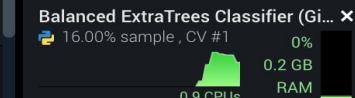
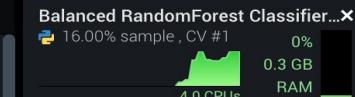
☰ Menu Q Search + Add New Model

Metric LogLoss ▾

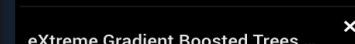
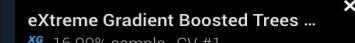
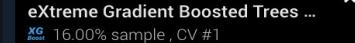
Model Name & Description	Feature List & Sample Size	Validation	Cross Validation	Holdout
Regularized Logistic Regression (L2) Regularized Linear Model Preprocessing v2 M22 BP55	Informative Features 16.0 %	0.4064	Run	
Vowpal Wabbit Classifier Vowpal Wabbit Preprocessing v2 M20 BP53	Informative Features 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 + Missing Values Imputed Standardized Elastic Net early_2012_2013_train.csv	Informative Features 0.4090	0.4090	Run	

☰ || Workers: 004 ▾

Processing (4)



Queue (5)



Vowpal Wabbit Classifier (66) ▾
16.00% sample , CV #1

Leaderboard Learning Curves Speed vs Accuracy Model Comparison

☰ Menu Q 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	
Nystrom 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 RP50	Informative Features 64.0 %	0.4039	Run	

Workers: 004 ▾

Autopilot has finished

Run Autopilot on a Different Feature List

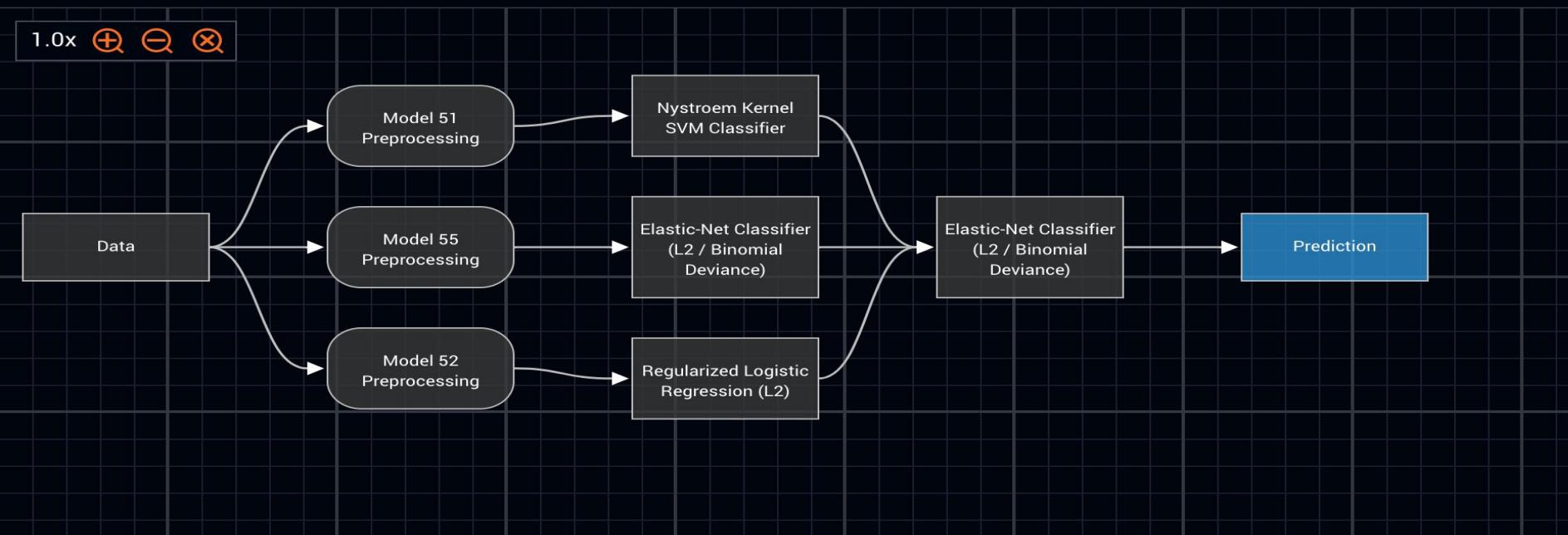
Unlock Holdout

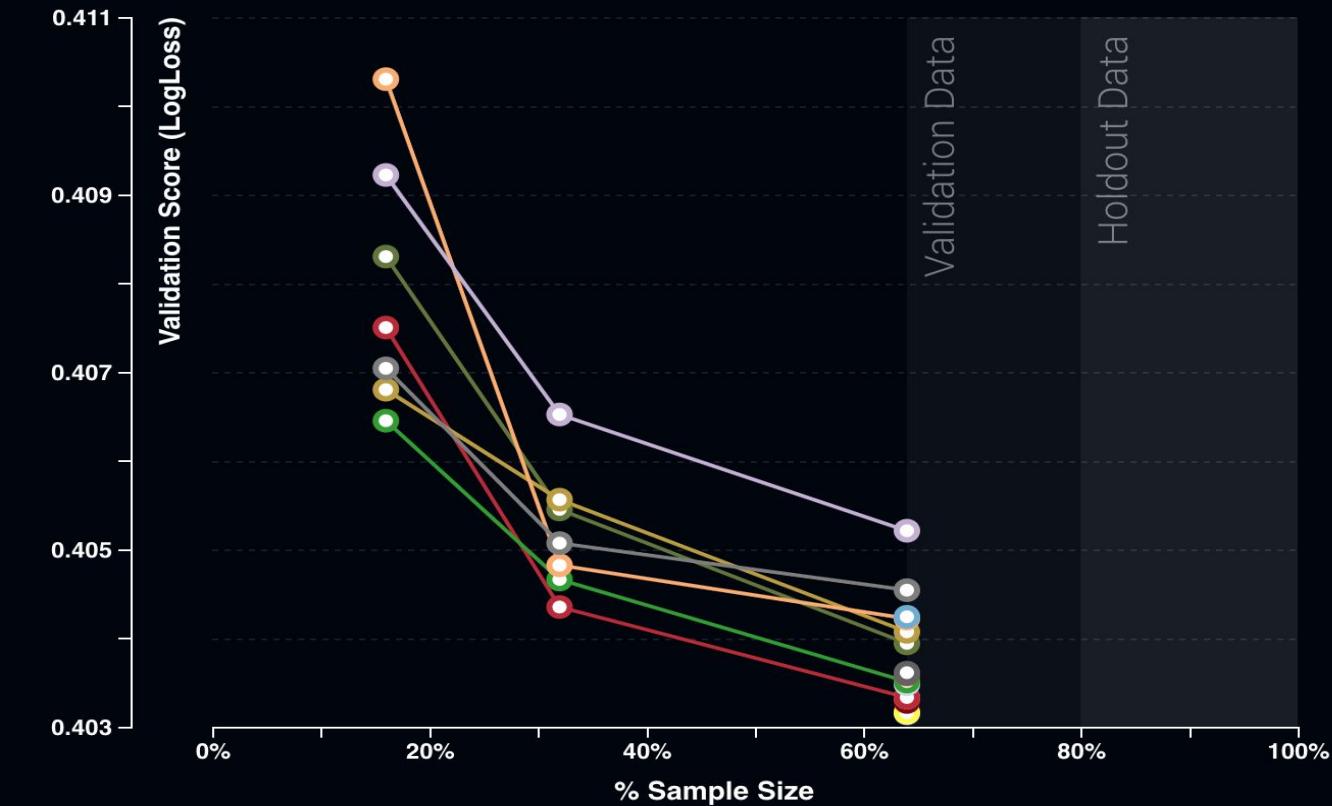
Model Name & Description

Holdout : 0.4085

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

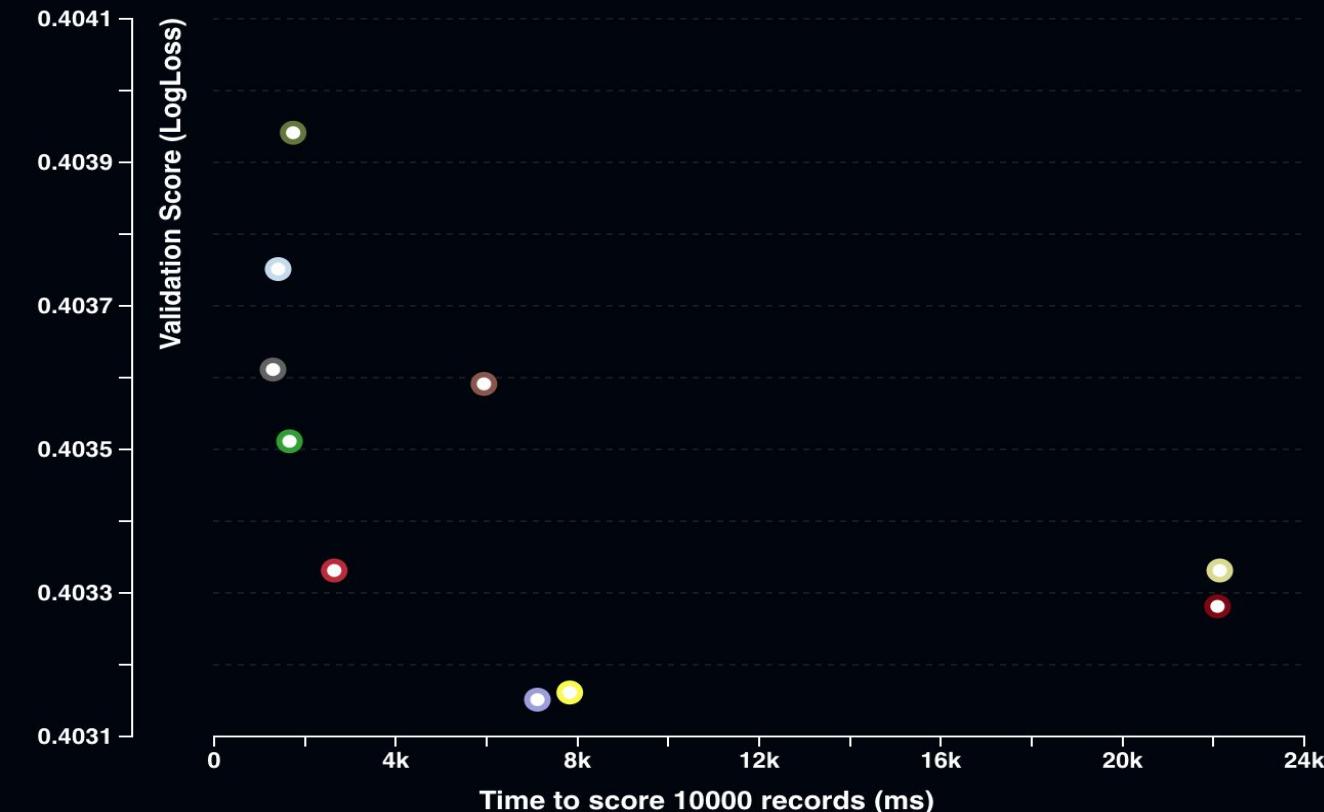
1.0x ⊕ ⊖ ⊗ ⊗





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 ▾

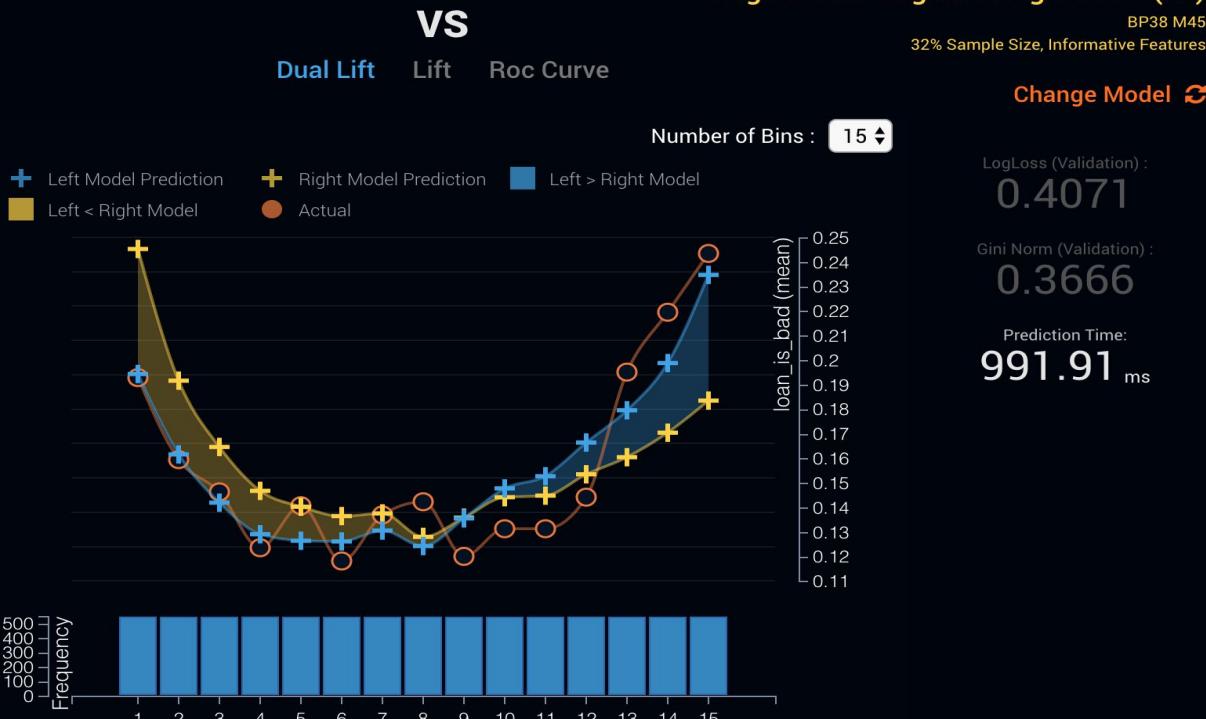
- ENET Blender
M55+51+52, M62
64% Sample Size, Informative Features
- AVG Blender
M55+51+52, M59
64% Sample Size, Informative Features
- ENET Blender
M57+55+51+56+52+..., M61
64% Sample Size, Informative Features
- Advanced AVG Blender
M57+55+51+56+52+..., M60
64% Sample Size, Informative Features
- 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

Leaderboard Learning Curves Speed vs Accuracy Model Comparison

ENET Blender

M55+M51+M52 M62
64% Sample Size, Informative Features

Change Model

LogLoss (Validation) :
0.4032Gini Norm (Validation) :
0.3856Prediction Time:
8368.36 ms

Regularized Logistic Regression (L2)

BP38 M45
32% Sample Size, Informative Features

Change Model

LogLoss (Validation) :
0.4071Gini Norm (Validation) :
0.3666Prediction Time:
991.91 ms

Autopilot has finished

Run Autopilot on a Different Feature List

Unlock Holdout

[Leaderboard](#) [Learning Curves](#) [Speed vs Accuracy](#) [Model Comparison](#)

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

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) :

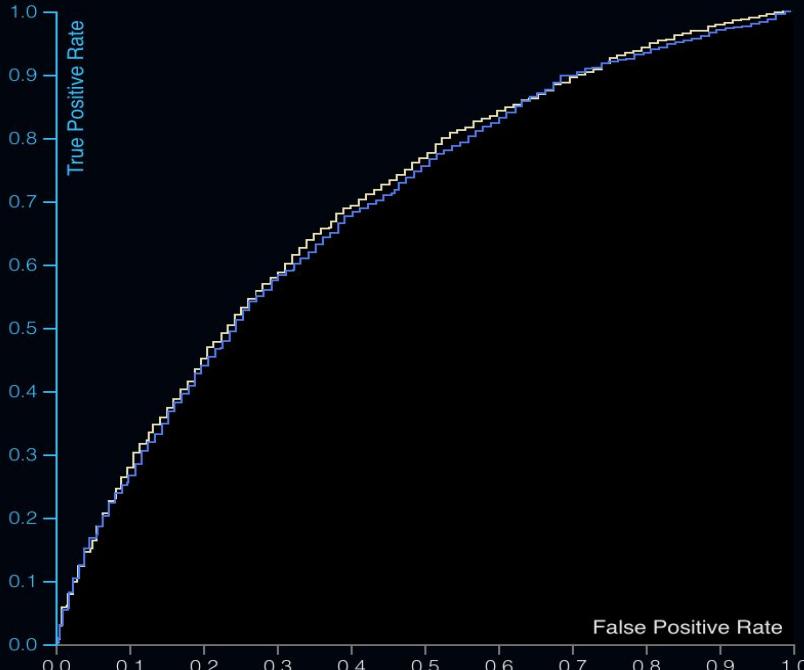
0.6833

Prediction Time:

1055.62 ms

Dual Lift Lift **Roc Curve**

Roc Curve Data Source: [Validation](#) [Holdout](#)



Model Name & Description

Feature List & Sample Size

Validation

Cross Validation

Autopilot has finished

ENET Blender										Informative Features		0.4032		Run	🔒
M62	M55+51+52									64.0 %	+				
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 ↗									Predicted		-	+		
Maximizes F1 Score	- NOT(1)	+ 1									Actual	-	4861 (TN)	1888 (FP)	6749
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			Actual	+	552 (FN)	699 (TP)	1251	
0.3643	0.5588	0.2797	0.7203	0.2702	0.898	0.695	0.2166					5413	2587	8000	

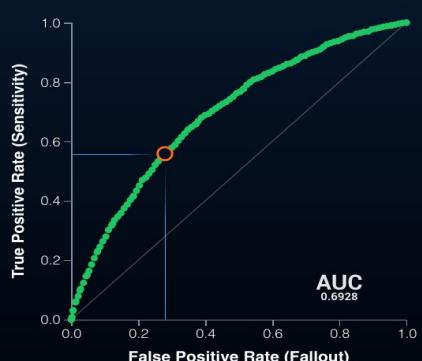
Run Autopilot on a Different Feature List

Section 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	Actual	-	4861 (TN)	1888 (FP)	6749	
0.3643	0.5588	0.2797	0.7203	0.2702	0.898	0.695	0.2166		+	552 (FN)	699 (TP)	1251	
										5413	2587	8000	

ROC Curve

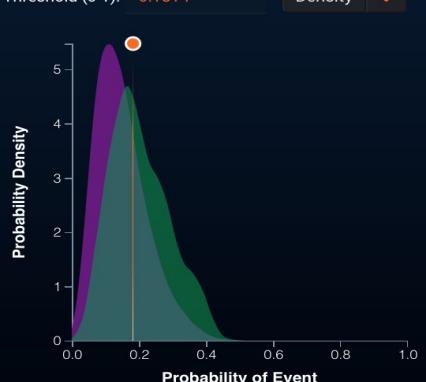
Data Source: Validation ▾



Prediction Distribution

Threshold (0-1): 0.1814

Density ▾



AVG Blender

32

Run

An open padlock icon, indicating that the content is accessible.

[Leaderboard](#) [Learning Curves](#) [Speed vs Accuracy](#) [Model Comparison](#)

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**

Regularized Logistic Regression (L2)

BP34 M44

32% Sample Size, Informative Features

[Dual Lift](#) [Lift](#) [Roc Curve](#)Lift Data Source: [Validation](#)[Holdout](#)Number of Bins : [15](#) Left Model Actual Right Model Actual[Change Model](#)

LogLoss (Validation) :

0.4071

LogLoss (Holdout) :

0.4109

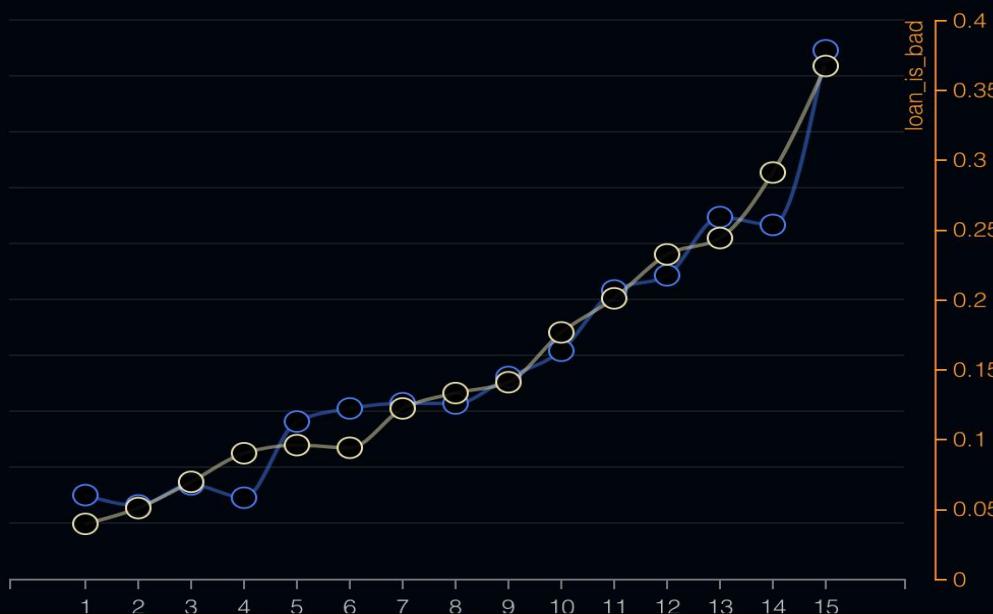
Gini Norm (Validation) :

0.3666

Gini Norm (Holdout) :

0.3297

Prediction Time:

1055.62 ms

Model Name & Description

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

Features



Sort: Effect Size ▾

Export

Search for features



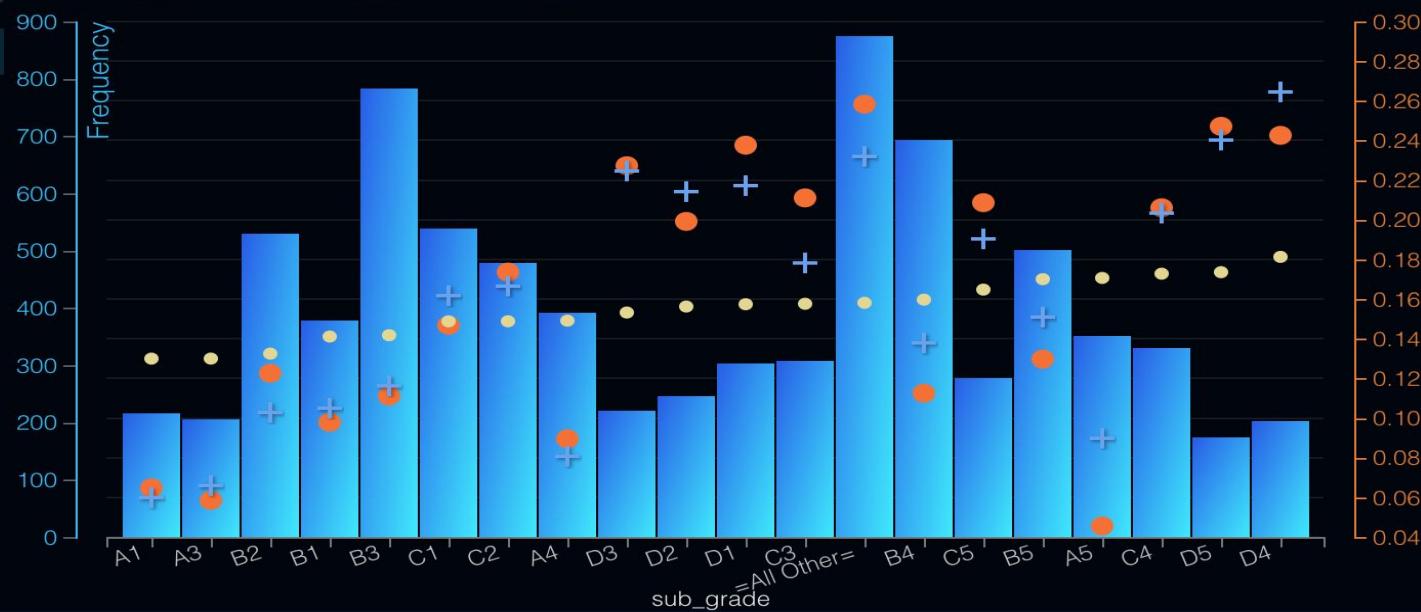
Predicted

Actual

Partial Dependence

sub_grade

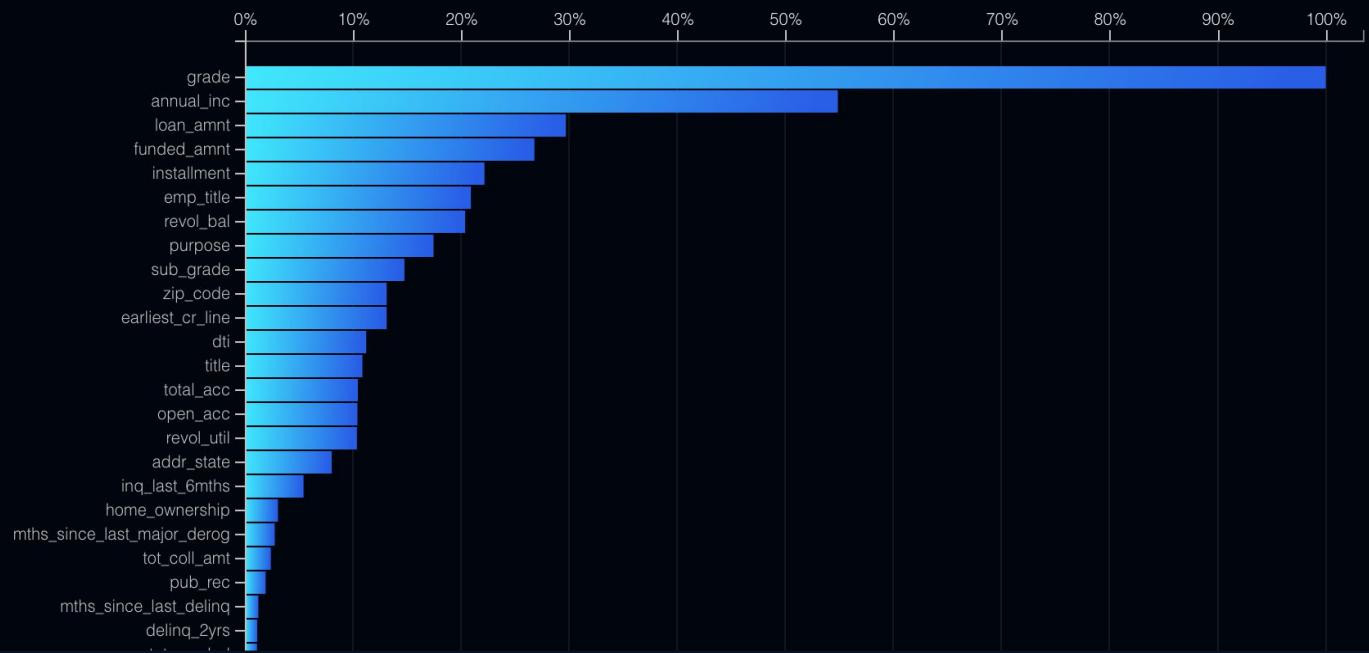
grade



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



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	

Model Name & Description

Feature List & Sample Size

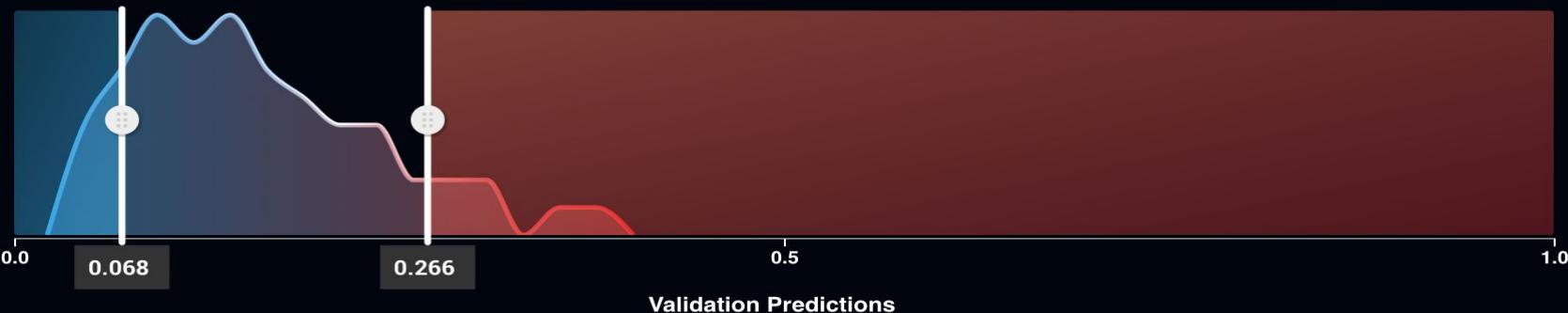
Validation

Cross Validation

Holdout

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	REASONS	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"