Credit default risk model

Cassidy Mentus

Code and repository

- train_models.ipynb – **Create and train models.**
  - Load data, create train test sets, build logistic regression and lightgbm sklearn pipelines with cross-validation.
- analyze_models.ipynb – **Analyze models and data.**
  - load train/test sets, load model .joblib files, analyze models (coefficients, SHAP, numerical results), analyze data.
- logistic_pipeline.html – **sklearn pipeline of logistic regression model.**
  - Includes grid-search cross-validation, feature transformations and model.
- Gbdt_pipeline.html –**sklearn pipeline of light gbm gradiend boosted tree model.**
  - Includes random-search cross-validation, lightgbm model and CV- probability calibration.

Goal

Design, build and benchmark an ML model to predict risk of default given applicant data and credit $ amount.

The model will be able to do the following:

1. Score credit applications. Calculate a risk score given applicant credit indicators that ranks credit by it's default risk for typical credit amounts.
   Used to approve/deny credit applications.
2. Risk score for different credit $ amounts. For approved apps, calculate default risk given applicant info and user input credit amount.
   Used to determine credit limits.
3. Other predictions and estimates. E.g. Expected time to default, probability of default by 12 mo., expected cashflow.

To optimally perform these functions the model must be both **discriminative** and **calibrated** (accurate probabilities).

Project summary

1. Credit default data
   - Given 1000 rows of data consisting of applicant credit variables, duration from time of credit origination, credit $ amount and binary defaulted status.
   - Discuss the process of credit and $ amount approval.
   - Data can be used to estimate the probability of default occurring by any time duration, suggesting a way to …
2. Model credit default risk
   - Given applicant variables and loan amount, we predict the curve of probability of default as time elapses. Equivalent to a survival model approach.
   - Equivalent to the simpler task of classifying default using credit vars, amount and duration as inputs.
   - Candidate models based on gradient boosted decision trees and regularized logistic regression. Probability calibration of tree ensemble model.
3. How to score new applications and risk for different credit amounts.
   - Model outputs numbers for given duration. For an application, we can vary duration producing a curve of probabilities.
   - Score applications by flattening the model output. Average over duration and credit amount from training distribution, producing a single numeric value.

Project outline continued

4. Measuring model performance
   - Develop ways to measure ability to predict risk of default for credit originating at the same time (within cohort)
   - Idea: Calculate the metric for data at each fixed duration. Then take the average over all durations.
   - Present model performance metrics.
   - Model final selection.
5. Results and analysis
   - Demonstrate model outputs such as survival curves and duration x credit amt risk heat maps.
   - Feature importance using Shapely values.

Credit default data

Definitions

- x – Credit variables available upon application.
- a – Amount of credit given by to the customer upon credit approval.
- $t \in \{1,2, \dots, 72\}$ – Duration of time elapsed (rounded to nearest mo.) since time of origination.
- d – default status of the credit. 0 – not defaulted. 1 – defaulted.
- For the i^{th} credit line, the above values are denoted with subscript i : x_i, a_i, t_i.
- T_d random variable representing time of default. Not available in the data. Used in constructing the model.

The credit data generating process

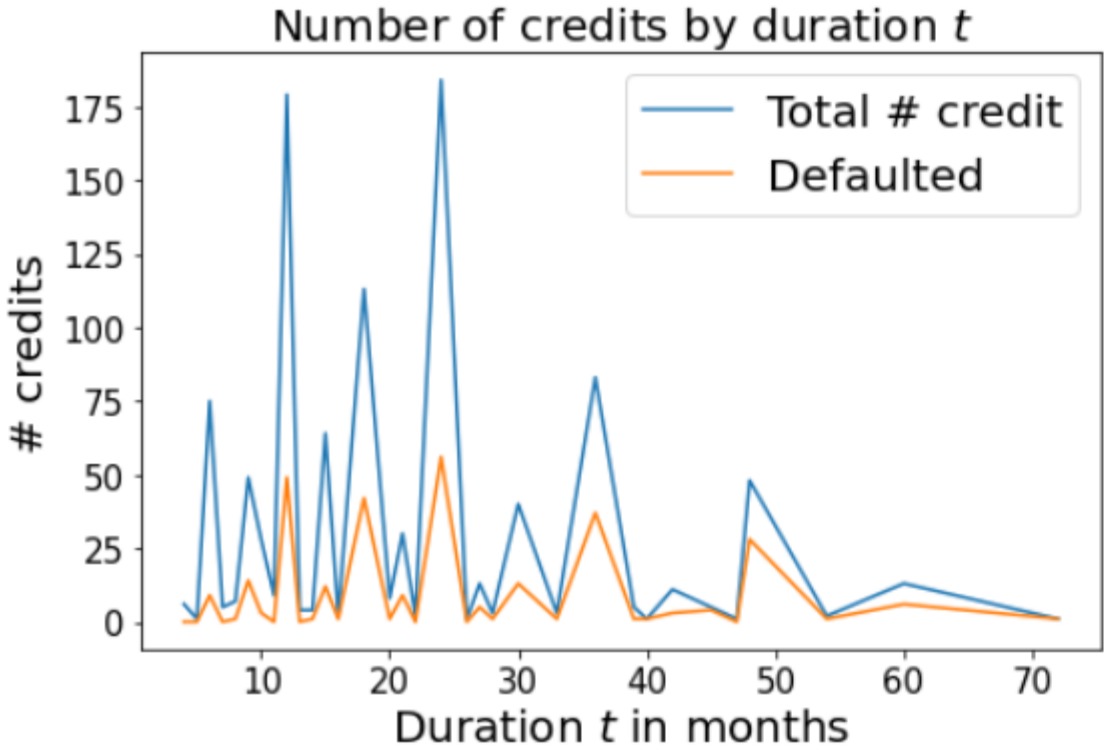
1. Credit application. Credit variables x supplied by applicant, retrieved from information service or internal data.
2. Approved applicants are entered into the dataset.
3. Credit $ amount a assigned after approval. Depends on risk assessment using credit-variables.
4. Record the duration t passing since time of origination (rounded to the nearest month).
5. Record defaulting on payment as a permanent status change. Defaulted credit is kept on file.

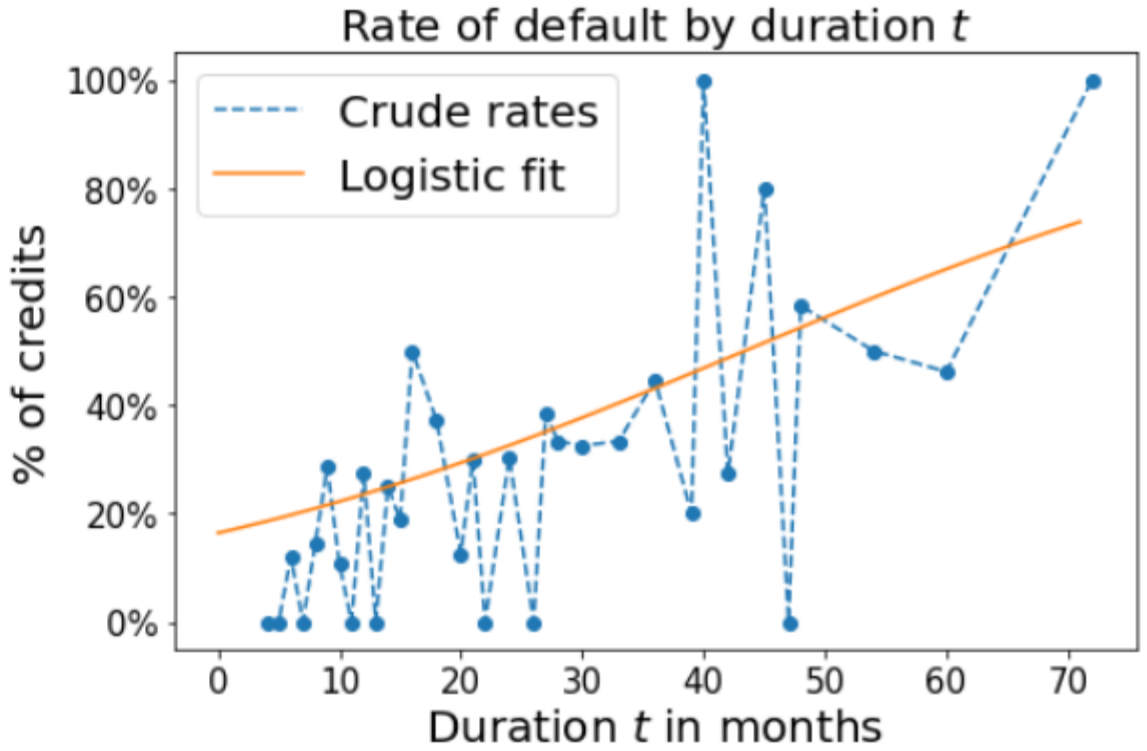
Dataset

Data for 1000 credit lines originating as far back as 6 years ago (72 months) from the current date (date of last data retrieval) with the following variables.

Column(s)	Name	Symbol	Description
CHK_ACCT, …, FOREIGN	Credit variables	x	28 credit features (x is a 28 dimensional vector) available at time of application supplied by applicant, retrieved from internal database or external data vendors.
DURATION	Duration	t	The amount of time in months (rounded to nearest whole number) elapsed since date of origination.
AMOUNT	Credit $-amount	a	$ amount of credit given to the customer. Assigned upon application approval. Assumed to be within a credit limit assigned by First Help Financial depending on the application data.
DEFAULT	Default status	d	Whether credit was defaulted on since time of origination and current date, duration of length t months. 1 – defaulted. 0 – not defaulted.

Data volumes and default rates by duration

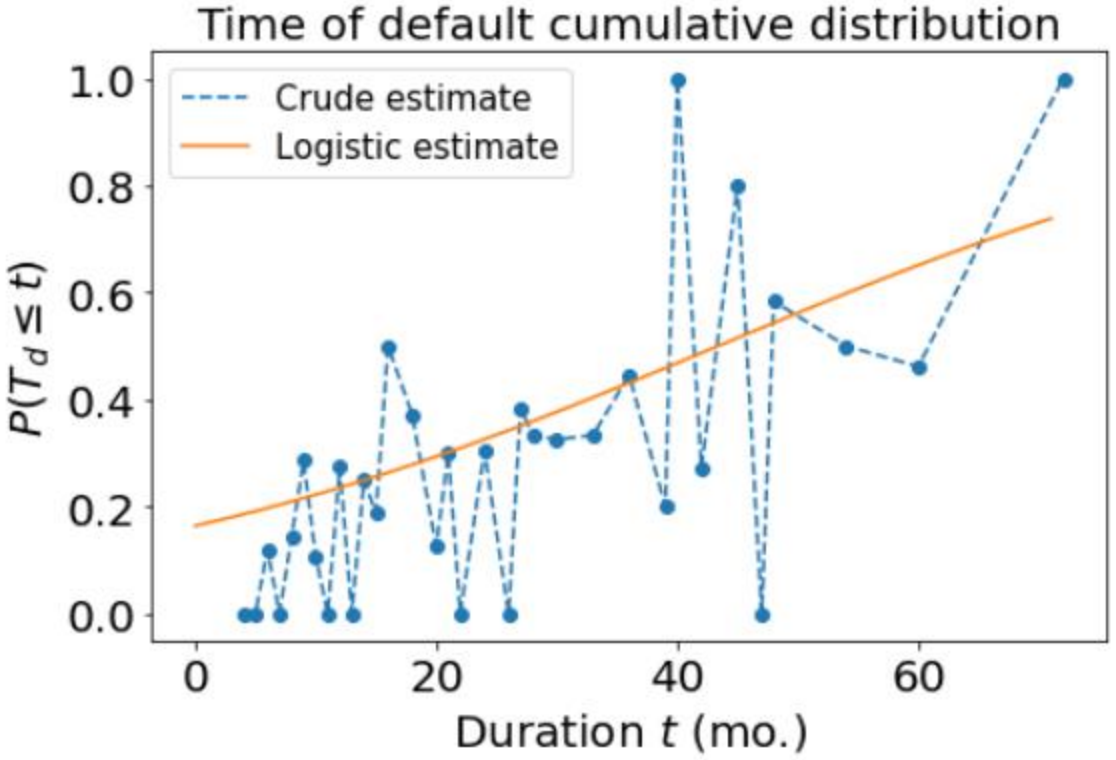


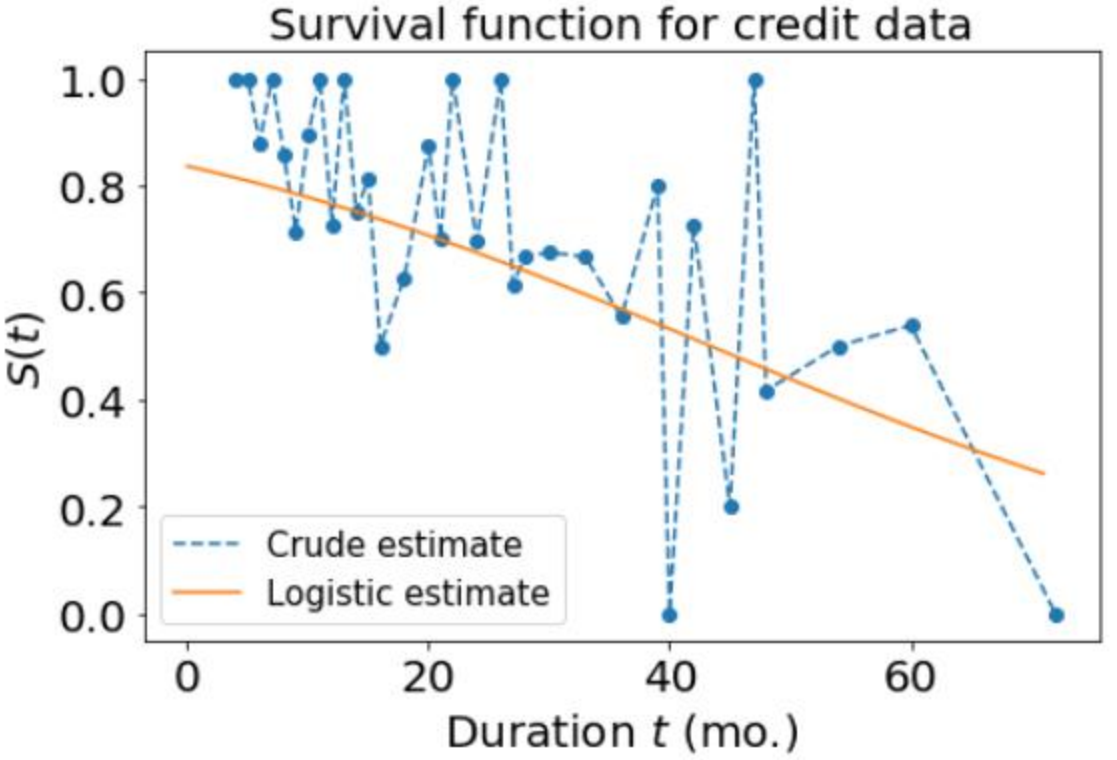


Modeling credit default risk

Predicting the risk that a customer defaults on their credit as time progresses based on credit variables and credit $ amount. Describe and justify model design choices leading up to the model blueprint.

Survival analysis of credit defaults





Right: Same plot as last slide: default rates as a function of duration. The fractions $\frac{\#\,\text{defaulted, duration t}}{\#\text{credits}}$ are crude estimates of $P(T_d \leq t)$ (see definition slide).

Right: The survival function (curve) $S(t) = P(T_d > t) = 1 - P(T_d \leq t)$. Survival models usually predict hazard rates or S(t).
Modeling $P(T_d \leq t \mid a, x)$ can be considered a survival model because its output is easily converted to S(t).

Credit survival expressed as classification problem

Model the prob of default by time t as a function F of x, a, t ,

$$F(x, a, t) \approx P(\, T_d \leq t \mid \text{features } x, \text{credit amt } a\,).$$

The probability on the RHS is equivalent to

$$P(\, \text{DEFAULT} = d_i \mid x_i, a_i, t_i\,),$$

for credit i. Therefore to build a model to predict $P(\, T_d \leq t \mid \text{features } x, \text{credit amt } a\,)$ we can train a classifier F on input variables x_i, a_i, t_i to predict target variable d_i.

The loss function is the negative log likelihood for binomial classification:

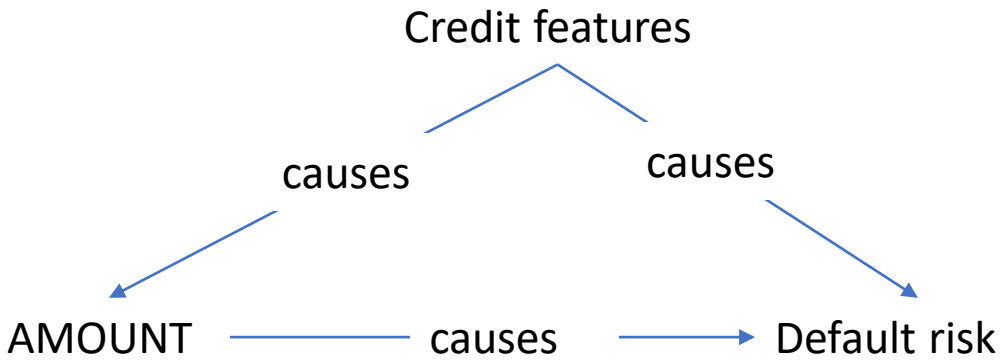
$$-\log\text{-likelihood}(\text{data} \mid F)$$

$$= \sum_{\text{credit line } i} d_i \times \log\big(F(x_i, a_i, t_i)\big) + (1 - d_i) \times \log\big(1 - F(x_i, a_i, t_i)\big)$$

F is trained via minimization of this function (with some penalty terms to prevent overfitting).

Effect of credit $ amount on default risk

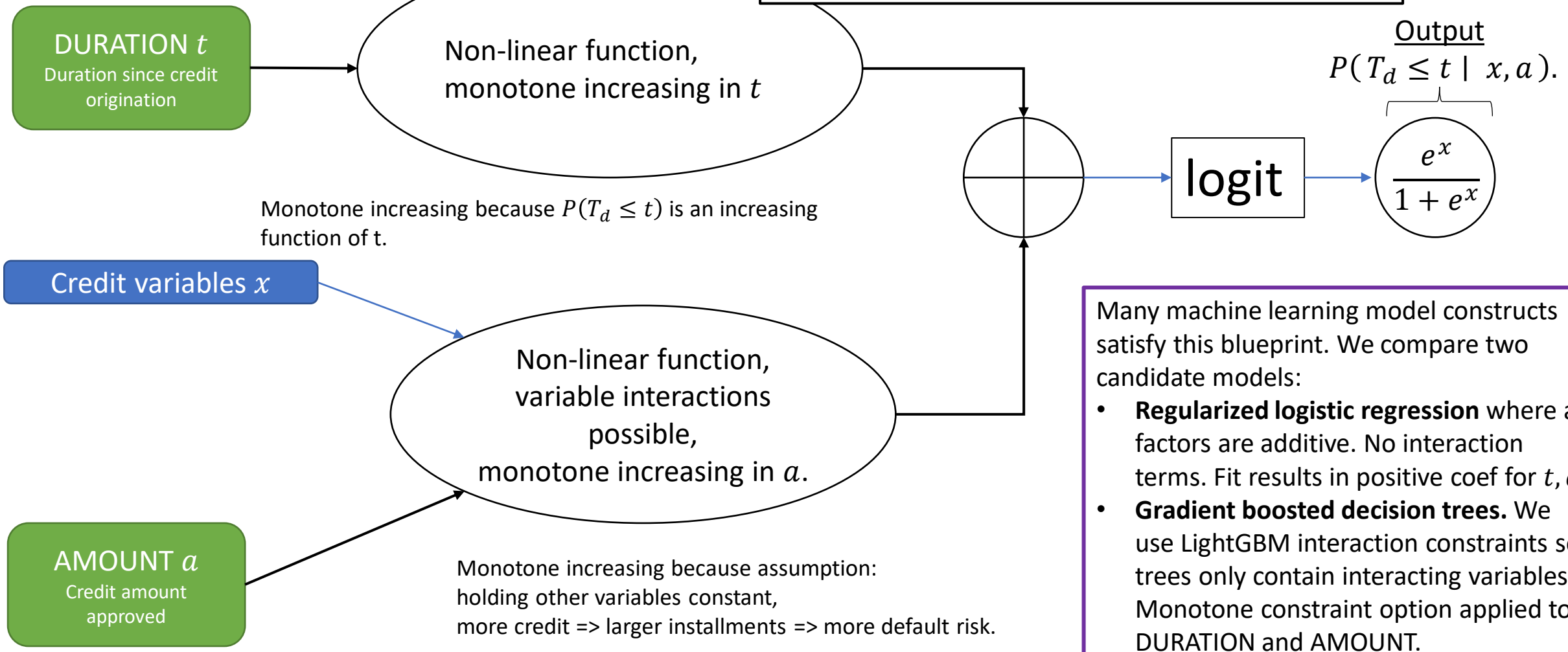
- Predicting credit limits requires estimating the effect different credit amounts have on default risk.
- Data credit amounts are the result of the firm's estimated risk for the customer using applicant data.
- There is a causal link between the customer's credit features and the AMOUNT variable in the data.



- Since credit variables cause both the
- Accounting for the confounding credit variables requires interactions terms with AMOUNT and credit features in the model.

Model blueprint

Duration t has an additive effect, separated from interactions with other features.
Results in credit time-risk curves having same baseline trend shifted higher/lower by a function of a, x.
Guarantee risk extrapolates to later durations.



Many machine learning model constructs satisfy this blueprint. We compare two candidate models:

- **Regularized logistic regression** where all factors are additive. No interaction terms. Fit results in positive coef for t, a**.**
- **Gradient boosted decision trees.** We use LightGBM interaction constraints so trees only contain interacting variables. Monotone constraint option applied to DURATION and AMOUNT.

Model descriptions

Design and describe in details models to predict default risk according to our blue print. Models represent the two major opposing approaches to ML:

1. Explainable by design, transparent, and often parametric.
2. Algorithmic, black-box, yet expressive models. Potential for high performance.

Logistic regression. Gradient boosted decision trees.

Model description: Logistic regression (LR)

Feature interactions and constraints

See appendix for implementation details

- Features are additive factors. See the next slide for feature transformations.
- Fitting with interaction terms had good performance but the model violated assumption of increasing risk with unilaterally increasing credit AMOUNT.

Regularized logistic regression

- To prevent overfitting we use elastic net regularization.
- Elastic net combines the following penalties
  - Lasso, $\|\beta\|_1 = \sum_{i=1}^{30} |\beta_i|$ – leads to parsimonious model. Causes smaller, less significant coefficients to shrink to 0.
  - Ridge, $\|\beta\|_2^2 = \sum \beta_i^2$ – stabilizes regression when variables are multicollinear. encourages ~similar effect size for the most correlated variables.
- Elastic net: $\theta\|\beta\|_1 + (1-\theta)\|\beta\|_2^2$ where tuneable $0 \leq \theta \leq 1$ controls relative amounts lasso and ridge penalties.
- Maximum likelihood with elastic-net penalty (mixture of lasso and ridge)

  $\text{minimize: } -\log\text{−likelihood}(\text{ data} \mid \beta, \beta_0\ ) + \lambda(\theta\|\beta\|_1 + (1-\theta)\|\beta\|_2^2).$

  λ controls total penalty amt.
- Optimal λ, θ selected using cross validation. See model selection slide.

Logistic regression feature pre-processing

- **Categorical variables** are **one-hot encoded**.
  - In the future we may use numeric values as one variable because some categorical variables represent ordinal variables such as increasing bins for check account balance.
- **Numeric variables**, except for AMOUNT, are transformed using **power transform (Yeo Johnson transform)**.
  - Power transforms, imprecisely, raise feature to optimal power γ so that the resulting distribution is ~Normal. An edge case is $\gamma = 0$ $\mathrm{x} \mapsto \log x + 1$.
  - Automatic way to find appropriate scaling, remove skew (e.g. $ amounts may be log-normal distributed). Approx normal variables are unaltered.
  - Otherwise we would have to look at definitions, histograms to determine transformations.
- **Binary variables** take on **0-1 values**. We apply min-max scaling in-case the variable is not scaled.
- **Amount credit**. Although this is ~log-normal, multiplicatively **scale interquartile range to 1**.
  - Reasons: Lasso zeros out coef if we use log transform. Better cross validation performance.
  - Preferred method, including AMOUNT x feature interactions, resulted in invalidating assumption: increasing AMOUNT -> increasing default rate.

Model description: Calibrated gradient boosted decision trees (GBDT)

See appendix for implementation details

Implementation

- LightGBM (Microsoft) GBDT is used for its performance, speed, efficient handling of categorical variables and options for model tuning.
- DART booster drops a 10% of trees each iteration preventing overfitting and sensitivity to number of iterations.

Feature interactions and constraints

- LightGBM interaction constraint applied to allow interactions within the groupings {DURATION}, {AMOUNT, all other credit features}.
- Results in two types of decision trees:
  - trees containing only splits along DURATION,
  - trees containing splits along any input variable except for DURATION.
- Monotone increasing constraint applied to DURATION and AMOUNT.
- Feature transformations are unnecessary for tree-based models. LightGBM does not require one-hot encoding of categorical variables.

Probability calibration

- Tree ensembles, although discriminative classifiers, tend to produce inaccurate likelihood estimates. They are uncalibrated.
- Logistic regression does not face this problem.
- Cross-validation (CV) calibration with logistic regression method:
  - Generate 5 fold cross-validation train-test subset pairs.
  - For each fold: fit GBDT on ‘train’ subsets + fit logistic regression: GBDT outputs -> DEFAULTED on ‘test’ subsets
  - Finally, ensemble average the 5 GBDT -> logistic regression pipelines.

Credit risk scores

Using the model, we define a single numeric credit score to form a criteria for credit approval/denial given their application info. Define a score incorporating credit amount to use in deciding credit limits.

Score definitions

Base score

For x, a, t define the basic risk score is

$$\text{Score}(x, a, t) = F(x, a, t).$$

How to score using only subsets of features

- Inputs AMOUNT and DURATION are not included in the application.
- Create a score using subset of variables by taking the expectation over excluded variables using training data.
- Denoted with excluded variable in the subscript.

Scores for application approval and credit limit approval

- **Application score.** Applications only contain credit variables x. The average risk of applicants with similar credit amounts:

$$\text{Score}_{a,t}(x) = \frac{1}{N_{\text{train}}} \sum_{i \in \text{train}} F(x, a_i, t_i).$$

- **Credit limit score.** Score is a function of x, a is because it measures risk of a particular customer defaulting on a specific $ amount credit.

$$\text{Score}_{t}(x, a) = \frac{1}{N_{\text{train}}} \sum_{i \in \text{train}} F(x, a, t_i).$$

Model performance benchmarks and selection

Define useful metrics for measure the performance of models predicting default risk over time. Train (cross-validate) and benchmark scores (see last section) for both models. Report cross-validation and benchmarking metrics. Select one model from our choices.

Model performance metrics

Log likelihood

- How well probability predictions (directly from the model) describe the data.
- Measures how well model predicts different default risk of credit from same time of origination.
- Main cross-validation metric for hyperparameter tuning.

Avg t-AUC

- AUC, although a choice metric for classification, is inadequate for measuring performance of ranking default risk of credit originating at the same time.
- Recall: AUC is equivalent to the probability, given two datapoints with different classes {0,1}, the score correctly ranks them by giving the point in class 1 a higher score. Expressed mathematically

$$\mathrm{AUC} = P(\,\mathrm{score}(\mathrm{a}) \leq \mathrm{score}(b) \mid \mathrm{class}(b\,) = 1\,, \mathrm{class}(\mathrm{a}){=}0).$$

- Shortcoming: Compares pairs of credit with different durations which is strongly influenced by likelihood of default growing over time.
- **Solution:** For data with duration $== t$**,** compute the AUC (**t-AUC**). Compute the average of all t-AUC.

AUC

- AUC measures the ability to classify default considering all variables duration included.

Training and benchmark setup

- Training and testing datasets selected using 80:20 random stratified train-test split.
  - Training data – 800 rows
  - Testing data – 200 rows
- Hyperparameters for both models tuned on the training data using the same stratified 10-fold cross-validation folds.
- Both cross validations use same folds, making it valid to compare cv metrics for both model constructs.

Model performance

Logistic regression

Benchmark performance (test set)

Score name	Score	Metric	Value
Base score	Score	-(log-likelihood)	**.48**
Credit limit score	Score_t	Weighted mean t-AUC	**.79**
Application score	Score_{at}	Weighted mean t-AUC	**.79**
Base score	Score	AUC	**.81**

Cross validation (training set)

Score type	Metric	Value
Score	-(log-likelihood)	-0.5
Score	AUC	.78

Gradient boosted decision trees

Benchmark performance

Score name	Score	Metric	Value
Base score	Score	-(log-likelihood)	.50
Credit limit score	Score_t	Weighted mean $t-\mathrm{AUC}$	.76
Application score	Score_{at}	Weighted mean $t-\mathrm{AUC}$	.74
Base score	Score	AUC	.80

Cross validation (training set)

Score type	Metric	Value
Score	-(log-likelihood)	-0.5
Score	AUC	.78

Model selection

- Cross validation metrics for logistic and GBDT models are too close to unambiguously declare one to be the winner.
  - Cross validation metrics are equal up to 2 decimal places while standard deviation is ~.04.
- Logistic regression performs better on benchmarks, especially time-averaged AUC.
  - Might be due to averaged predictions over AMOUNT combined with other covariates are unlikely, requiring some extrapolation. Tree models perform poorly on extrapolation.
  - The standard errors are still too large to make a call based only on metrics.

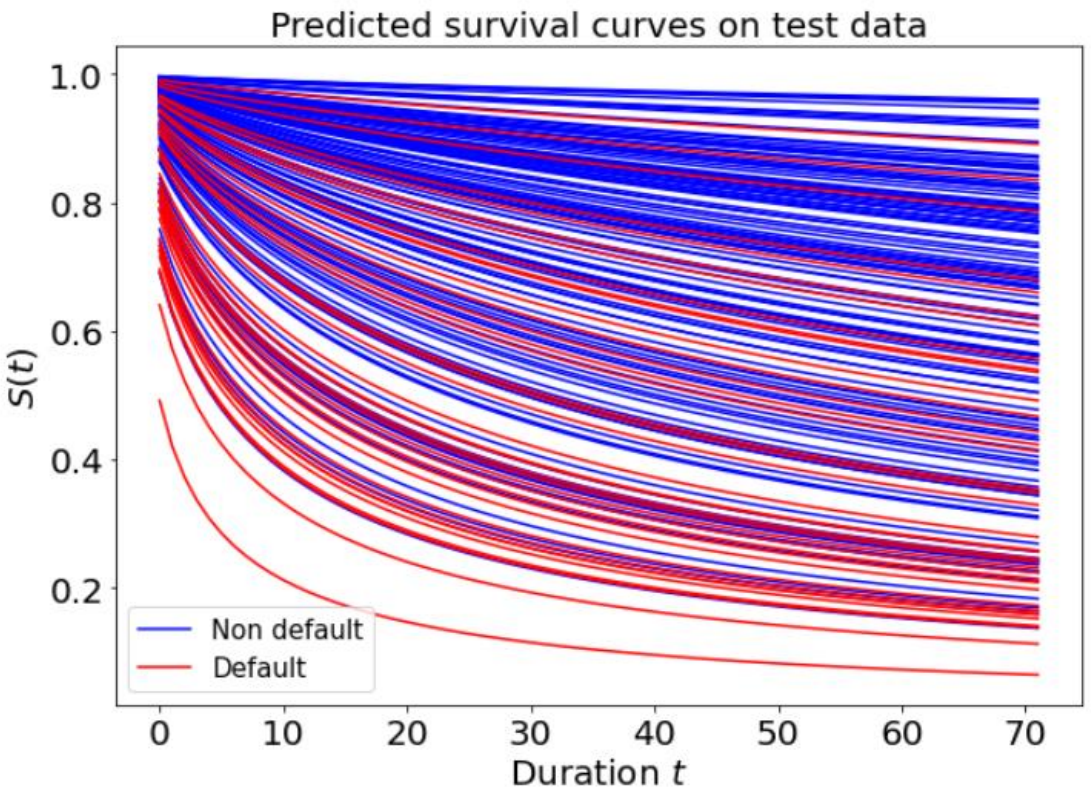
Select **logistic regression** for the following reasons:

- Assumption of linear relation between credit variables and risk makes sense and will extrapolate reasonably well to unseen combinations of AMOUNT and credit variables in a predictable manner.
- More confident about the strengths, limitations of predictions and estimating effects of credit amount and duration.
- Readily interpretable. Easier to adhere to regulations and give adverse action notices for denials.

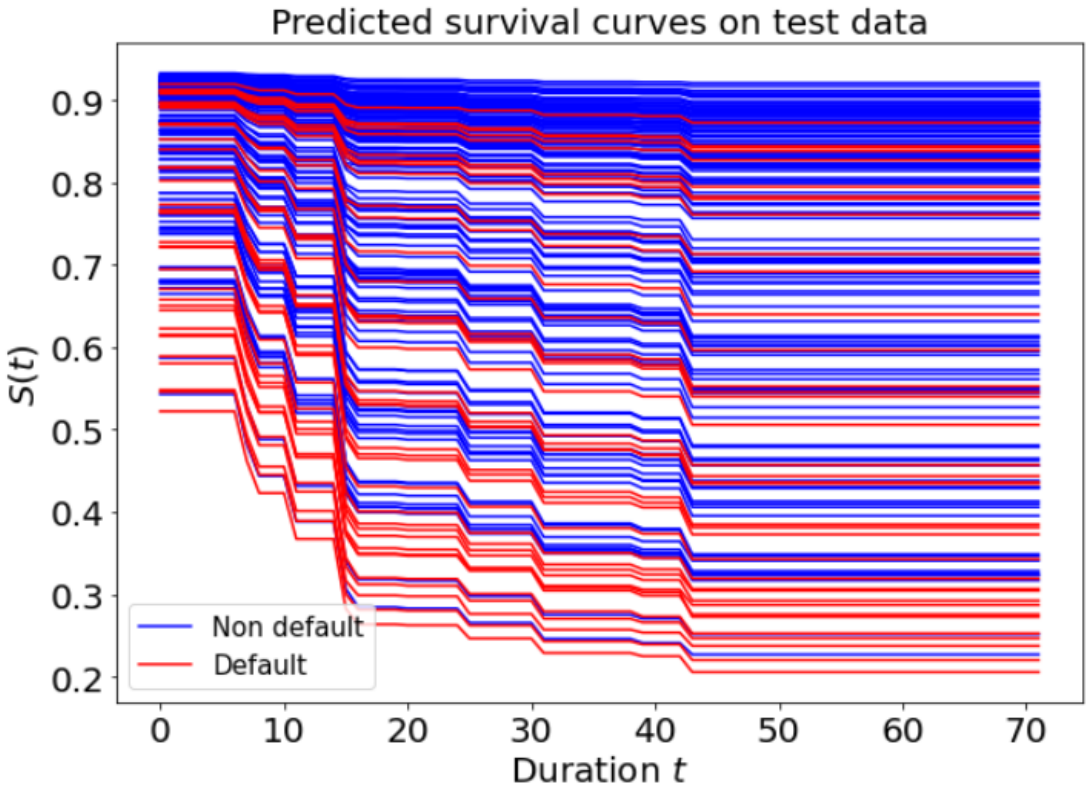
Results and analysis

Compute and analyze predictions and estimates using our models. Analyze model structure through parameters and feature importance.

Survival curves



Logistic regression



Gradient boosted decision trees

To get a feel for our models' predictions, we plot predicted survival curves for test data with duration < 24 mo. using credit features

Default risk given credit amount and duration

Heatmaps of default risk as a function of credit amt and duration predicted by the models with application data fixed. Examples from test data .

top row: true default =0, bottom row: truedefault = 1.

Example: Default =0

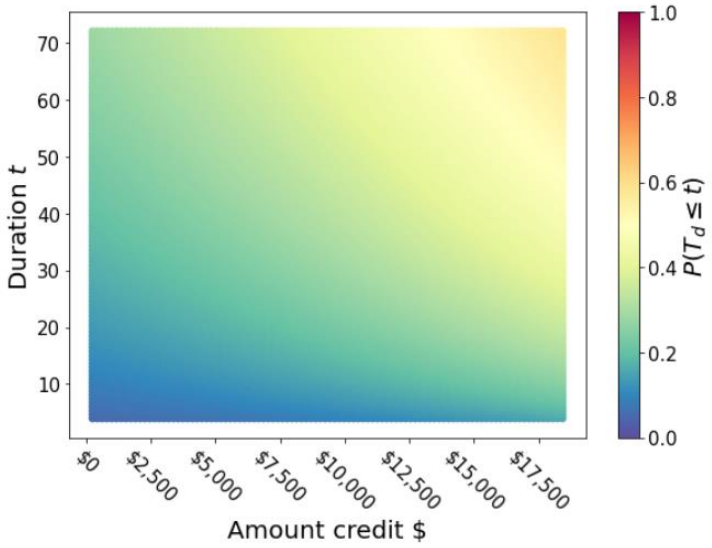
The logistic regression keeps increasing risk as credit amount grows to extremes while the GBDT model flattens out around $12.5k.

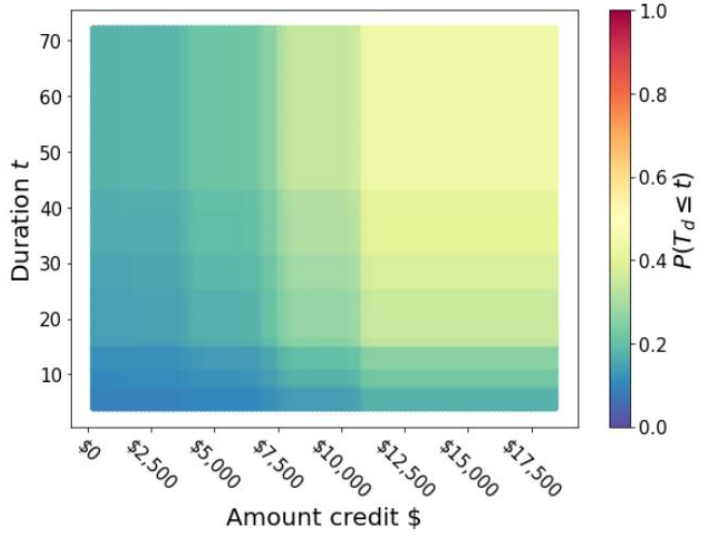
Logistic regression seems to recommend more risk adverse policies.

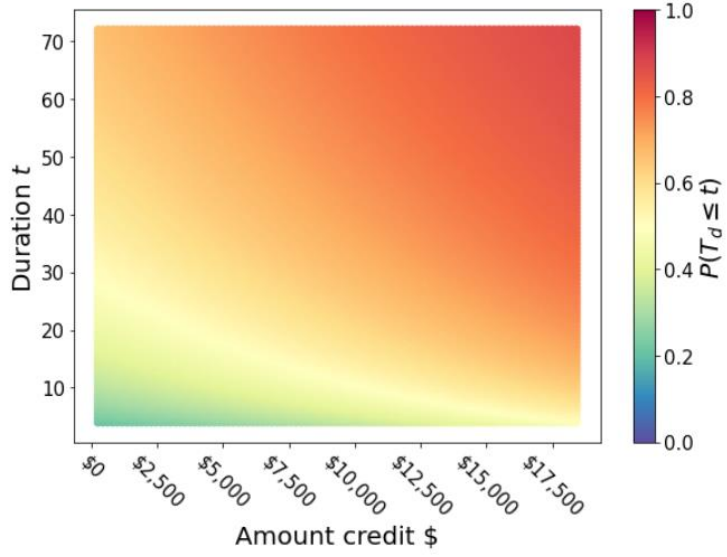
Example: Default =1

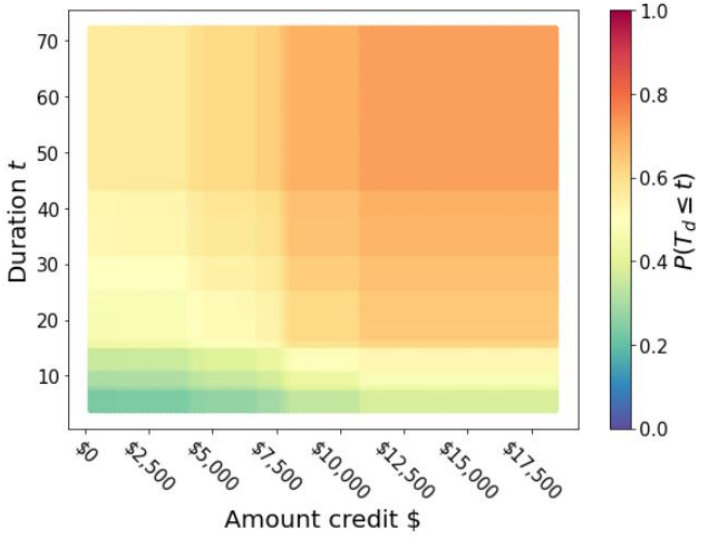
Logistic model

GBDT model









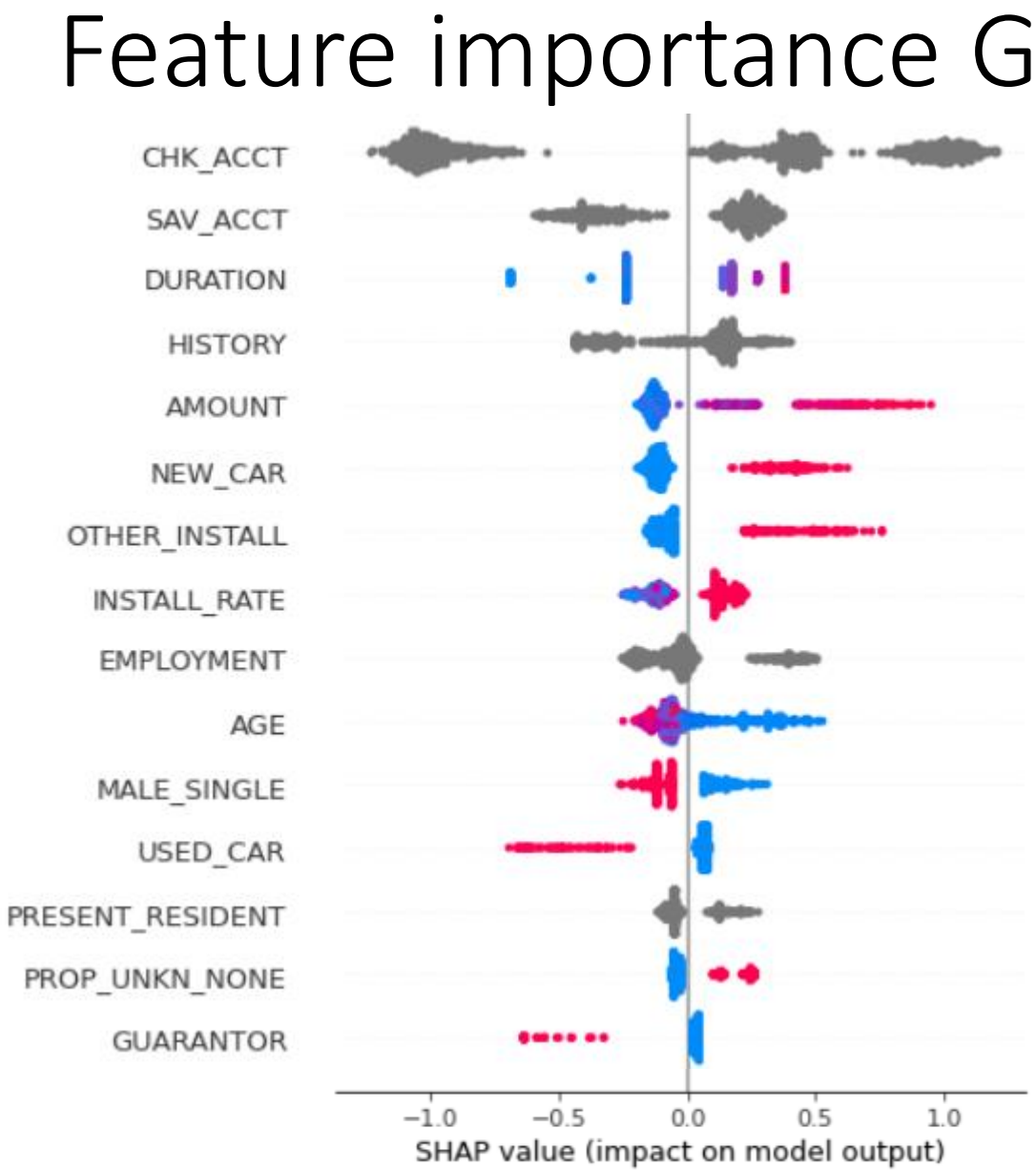
Feature importance logistic regression

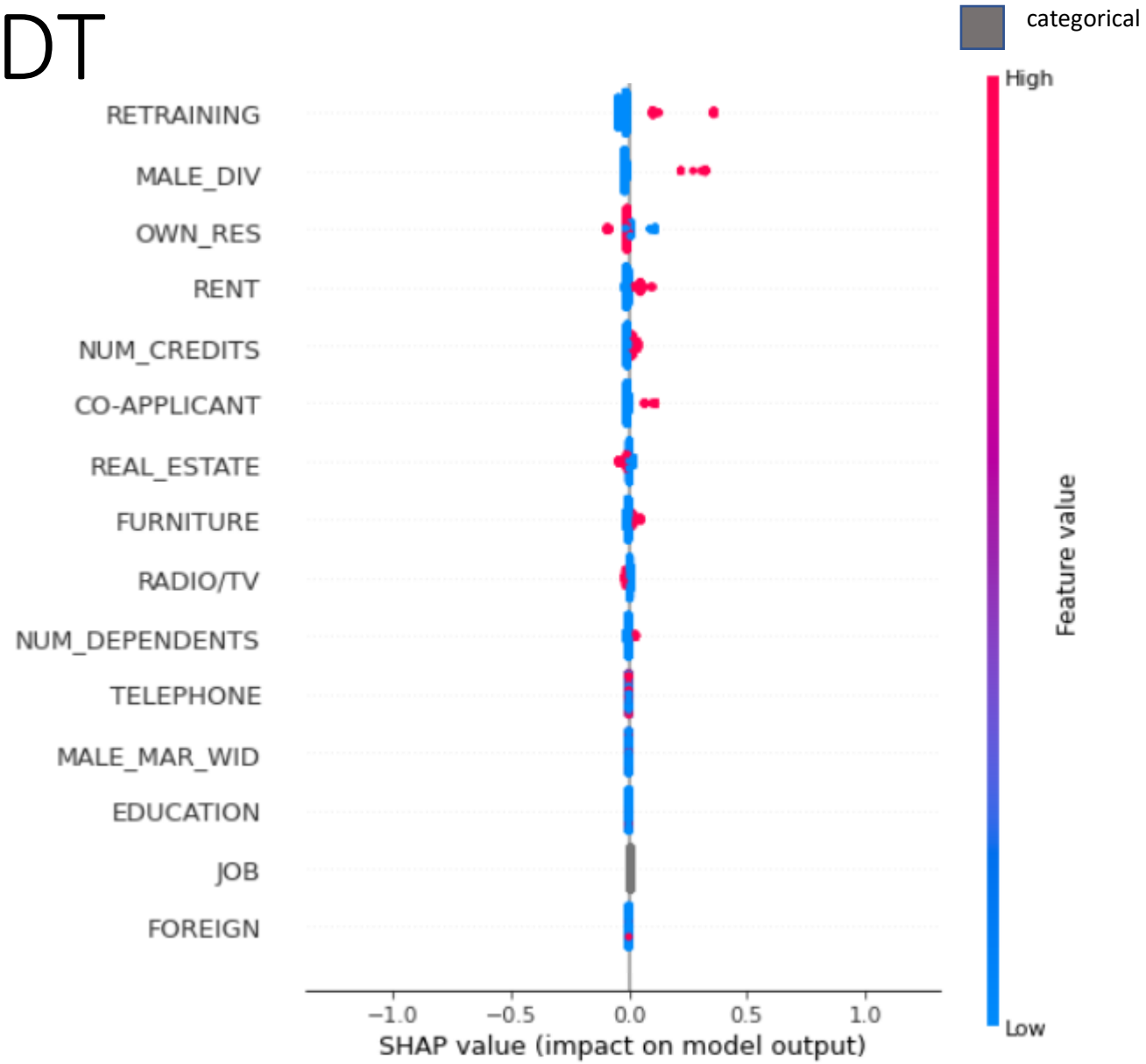
Feature	Transformed feature	Coefficient	Feature importance	Transformed feature importance
CHK_ACCT	CHK_ACCT_0	0.76	0.85	0.30
CHK_ACCT	CHK_ACCT_1	0.20	0.85	0.08
CHK_ACCT	CHK_ACCT_2	0.00	0.85	0.00
CHK_ACCT	CHK_ACCT_3	-0.99	0.85	0.47
HISTORY	HISTORY_1	0.42	0.50	0.04
HISTORY	HISTORY_0	0.25	0.50	0.02
HISTORY	HISTORY_2	0.12	0.50	0.06
HISTORY	HISTORY_3	-0.25	0.50	0.04
HISTORY	HISTORY_4	-0.84	0.50	0.35
SAV_ACCT	SAV_ACCT_0	0.43	0.40	0.21
SAV_ACCT	SAV_ACCT_1	0.20	0.40	0.04
SAV_ACCT	SAV_ACCT_2	0.00	0.40	0.00
SAV_ACCT	SAV_ACCT_3	-0.34	0.40	0.03
SAV_ACCT	SAV_ACCT_4	-0.42	0.40	0.13
DURATION	DURATION	0.40	0.33	0.33
EMPLOYMENT	EMPLOYMENT_1	0.49	0.28	0.14
EMPLOYMENT	EMPLOYMENT_0	0.04	0.28	0.00
EMPLOYMENT	EMPLOYMENT_2	0.00	0.28	0.00
EMPLOYMENT	EMPLOYMENT_4	-0.07	0.28	0.03
EMPLOYMENT	EMPLOYMENT_3	-0.38	0.28	0.11
NUM_CREDITS	NUM_CREDITS	0.23	0.22	0.22
PRESENT_RESIDENT	PRESENT_RESIDENT_2	0.38	0.22	0.16
PRESENT_RESIDENT	PRESENT_RESIDENT_4	0.00	0.22	0.00
PRESENT_RESIDENT	PRESENT_RESIDENT_3	-0.08	0.22	0.02
PRESENT_RESIDENT	PRESENT_RESIDENT_1	-0.18	0.22	0.04
INSTALL_RATE	INSTALL_RATE	0.24	0.22	0.22

Feature	Transformed feature	Coefficient	Feature importance	Transformed feature importance
NEW_CAR	NEW_CAR	0.55	0.20	0.20
OTHER_INSTALL	OTHER_INSTALL	0.58	0.17	0.17
AMOUNT	AMOUNT	0.19	0.14	0.14
MALE_SINGLE	MALE_SINGLE	-0.26	0.13	0.13
USED_CAR	USED_CAR	-0.62	0.11	0.11
RENT	RENT	0.32	0.09	0.09
REAL_ESTATE	REAL_ESTATE	-0.21	0.08	0.08
TELEPHONE	TELEPHONE	-0.17	0.08	0.08
PROP_UNKN_NONE	PROP_UNKN_NONE	0.26	0.07	0.07
AGE	AGE	-0.08	0.07	0.07
RADIO/TV	RADIO/TV	-0.14	0.06	0.06
GUARANTOR	GUARANTOR	-0.53	0.05	0.05
FOREIGN	FOREIGN	-0.51	0.04	0.04
CO-APPLICANT	CO-APPLICANT	0.39	0.03	0.03
NUM_DEPENDENTS	NUM_DEPENDENTS	0.04	0.03	0.03
MALE_DIV	MALE_DIV	0.30	0.03	0.03
EDUCATION	EDUCATION	0.24	0.02	0.02
JOB	JOB_0	0.03	0.01	0.00
JOB	JOB_3	0.00	0.01	0.00
JOB	JOB_2	0.00	0.01	0.00
JOB	JOB_1	-0.02	0.01	0.01
FURNITURE	FURNITURE	-0.02	0.01	0.01
RETRAINING	RETRAINING	0.00	0.00	0.00
OWN_RES	OWN_RES	0.00	0.00	0.00
MALE_MAR_WID	MALE_MAR_WID	0.00	0.00	0.00

Importance = $\sum_{data\ point\ j} \left| \beta_{ij} x_{ij} - avg\left(\beta_{i,.} x_{i,.}\right) \right|$. This is equivalent to Shapely values (interventional). For categorical variables with sum importance over categories.

Feature importance GBDT





Contribution, importance = interventional Shapely values. We plot the feature contributions for each point and the feature value (small =blue, large = red). This shows the trend for each variable.

Appendix

Implementation details

Logistic regression

- sklearn's LogisticRegression model class.
- Penalty= 'elasticnet'
- Solver = 'saga' (stochastic average gradient accelerated method) because it is able to quickly, accurately optimize the lasso penalty.
- Feature transformations concatenated to return one dataframe using sklearn ColumnTransformer. Sklearn Pipeline to create one model instance for the composition of transformation -> LogisticRegression.
- See train_models.ipynb for code.

Gradient boosted decision trees

- Lightgbm.LGBMClassifier for compatibility with sklearn.
- Boosting type = dart
- Hyper parameters= number of iterations, learning rate, max tree depth, l2 regularization, fraction of features used for each tree, bagging.
- Calibration with sklearn CalibratedClassifierCV using a final logistic regression layer (Platt's method).
- See train_models.ipynb for code.

Hyperparameter tuning

- For both models 10 fold stratified cross validation maximizing log-likelihood on test sets.
- We report AUC but do not use it because it compares data at different DURATIONs so it is not a good indicator of our ability to rank credit with the same time of origination.

Logistic regression

- Logistic regression uses grid-search (sklearn.model_selection.GridSearchCV).
- Regularization rate λ is chosen from 10 equally logarithmically spaced values between (including) 1E-4 and 1E4.
- Lasso-ridge mixing parameter θ is chosen from 10 equally linearly spaced values between (including) 0 and 1.

LightGBM

- LightGBM uses random search (sklearn.model_selection.GridSearchCV). with 64 random hyperparameter combinations.
- Max depth chosen from powers of 2 {1,2,…,16}
- N iterations from {50,100,,200,400}
- L2 regularization: 50 logarithmically spaced values between 1E-3, 1E3.
- Feature fraction {.25,.5,1}
- Bagging frequency from {No bagging, 1, 5, 10}. Bootstrap samples the same size as the data.