Predicting Repaid Loans

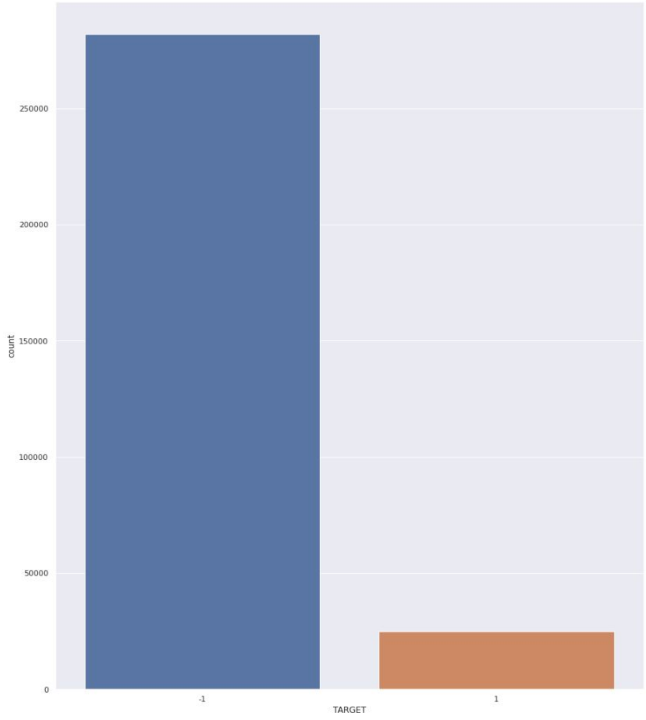
Ryan, Casey, Rachel

The Data

- 307,511 rows
- 122 columns
- Target
  - 1: Client had difficulties with loan payments
  - 0: No difficulties with loan payments

Data Cleaning

- Null Values
  - 62 columns with greater than 1000 null values
  - Remaining rows with null values dropped
  - Left with 306,562 rows, 60 columns
- Target column
  - 0’s - 281,810
  - 1’s - 24,752
  - Incredibly imbalanced

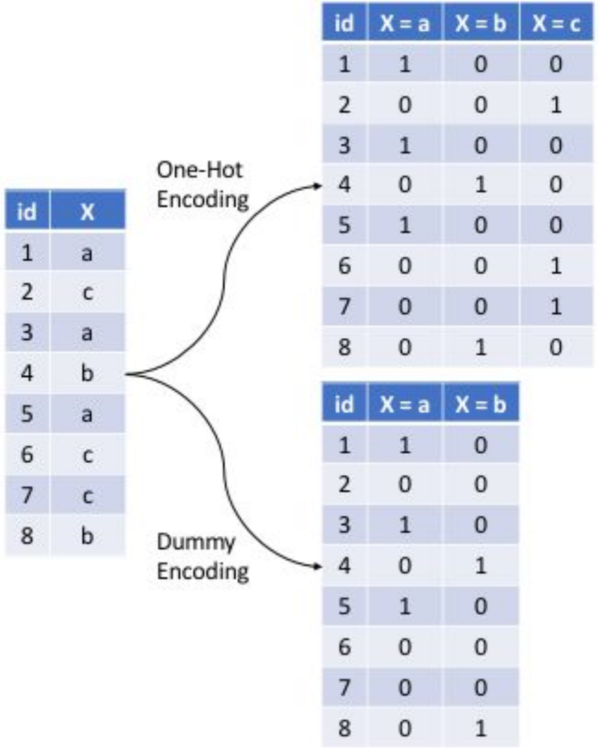


Variable Selection - Domain Knowledge

- 17 Features:
  - AMT_GOODS_PRICE, NAME_CONTRACT_TYPE, CNT_CHILDREN, etc.
  - Left gender variable out
  - A lot of variables seemed to have nothing to do with defaulting

Data Formatting

- One-hot encoding
  - 38 features
- Standard Scaling
- AMT_GOODS_PRICE variable
  - AMT_GOODS_PRICE / AMT_CREDIT



id	X
1	a
2	c
3	a
4	b
5	a
6	c
7	c
8	b

id	X = a	X = b	X = c
1	1	0	0
2	0	0	1
3	1	0	0
4	0	1	0
5	1	0	0
6	0	0	1
7	0	0	1
8	0	1	0

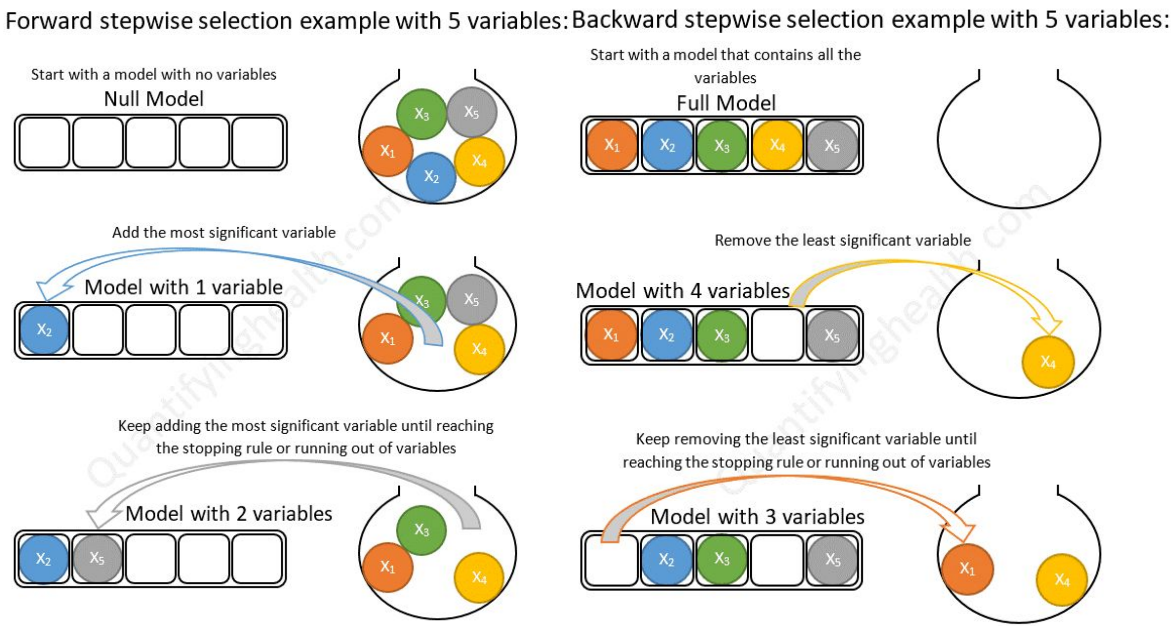
id	X = a	X = b
1	1	0
2	0	0
3	1	0
4	0	1
5	1	0
6	0	0
7	0	0
8	0	1

Variable Selection Options:

Stepwise Variable Selection:

Doesn't check all combinations

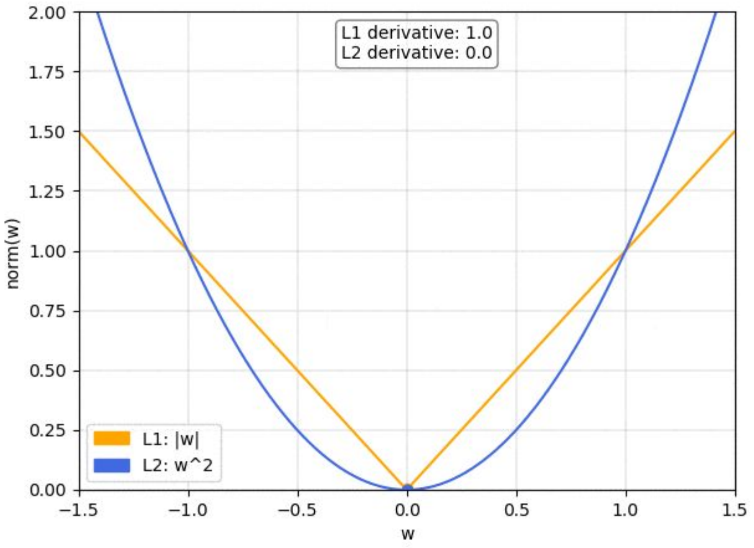
Future added variables could invalidate previous ones



Variable Selection - L1 Regularization

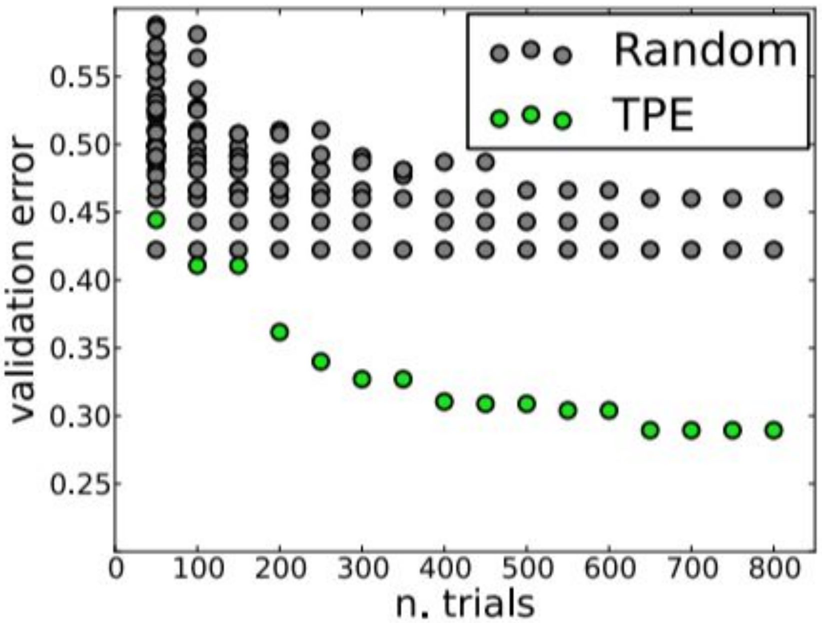
$$J(\alpha, \beta) = \frac{1}{n}\sum_{i=1}^{n} -y_i(\alpha + \vec{x}_i\vec{\beta}) + log(1 + e^{\alpha + \vec{x}_i\vec{\beta}}) + \lambda \sum_{j=1}^{p} \beta_j^2$$

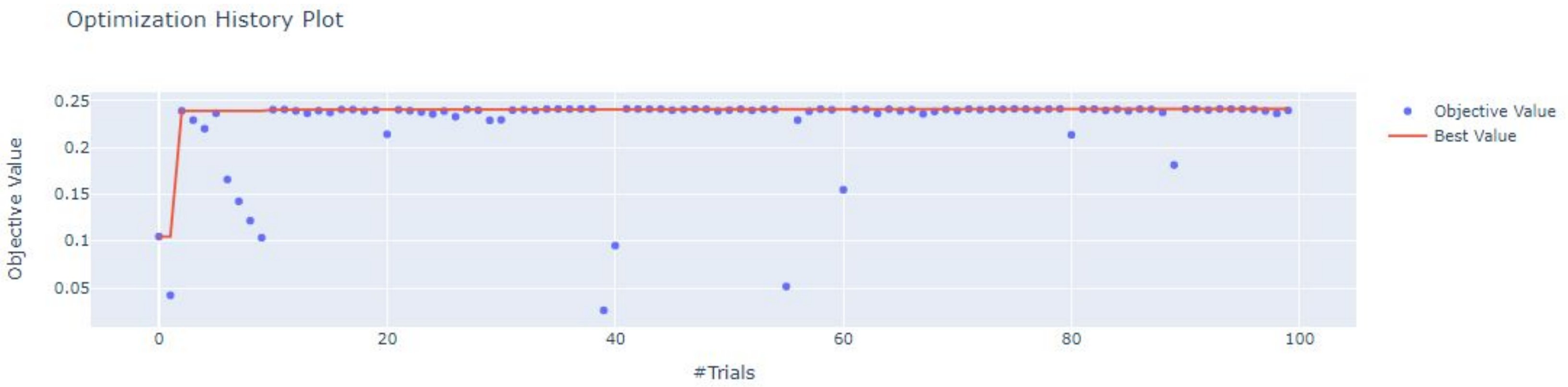
- L2 steps sizes decrease as we converge
- L1 steps sizes constant
- Some coefficients go to 0
- Logistic Regression w/ hyperparameter tuning

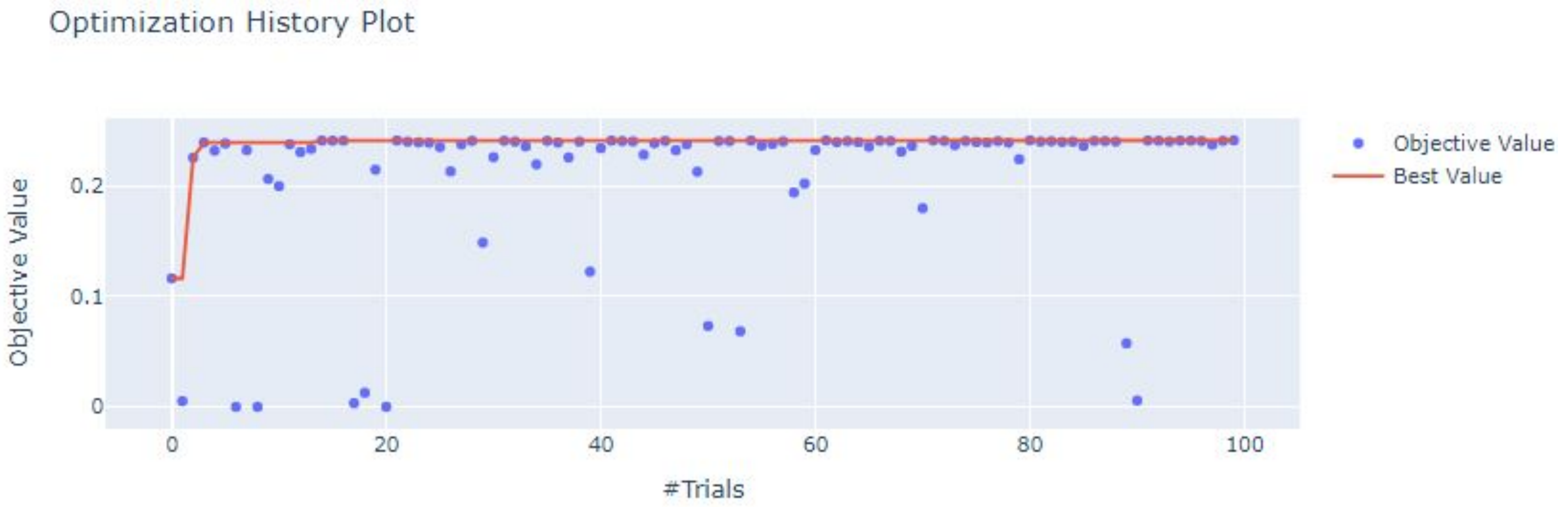


Hyperparameter Tuning

- RandomSearch: isn't exhaustive
- GridSearch: searching all combinations is inefficient
- Both don't learn from past trials
- Bayesian Optimization (TPE) uses past trials to determine new values to try




Optimization History Plot
Objective Value
0.25
0.2
0.15
0.1
0.05
0
20
40
60
80
100
#Trials
Objective Value
Best Value


Optimization History Plot
Objective Value
0.2
0.1
0
20
40
60
80
100
#Trials
Objective Value
Best Value

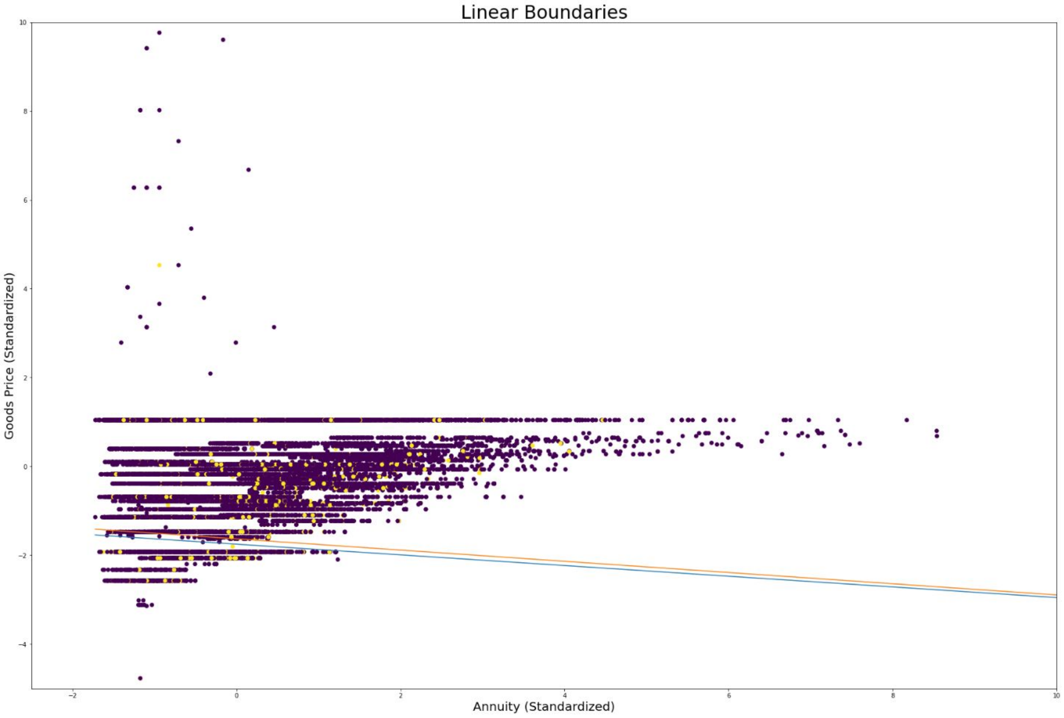
Best Class Weights Model

Final lambda = 0.02775159018992187
Final class weight = 7.839529551742979.

	Perceptron	Logistic Regression
F1 Score	0.2408	.2413
Accuracy	0.7884	.7968

Decision Boundary Plots

- Blue - Perceptron
- Orange - Logistic Regression
- SVM failed a lot
- Multiple different variables tried
- Perceptron and Logistic Regression very similar

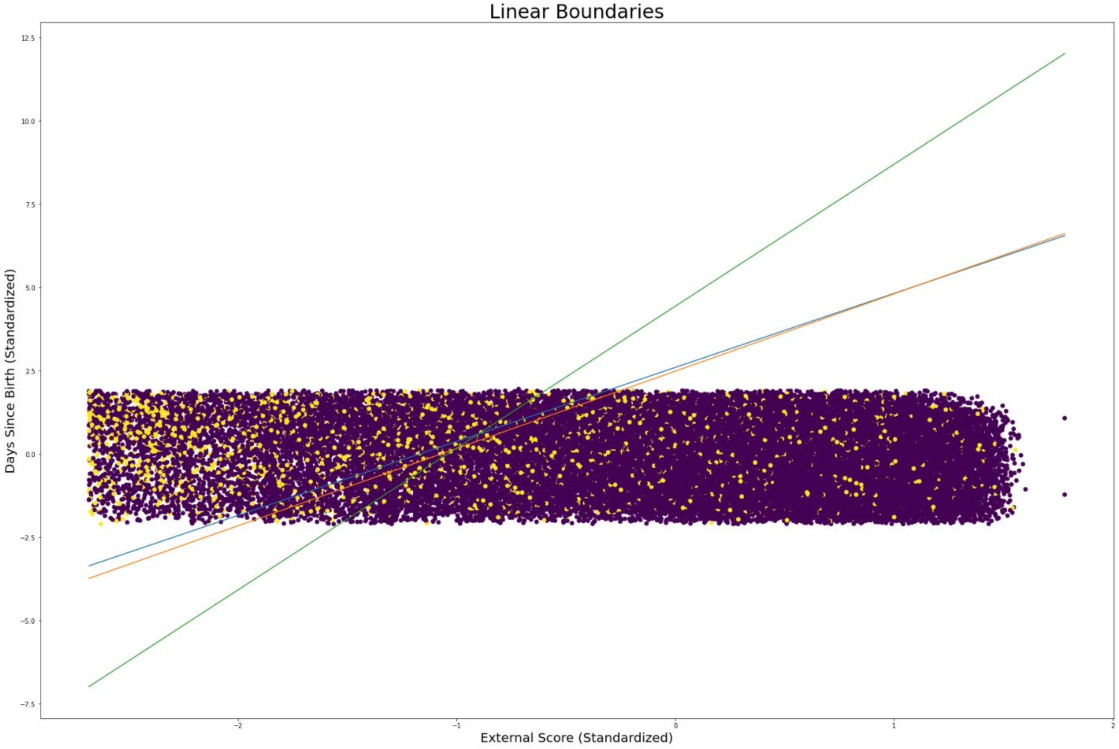


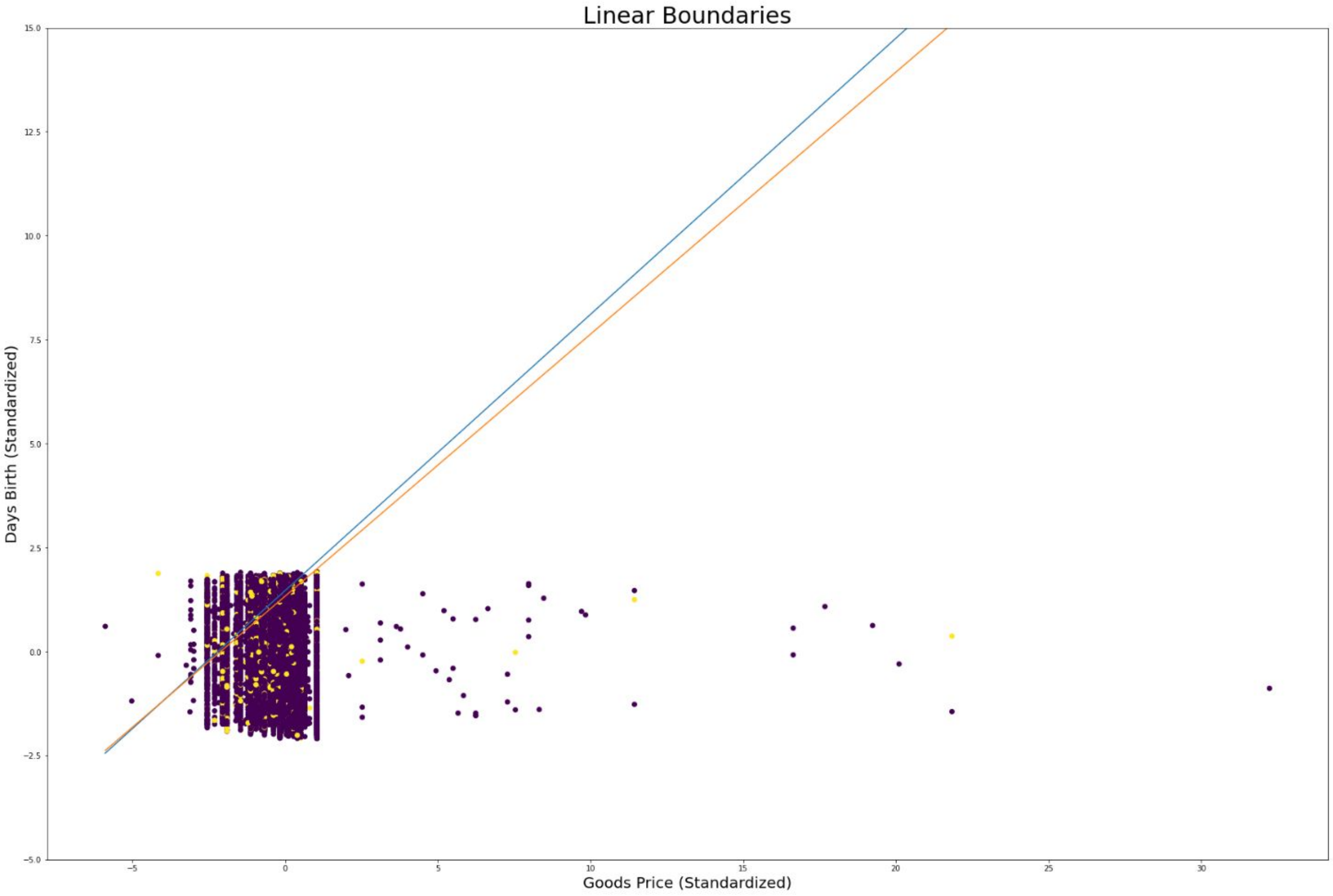
Decision Boundary plots

Blue - Perceptron

Orange - Logistic Regression

Green - SVM





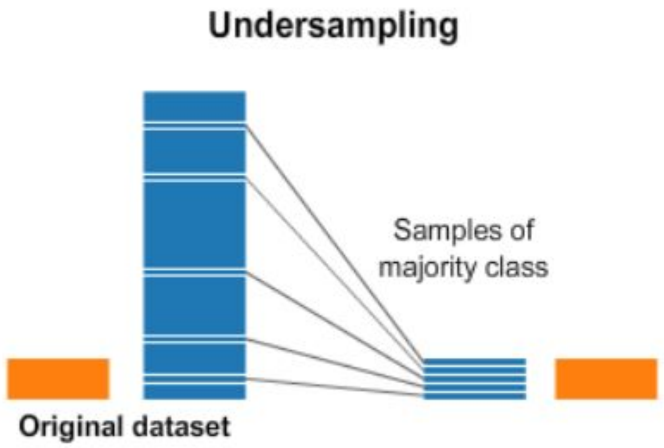
Linear Boundaries
Days Birth (Standardized)
Goods Price (Standardized)
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−5
0
5
10
15
20
25
30

Own Implementations

Didn’t add class weights to our implementations

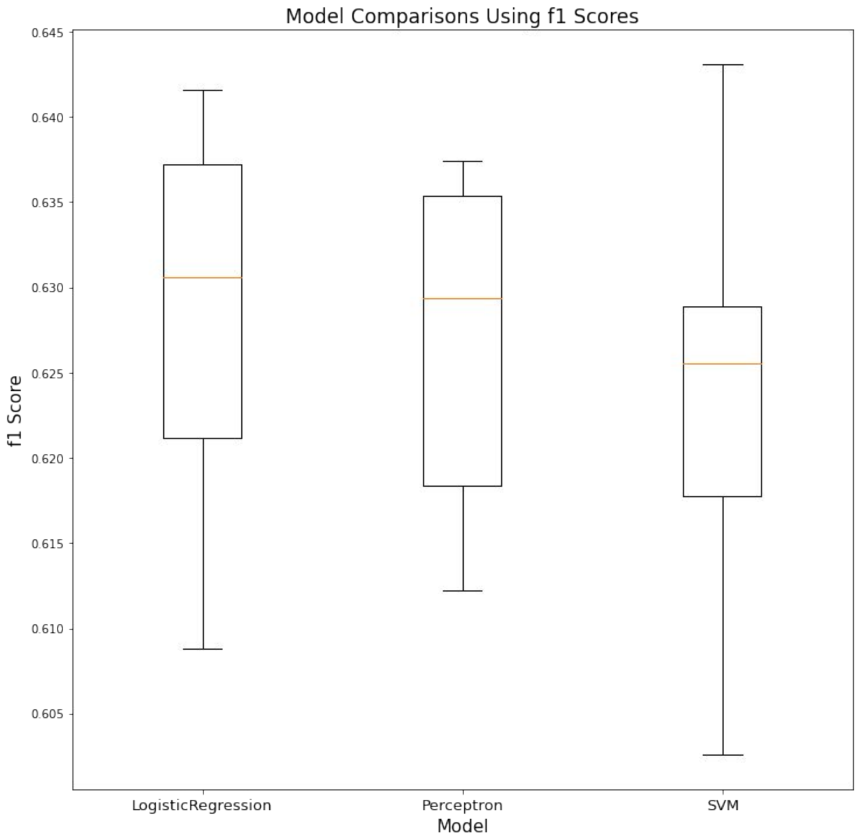
Used sub-sampling with balanced target labels

Trained on sub-sample then predicted on entire dataset



Model Comparison

- Cross-validation F1 Scores when trained on sub-sample of data with balanced target labels
- Calculate F1 scores after predicting on entire data



Final Model

- Perceptron
  - 16 features
  - Lambda = 0.108

	Logistic	SVM	Perceptron
Accuracy	0.755	0.751	0.790
F1 Score	0.397	0.393	0.435

```
(-5.786649851087536e-17,
 array([-0.00076532, -0.00335159,  0.00281297, -0.00329937,  0.11590957,
        -0.605524  , -0.00666676, -0.00420152, -0.00246973, -0.09627809,
        -0.00184908, -0.05588842,  0.01078155,  0.00664245, -0.00429602,
         0.02765359]))
```

Ethical Considerations

- Other columns that could be excluded like unemployed, because it is unfair
- Excluded gender column
- We wanted to exclude gender from our model because we didn’t want our model to discriminate against the basis of gender
    - But maybe there are inherently different practices (though unethical) that incorporate gender to make predictions?
- We could have also excluded unemployment and other variables for the same reason
    - Though unemployment

Reflection

- We could’ve used domain knowledge from a real domain expert
- Despite the unbalanced data, we dealt with it the best we could using class weights or balanced sub-sampling
- Our F1 scores are too low to recommend usage of these models