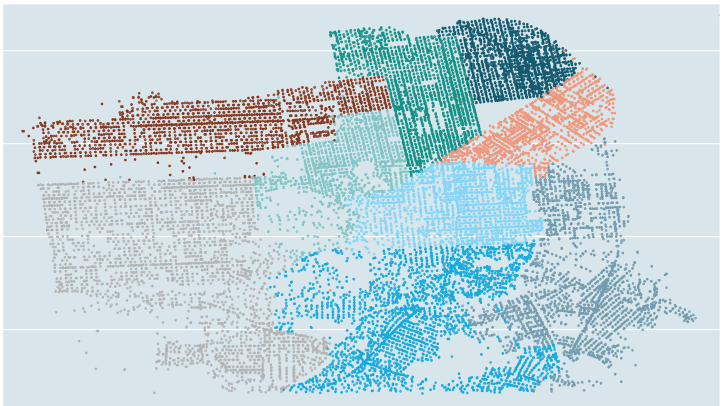
University of British Columbia

Crime Classification Through Supervised Learning

Alvin Nursalim, Deonne Millaire, James Chuang, Michael Yuen

Motivation



- In this **Application Bake Off**, we compare various Machine Learning algorithms and their abilities to **predict crime based on time and location**
- We chose this **Kaggle** dataset since we wanted to see the limitations of ML as the best log loss in this competition was only 1.95936
- We are going to experiment various generative and discriminative models, and ensemble learning methods
- We expect the ensemble models to outperform the rest of the algorithms

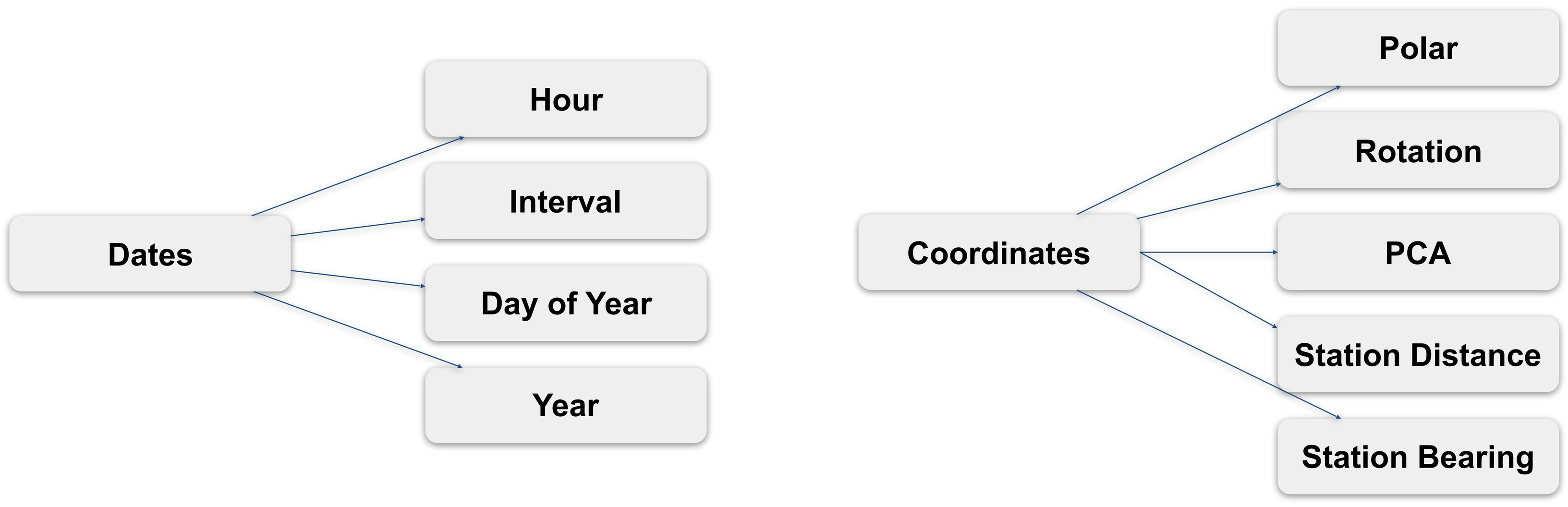
Why Logarithmic Loss?

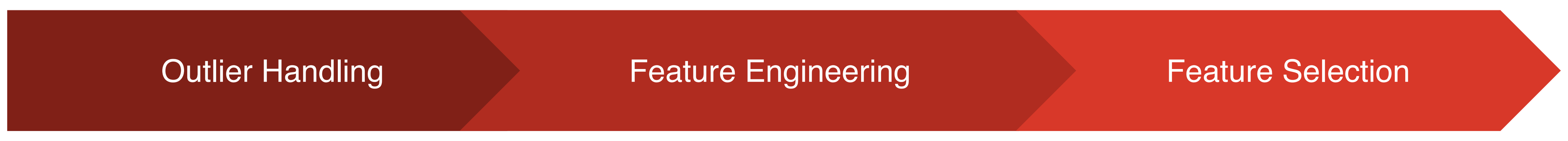
$$\text{Logarithmic Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

N is number of observations
M is number of class labels
y_{ij} is indicator for correct classification
p_{ij} is prediction probability

- Log loss metric provides us with an uncertainty measure of the predictions
- Penalize the classifiers each time they predict correctly with low certainty or incorrectly with high certainty

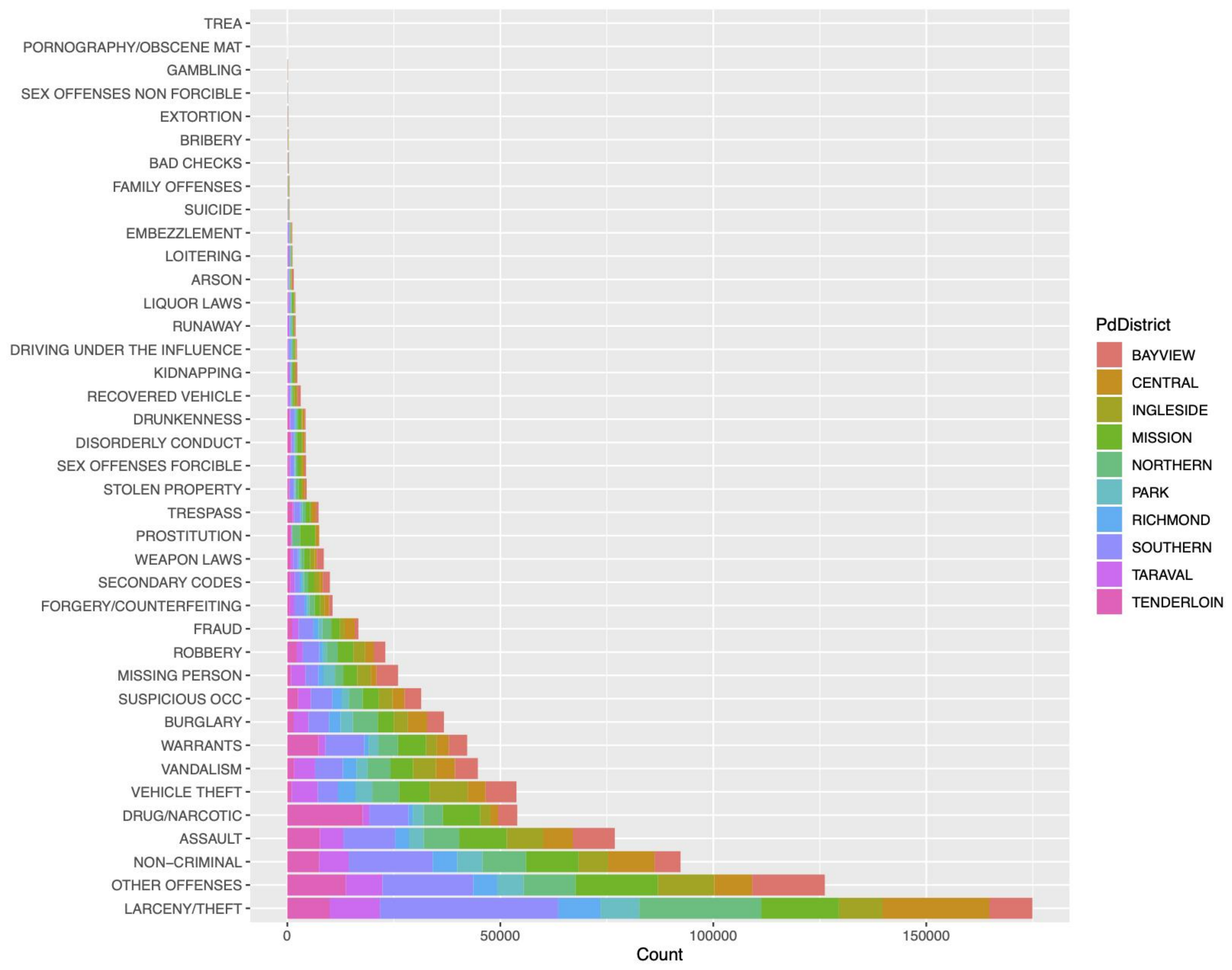
Pre-processing





- Data contains 7 features and 39 class labels
- Training set contains 878049 observations and test set contains 884262 observations
- Outlier handling using address matching and mean coordinates by district
- Intersection indicator feature based on Address
- 1-of-k encoding for Police Department Districts
- Cyclic representation for Hour, Day of Week, and Day of Year

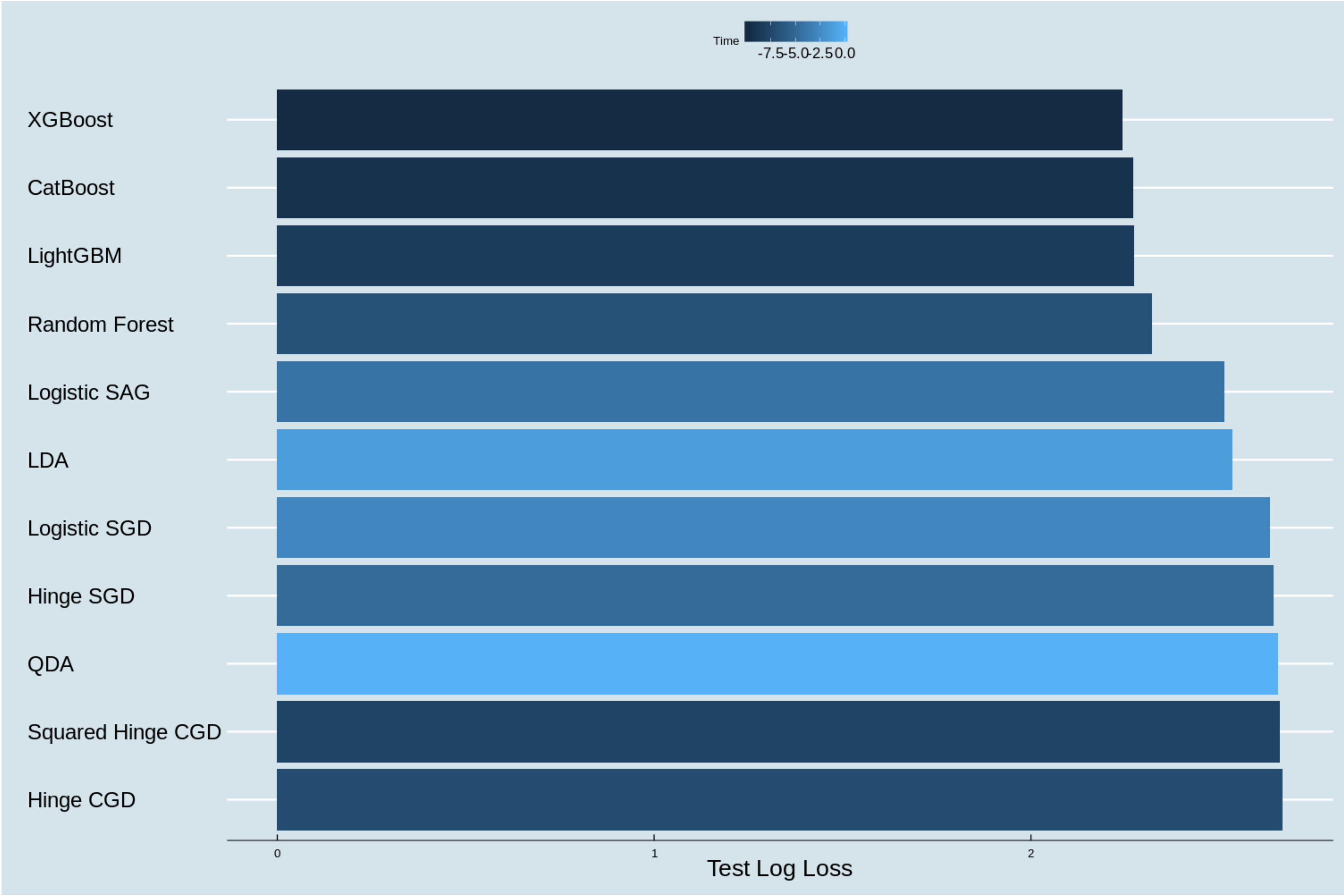
Crime Distribution



Contribution

- In the end, we concluded that there are no current algorithms that can accurately predict unbalanced multi-class datasets surrounding the crimes in San Francisco
- This approach can be useful to the San Francisco Police Department, helping increase efficacy of resource allocation and maximizing response time
- The data collected may be insightful for tourists to avoid areas with a higher crime rate
- Can aid district officials who are looking to enforce crime-related policies

Bake Off



Discussion

Results

- Our best classifier achieved a 2.24266 test log loss with the **XGBoost** Python library
- Standalone models were less performant, with a best test log loss of 2.51241
- Based on our results, we generalized that unbalanced multi-class datasets are best classified through ensemble learning, but with a cost of time
- Depending on resource availability and desired log loss score, there does not seem to be an ideal algorithm that can accurately predict unbalanced multi-class SF crime dataset

Setbacks

- One setback of this dataset was that the examples did not have a good representation of several crime categories (e.g. refer to Trespassing on the Crime Distribution graph), which in turn makes some labels hard to predict
- Our original dataset only provided us with time and location features, it has been shown that these features alone are not enough to predict crime accurately

What's next?

- In order for the dataset to be a better representation of crime categories, additional features like age, gender, or median household income of the culprit may be helpful rather than just crime location and time
- We would also like to include additional datasets from 2015-2019 and see if we are able to improve our prediction accuracy, as well as to confirm our initial model
- It would be interesting to experiment crime datasets from other cities in the United States to analyze if there are trends indicative of a larger pattern in crime

References

1. Anna Veronika Dorogush, Vasily Ershov, Andrey Gulin "CatBoost: gradient boosting with categorical features support". Workshop on ML Systems at NIPS 2017.
2. Pedregosa *et al.*, Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.
3. Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In 22nd SIGKDD Conference on Knowledge Discovery and Data Mining, 2016
4. Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". Advances in Neural Information Processing Systems 30 (NIPS 2017), pp. 3149-3157.
5. Kaggle: https://www.kaggle.com/c/sf-crime/

Check out our code!