Predicting Service Level Agreement (SLA) Violations for NYC 311 Service Requests

Kim Sha (ks966@cornell.edu), Subin Yun (sy464@cornell.edu) | Project Folder
CS 5785: Applied Machine Learning | Fall 2023

1 ABSTRACT

Predicting Service Level Agreement violations for NYC 311 service requests allows for more efficient and responsive urban governance. City agencies can use these predictions to triage urgent incidents, offering a pathway for optimizing municipal resources. We develop gradient-boosted decision trees to achieve this in both agency-agnostic and agency-specific settings. Both approaches yield high performance levels in predicting SLA violations, underscoring the potential of machine learning in transforming public service operations.

2 INTRODUCTION

Timely response of city agencies to 311 service requests is an integral part of a city's commitment to its residents. NYC residents rely on 311 to report non-emergency issues, which can range from noise complaints to infrastructure problems. The Service Level Agreement (SLA) is a period of time defined by the assigned agency to provide the submitter with the expected time until an update or final resolution of the request. When agencies fail to meet SLA commitments, it affects the quality of life for residents and their trust in public services.

This project develops models to predict the likelihood that a NYC 311 service request will be *closed* by the resolving agency within the period of time defined by the SLA. The city can use this to proactively allocate resources, prioritize pressing issues, and ensure smooth operations - ultimately to improve resident satisfaction.

3 BACKGROUND

Residents from all 5 boroughs in NYC can submit 311 services requests. These records are made available daily through the NYC Open Data portal.

I. Dataset

This project relied on merging 2 datasets on 3 shared fields: agency, complaint type, and problem description:

(1) 311 Service Requests from 2010 to Present: provides 311 reporting as far back as 2010, and is updated on an automated daily basis. Each record in this dataset corresponds to a 311 service request.

(2) 311 Service Level Agreements: Provides the time commitments that City Agencies have made to respond to their assigned 311 Service Requests.

To iterate with constrained compute, we work primarily on 2 samples drawn from cases completed in 2023:

A. **Agency-agnostic:** 250k sampled records stratified by agency in a 60:20:20 train-calibration-test split.

B. **Agency-specific:** all of ~70k requests serviced by the Department of Transportation (DOT) with a 75:25 train-test split.

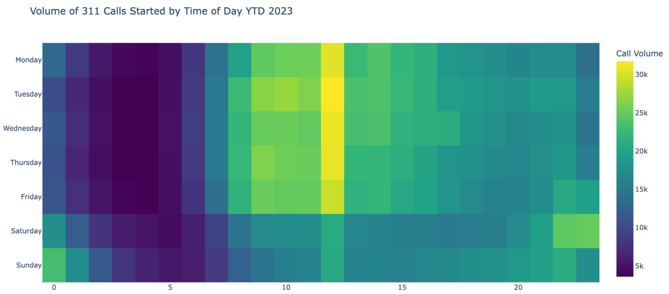


Figure 1. Heatmap of when 311 calls occur in 2023.

The set of features selected covered creation date / time (Fig. 1), assigned agency, coordinate location, and medium used to submit the request (phone, web etc.). It also included free text descriptions of the incident and high-cardinality categorical features such as address and location type. These were combined together into documents and transformed into embeddings. Some features were left out, such as a description of how the request was ultimately resolved, as such information is likely unavailable at inference time in practice.

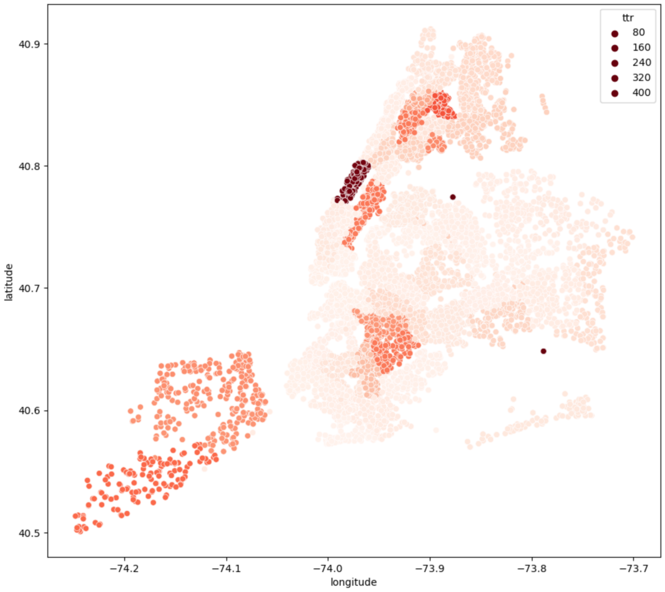


Figure 2. Time to resolve requests (in hrs) by geography.

The target variable is a binary indicator of whether the time taken (Fig. 2) to address each report exceeds and ‘breaches’ the corresponding SLA timeline. After analyzing the distribution of breaches across agencies to identify any trends or outliers that suggest issues with data quality, we ultimately narrowed to 9 agencies.

4 METHOD

We approach this problem as a standard binary classification task where the target labels indicate whether the agency in charge responded to the complaint in time (0) or violated their SLA (1). The primary target will be to develop an agency-agnostic model, which tackles this task for dataset A. However, we also present an example of an agency-specific model focused on the DOT and trained on dataset B. This approach acknowledges that different agencies may have distinct factors affecting their SLA compliance, which a single model might not capture effectively. We can use the DOT model to evaluate the necessity and feasibility of developing separate predictive models by agency. The following sections cover the overall methodology used to develop both models.

I. Document Embeddings

To encapsulate the semantic content of our textual data, we employed a document embedding technique leveraging the pre-trained 'en_core_web_md' model from spaCy, a popular natural language processing library. This model provides medium-sized word vectors set from the GloVe project, which represents words in a 300-dimensional vector space.

The construction of a document-level representation was achieved by concatenating various text fields to form a single document, before averaging the word vectors. The fields included are complaint type, descriptor, location type, community board, incident address, park facility name, bridge/highway name, and vehicle type.

The resultant vectors serve as a robust feature set for our predictive models, facilitating a nuanced understanding of the text's semantic properties, which is critical for the accurate classification of documents in our dataset.

II. Model Selection

Having defined our independent and corresponding dependent variables, we iterated through a series of standard supervised learning models. The initial strategy was to run some quick tests of models that are roughly increasing in complexity and predictive power, without doing any exhaustive hyperparameter tuning.

To that end, we started by evaluating the family of models with linear decision boundaries: logistic regression, linear discriminant analysis (LDA), and linear support vector machines (SVM). Then, we tested some non-linear models: k-nearest neighbors (a non-parametric lazy learner), random forests, XGBoost, and CatBoost (tree-based ensemble classifiers).

The latter 2 variations of gradient-boosted decision tree (GBDT) classifiers were ultimately selected for hyperparameter tuning. GBDTs combine ensembles of ‘weak’ learning shallow trees through boosting, where each tree is built sequentially to correct for prior errors:

XGBoost [1]: a variation of GBDT that’s also trained in a forward stagewise additive manner, but is optimized for computational performance and scalability. At each iteration t, we add a tree f_t to minimize the following L1 and L2-regularized objective function:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \ell\left(y_i, \hat{y}_i^{(t-1)}\right) + f_t\left(\mathbf{x}_i\right) + \Omega(f_t)$$

- $\ell(y_i, \hat{y}_i^{(t-1)})$: loss function (log-loss) which measures the discrepancy between the predicted value $\hat{y}_i^{(t-1)}$ and the actual label y_i for the i-th instance.
- $f_t(\mathbf{x}_i)$: score given by the t-th tree for the i-th instance with feature vector $\mathbf{x}_i$.
- $\Omega(f_t)$: regularization term for the t-th tree to penalize model complexity.

CatBoost [2]: a GBDT variation that is unique in constructing symmetric trees (standardizing splits at each level to the same feature), which has a regularizing effect to prevent overfitting.

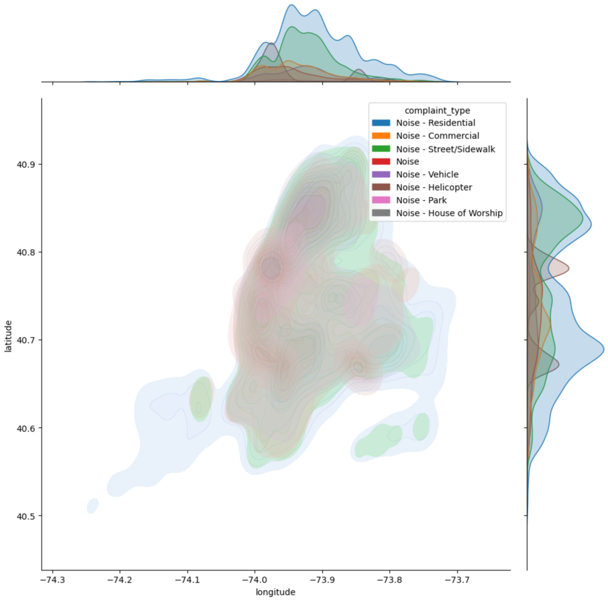


Figure 3. KDE visualization of noise complaints across NYC. To familiarize ourselves with the domain, we also did some exploratory modeling that is not directly related to the prediction task.

III. Development Pipeline

Quick iterations of the models described were compared by cross-validation on the training set to average evaluation metrics across the 5 folds.

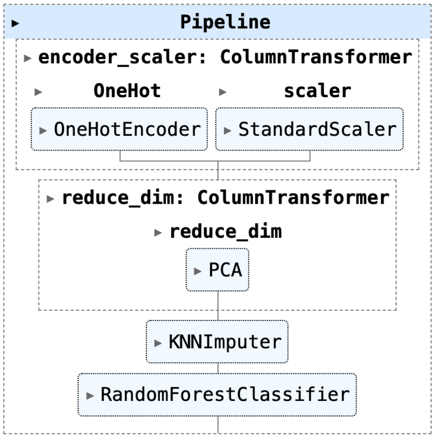


Figure 4. Visualization of the ML Pipeline.. The ColumnTransformer includes OneHotEncoder for categorical variables and StandardScaler for numeric feature scaling. PCA for dimensionality reduction is only applied to the document embeddings, which span 300 columns.

Most models (other than XGBoost and CatBoost) were also wrapped in a feature engineering pipeline (Fig. 4) consisting of 4 layers:

1) **encoder_scaler**: for encoding categorical features and scaling continuous features.
2) **reduce_dim**: applies dimensionality reduction techniques such as Principal Component Analysis (PCA) or t-Distributed Stochastic Neighbor Embedding (t-SNE) to manage the high-dimensional feature space that typically results from processing text, ensuring that the models remain interpretable and computationally feasible.
3) **KNNImputer**: to handle some missing values in geographic coordinates.
4) **Classifier**: the selected model that takes the previously transformed data as training input.

This pipeline ensures a systematic and reproducible approach to training the classification model.

While we briefly examined methods to address class imbalance, such as Synthetic Minority Over-sampling Technique (SMOTE) for oversampling the minority class, we found that weighting errors made on the minority (1) class more (‘scale_pos_weight’ in XGBoost) tended to be the most consistently effective.

After model selection, a search is implemented using 5-fold cross-validation to tune hyperparameters on the training set, before the best model is evaluated on the untouched test set. Conducting a gridsearch across a large search space is computationally prohibitive, so we opted for a Bayesian optimization approach (using HyperOpt). Hyperparameter tuning via cross validation negates the need for a dedicated validation set.

IV. Evaluation Metrics

We utilize a set of metrics to evaluate the performance of our predictive models rigorously. Accuracy measures the proportion of total predictions our model makes correctly. Accuracy alone can be misleading in this case, where the data is imbalanced. F1 scores provide a more nuanced view of model performance for the minority class. As a result, we use this metric as the objective to maximize during hyperparameter tuning. Finally, the Area Under the Receiver Operating Characteristic (ROC) and Precision-Recall (PR) Curves give us an understanding of the model's ability to distinguish between classes across all thresholds. In particular, AUC for the PR curve is less sensitive to class imbalance.

We also consider log-loss, which is defined as the negative log-likelihood of the true class labels given the predictions made by the model. Log-loss is particularly useful as it takes into account the uncertainty of the model's predictions, making it a robust metric for model calibration. This robustness is further assessed through visualizing reliability diagrams, ensuring that the model's probability outputs correspond well with the actual likelihood of an event occurring.

5 EXPERIMENTS

I. Baseline Modeling

The table below summarizes baseline performances for the *agency-agnostic* model:

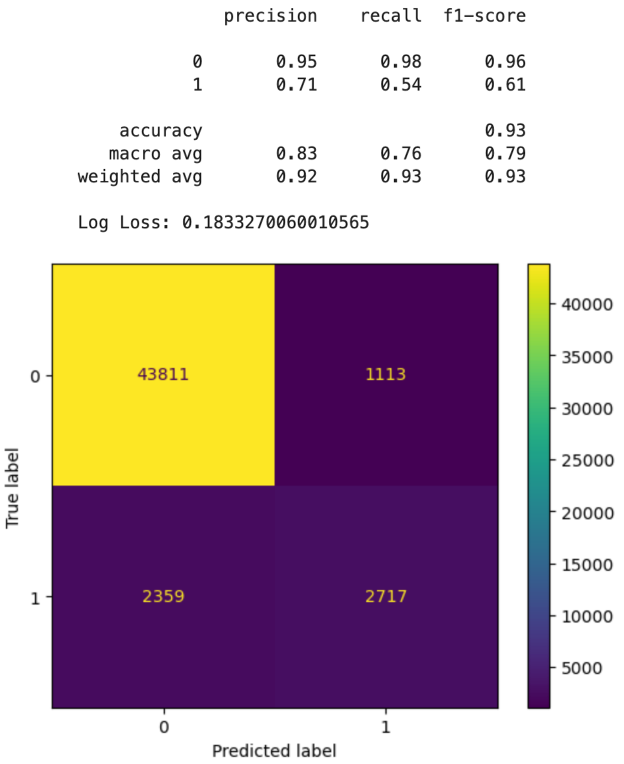
Model	Accuracy	F1	AUC-ROC	Log-Loss
Logistic Regression	0.902	0.269	0.830	0.252
LDA	0.895	0.439	0.823	0.282
Linear SVM	0.899	0.157	0.825	N/A
KNN	0.921	0.552	0.842	1.009
Random Forest	0.922	0.541	0.90	0.342
XGBoost	0.929	0.593	0.927	0.183
CatBoost	0.933	0.608	0.930	0.177

Table 1. Evaluation metrics *averaged* across 5 cross-validation folds

From these initial experiments, both the XGBoost and CatBoost classifiers were seen as good candidates to tune further. While we illustrate the agency-agnostic model with the former and the agency-specific DOT model with the latter, we saw in practice that the performance of the 2 classifiers tended to converge after extensive tuning.

II. Outcomes: Agency-Agnostic

Following hyperparameter tuning on XGBoost, we dived deeper into the model's performance on the 50k sample test set:



```
              precision    recall  f1-score

           0       0.95      0.98      0.96
           1       0.71      0.54      0.61

    accuracy                           0.93
   macro avg       0.83      0.76      0.79
weighted avg       0.92      0.93      0.93

Log Loss: 0.1833270060010565
```

Figure 5. Classification report and confusion matrix for the model. The matrix highlights the model's slight tendency to predict non-violations (0) more often than SLA violations (1), as evidenced by the higher number of false negatives (2359).

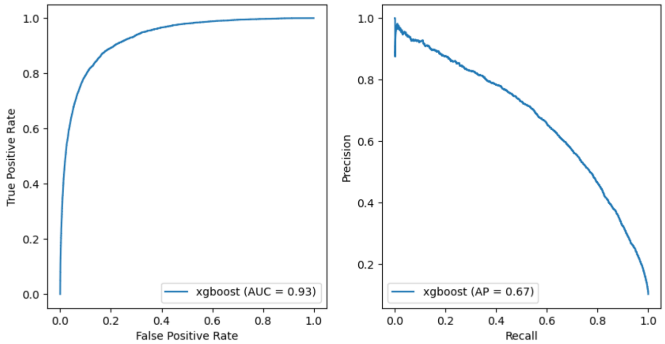
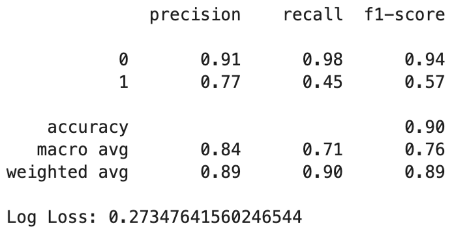


Figure 6. ROC curve (left) and Precision-Recall curve (right) for the XGBoost Classifier. AUC-ROC of 0.93 indicates a high level of predictive accuracy for the classifier. The PR curve illustrates the tradeoff of capturing more of the positive class (increasing recall), which also increases false-positives (decreasing precision).

III. Outcomes: Agency-Specific (DOT)

Applying CatBoost to the DOT's 311 requests yielded the following performance on the ~17.5k sample test set:



```
              precision    recall  f1-score

           0       0.91      0.98      0.94
           1       0.77      0.45      0.57

    accuracy                           0.90
   macro avg       0.84      0.71      0.76
weighted avg       0.89      0.90      0.89

Log Loss: 0.27347641560246544
```

Figure 7. Classification report for the agency-specific DOT model.

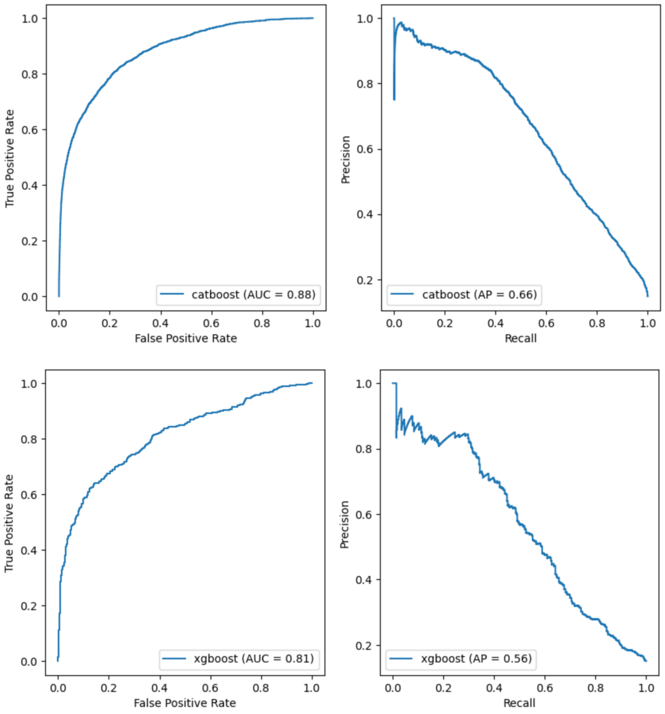


Figure 8. A comparison of the ROC and Precision-Recall curves between the agency-specific DOT model (top), and the agency-agnostic model from II (bottom), when filtered to DOT 311 service requests.

From Fig. 8, it became clear that the difficulty in generalizing across agencies resulted in a limited level of performance for the agency-agnostic model in specific use-cases.

IV. Error Analysis

Both the agency-agnostic and DOT models yielded at least 90% accuracy on their test sets. This indicates that the models are effective overall at classifying instances. In the same vein, a random-guessing predictor would yield an AUC-ROC of 0.5. The agency-agnostic/specific models both significantly outperform this bar, with scores of 0.93 and 0.88, respectively. While this suggests good discriminative ability, F1 scores and PR curves paint a more nuanced picture.

F1 scores for the agency-agnostic and DOT models are less impressive, at 0.61 and 0.57, respectively. In particular, our recall is low — indicating that the models are not as effective at classifying the positive (minority) class. The confusion matrices are in line with the conclusion, showing a relatively high false negative rate. This was expected, which is why our hyperparameter tuning focused explicitly on maximizing F1 scores. We still see the benefits of this, despite neither model achieving PR-AUC > 0.67. A random guesser would have a PR-AUC equal to the ratio of minority samples in the dataset. The agency-agnostic model outperforms this benchmark (~0.10) by 0.51. The DOT model outperforms this benchmark (~0.15) by 0.42.

Taken together, these metrics offer a comprehensive view: while our model correctly identifies most instances, it struggles more to correctly label the positive class. We mitigated this shortcoming by optimizing for the F1 score during hyperparameter tuning, which still leads all other model baselines (Table 1).

6 DISCUSSION

While we could not find any previous work that tackled this particular prediction task, it's inspired by the work of Liu et al. [3], which focused on a model to correct under-reporting in crowdsourced 311 service requests. The delay between an incident occurring and its resolution can be thought of as a combination of *reporting* delay and *servicing* delays. Liu et al. focused on the former, while this report investigates the latter. Several takeaways stem from this research.

We used an agency-agnostic approach as an illustration of model capabilities at scale. However, agency-specific models should be preferred over an agency-agnostic approach in practice. This yields both performance and scalability benefits (Fig. 8). There's also little reason why independent agencies would have any practical need for the flexibility of a cross-agency model.

Accounting for class imbalance by penalizing incorrect classifications on the minority class *more* was extremely effective in improving recall. However, city agencies may want not only to predict the class label, but also to obtain a *probability* of SLA violations. This would provide them with some kind of confidence on the prediction for triaging purposes.

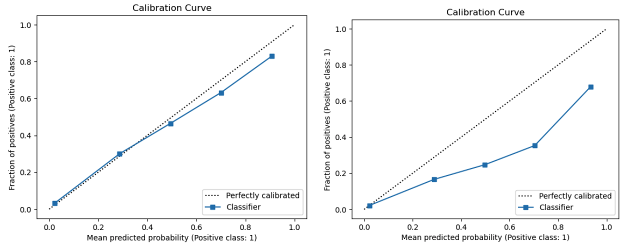


Figure 9. Our current well-calibrated model (left), and how scaling the minority class weight can result in poor calibration (right). Reliability diagrams plot the frequency of the positive label against the predicted probability of a model.

We quickly found there is a tradeoff between addressing class imbalance and ensuring well-calibrated predicted probabilities. Over-correcting for false negatives leads to overestimates of SLA violations, both in terms of false positives and inflated likelihoods of occurrence (Fig. 9). In practice, this may strain agency resources in a more wasteful manner than false negatives. Since this effort also didn't yield higher F1 scores, we present the well-calibrated model here.

As we iterated, we found performance improvements from many areas, such as more complex models, optimized hyperparameter tuning, and richer features from document embeddings. An often overlooked factor is simply using more training data. Doubling the training data from 75k to 150k for the agency-agnostic model not only improved metrics, but also smoothed out noise and reduced overfitting. In this sense, compute is king.

7 CONCLUSION

Our findings lay a strong foundation for future work in this domain. We proposed 2 models that perform well at this binary classification task. Further refinement and development could go in different directions:

Agency-Specific Data: research by Prof. Nikhil Garg on NYC 311 focuses on the Department of Parks and Recreation, which has some even richer text data from transcripts / free text description that could be leveraged for an agency-specific DPR model.

Calibrated Classifiers: pre-existing techniques to calibrate a classifier that has already been trained could be used on the dedicated calibration set.

Scaled Training: frameworks like XGBoost support distributed GPU training, which would allow us to fit a much larger and diverse training set.

Ultimately, the goal of enhancing the predictive power of our models is to provide actionable insights to NYC agencies, improve the timeliness of 311 service request responses, and boost resident satisfaction.

8 REFERENCES

[1] Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, https://doi.org/10.1145/2939672.2939785.

[2] Liu, Zhi, et al. "Quantifying spatial under-reporting disparities in resident crowdsourcing." Nature

Computational Science, 2023, https://doi.org/10.1038/s43588-023-00572-6.

[3] Prokhorenkova, Liudmila, et al. "CatBoost: Unbiased Boosting with Categorical Features." ArXiv.org, 20 Sept. 2019. https://arxiv.org/abs/1706.09516.