**CS 677 Machine Learning**

**Spring 2025 / CRN 22928**

**Final Project Report**

**May 2, 2025**

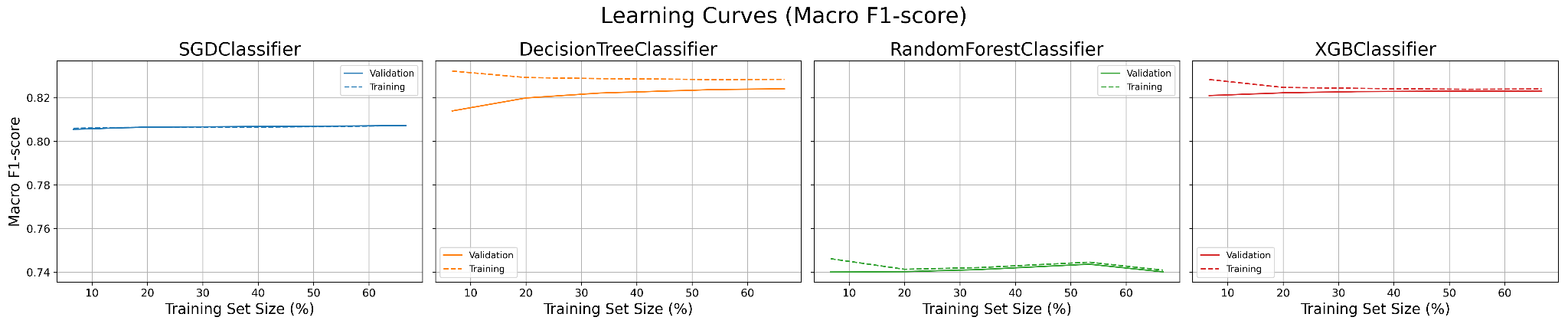
**Joshua Gottlieb, Alex Yoo, Aryan Lagalwar, Jinhwi Lee**

**Project Overview**

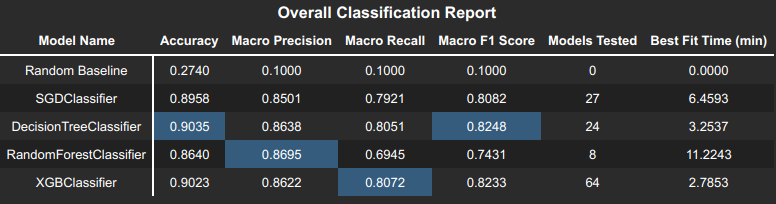
The 311 request line is a non-emergency phone number for filing requests and complaints in New York City (NYC311). Unlike 911 which primarily reaches the police department, ambulance services, or the fire department, 311 is not meant to report crimes and can be redirected to a wide variety of agencies. The aim of this project is to use the 311 service data from the year of 2024 to create a machine learning model that automatically predicts which agency should handle an incoming request (NYC Open Data). The 311 data for 2024 contains over 3 million records, giving ample data to use for training a machine learning model. This corresponds to a multiclass classification problem, as there are more than two agencies to choose.

Correct assignments to agencies are important to ensure requests are resolved and to prevent wasted time and resources. In order to evaluate model strength and to select models, it is necessary to choose an appropriate metric. In the 311 dataset for 2024, 46% of the requests are handled by the New York Police Department, with the next most common agency being the Department of Housing Preservation and Development which handles only 21% of the requests. There are large class imbalances, and for this reason, accuracy is not a reliable metric. In the context of this project, precision measures the ability of the model to correctly direct the caller to the correct agency. A low precision means that callers are erroneously directed to that agency and represents wasted time and manpower. Recall measures the ability of the model to correctly capture all of the callers should be directed to a given agency. A low recall means that callers are not being directed where they should be and represents low resolutions rates and effectiveness. Thus, both precision and recall are important, so the metric chosen to evaluate the models was the macro f1-score. The macro f1-score was used over the weighted f1-score because the macro f1-score treats each class as equally important, which is vital for a task with large class imbalances such as this one. However, it is equally important that the model is scalable and easy to train, so a secondary scoring metric was the total time required to train the model.

**Chosen Model**



Learning Curves for Tested Models



Classification Report for Tested Models

Four different models were tested: a logistic regression model using stochastic gradient descent (SGDClassifier), a decision tree model (DecisionTreeClassifier), a random forest model (RandomForestClassifier), and an XGBoost model (XGBClassifier). The best performing models were the decision tree and XGBoost. Both had very similar macro F1-scores and performed nearly identically in all metrics and across all classes. However, the XGBoost was faster to train at 2.78 minutes for training time versus 3.25 minutes for training time. This is because XGBoost can take advantage of GPU acceleration, whereas the decision tree cannot, at least not in the sci-kit learn implementation. The XGBoost also showed fewer signs of variance in the learning curve, as the training and validation curves had less spread. For these reasons, the XGBoost model was chosen as the best model for this project.

**Lessons Learned**

The biggest lesson learned from this project is the importance of data cleaning, feature selection, and feature engineering. The original 311 dataset had 41 columns, but most of these columns were unnecessary. Columns such as Vehicle Type or Taxi Pickup Location only apply to a small subsection of the data, which makes them very poor features for modeling a diverse number of complaints. Other columns such as Descriptor were not necessary because they mostly repeated information present in other columns, such as the Complaint Type. Cutting down on the number of features helps to reduce dimensionality and remove unnecessary noise from the dataset.

Cleaning the data was effective for reducing dimensionality, and it was especially helpful for boosting interpretability of the features and for making features that were strong at separating classes. The original data had 191 unique complaint types and 141 unique location types. During preprocessing, we created categories to group types together and ended up with only 9 complaint types and 6 location types. This made these features much easier to understand graphically and created some strong connections between complaint or location type and agency which is part of why the models performed so well. When one-hot encoding these features, rather than having hundreds of newly created features, only 13 new features were created which assisted the models in not overfitting on the data.

**References**

*DecisionTreeClassifier* [Documentation]. (n.d.). scikit-learn.<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>

NYC311. (n.d.). *About NYC 311*. NYC311. Retrieved May 1, 2025, from<https://portal.311.nyc.gov/about-nyc-311/>

NYC Open Data. (n.d.). *311 service requests from 2010 to present*. City of New York. Retrieved May 1, 2025, from<https://data.cityofnewyork.us/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9/about_data>

*RandomForestClassifier* [Documentation]. (n.d.). scikit-learn.<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

*SGDClassifier* [Documentation]. (n.d.). scikit-learn. <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html>

*XGBClassifier* [Documentation]. (n.d.). XGBoost.<https://xgboost.readthedocs.io/en/latest/python/python_api.html#xgboost.XGBClassifier>