Houston 311 Service Requests

Stephen Huang, Levi Villarreal, Joshua Wong, Andrew Young

Problem Statement

A problem with 311 service requests is the common delays in addressing requests with public resources. With some service requests not being resolved until well over a month after their initial due date, we wanted to investigate the types of requests that are routinely delayed, and with further statistical analysis, determine possible actions that could be taken to reduce the delay. Since Houston is a very large city, we also wanted to see if the location from which the request was made played a factor in the tardiness. Our end goal was to be able to provide the citizens of Houston with an accurate way to see how long their service would take, based on historical service request data.

Data

The dataset consisted of 399,953 service requests made to 311 in the Houston area. Each record included information about the request, including various location attributes, the type of request, the time it was made and closed, and the trash/recycling information of the requester. Each request is assumed to be independent of other requests.

The features that we trained the models on were:

- *County* - The county of the service request.
- *District* - The geocoded council district of the service request.
- *Neighborhood* - The neighborhood of the geocoded service request.
- *Management District* - The management district of the service request location.
- *Department* - The department the service request is directed to (could be virtual department, e.g. 311 Help Line)
- *Division* -The division the service request is directed to (e.g. Knowledge)
- *Service Request Type* - The service request type.
- *Queue* -The queue the service request was put into.
- *SLA* - The Service Level Agreement (SLA) days. The number of days the department has from the create date to address the service request.
- *Service Request Month* - The month that the service request was created.
- *Channel Type* - How the ticket came in.

And the label for the data was

- *Overdue* - If the number is negative, it means the number of days the service request was completed before the SLA was due (or, how many days are remaining to the SLA if the case is still open). If positive, it means the number of days the service request was complete after the SLA was due (or, how many days past the SLA if the case is still outstanding).

Method

Many attributes were removed from the data set due to their irrelevance or redundancy to the analysis and to reduce the correlation between attributes. The following attributes were removed:

- *Case number* - This was dropped because the case number is a unique identifier assigned to every 311 requests, and thus could not be used to classify anything.
- *Trash Quad* - Most types of requests that are not related to trash collection. Those requests might be misclassified because they are in a different trash quad as other requests in the dataset.
- *Recycle Quad* - Same reasoning as Trash Quad
- *Trash Day* - Same reasoning as Trash Quad
- *Heavy Trash Day* - Same reasoning as Trash Quad
- *Recycle day* - Same reasoning as Trash Quad
- *SR Location* - Same reasoning as the case number.
- *Tax ID* - This ID is unique to the property that requested it.
- *Key Map* - This attribute is used in the cities internal system to keep track of requests and does not have any actual relation to the request
- *Due Date* - We are interested in predicting overdue-ness, and that is its own separate column.
- *Date Closed* - Same reasoning as the due date.
- *Title* - This is another unique identifier assigned to every 311 requests, and thus could not be used to classify anything.
- *x* - We already have latitude, and this attribute an approximation of latitude.
- *y* - We already have the longitude, and this attribute an approximation of longitude.
- *latitude* - This is too specific for our needs, and we have other location metrics.
- *longitude* - same as the latitude

After removing unnecessary attributes, requests that were not closed at the time the dataset was created were removed as we are trying to predict how long each request is overdue at the time of completion. The "Status" attribute was subsequently removed. We also cleaned certain attributes, such as "County" and "District," where there were duplicate entries that are not exact

copies. We then dropped all records that had at least one attribute with a NaN value, as it would be difficult to accurately fill in these values based on other requests.
We then used the "SR Create Date" to determine the month in which the request was made. This attribute is our only temporal attribute other than "overdue" and would be used to see if seasonality has anything to do with how much a request is overdue.

The class label "Overdue" was then binned for classification models. We chose to bin the data into time tasks (negative values), tasks done within a week past the due date (0 < x < 7), tasks done within a month past due date (7 <= x < 30), tasks done over a month past due date (x >= 30).

Most of the features were categorical, so we encoded them into numerical values in order for sklearn to process them correctly. Furthermore, we downsampled the dataset in order to speed up our training time.

Seven classifiers (decision trees, SVM, neural networks, k-nearest neighbors, Naive Bayes, AdaBoost with decision tree base, and random forest) were then tested on the dataset for accuracy. The strong correlation between certain chosen attributes caused the Naive Bayes classifier to perform poorly, so the final voting classifier was constructed with SVM, neural networks, k-nearest neighbors, and random forest.

Challenges

The biggest challenge was working with attributes that had user input as the data. Thankfully there were not too many variations of the data to account for, so it was a matter of determining the variations and standardizing the data. An example of this was the ‘COUNTY’ attribute, where some of the input was ‘HARRIS’, ‘Harris’, or ‘Harris County’, and this had to be corrected for the models to treat them as from the same county.

Furthermore, one problem we had was with class imbalances, as most of the data was on time, we had to adjust for that, and not just look at accuracy, but precision and recall as well.

Results

The Random Forest classifier performed the best with an accuracy of 0.65. Since our dataset consisted mostly of categorical data, it would make sense for the Random Forest classifier to be the most accurate, since they handle categorical data well, due to the fact that the splits directly separate the categories. Furthermore, they are usually better when dealing with features that are highly correlated, which could be the case for our features related to the region.

With four different classes, the models produced fairly mediocre results. While all models produced results better than guessing (25%), some faired better than others. Naive Bayes performed poorly with an accuracy of 36% due to the correlation that existed between attributes, and k-nearest neighbors had an accuracy of 60% with the dataset having 11 attributes. The random forest classifier had an accuracy of 65%, the highest of the models. This result is expected due to the attributes being largely categorical. The voting classifier had an accuracy of 64%, but is probably more robust than the preceding models.

Most models had high precision for the on time class, while not being very precise for other classes. The models also exhibited poor recall for all classes, suggesting that other factors were at play in causing service requests to be delayed. The models are good at classifying on-time requests in part due to the significant class imbalance that is only partially mitigated with SMOTE, and the model has a hard time determining how late a late request will end up being.

Next Steps

Further statistical analysis would allow conclusions to be drawn from the data to supplement our predictive models. The model included only trained on 0.5% of the available data, so more data could be used to train the model to improve the accuracy of the model up to a certain point. However, an accuracy of 64% is fairly reasonable given there being four different classes and a variety of other factors that could play a role in delaying the completion of a service request.

Furthermore, this model could be used to provide citizens of Houston with an accurate measure of how long their service request would take to complete. For example, a citizen could enter in their service request information online on the 311 portal, and the model would run that data point through a model containing recent fulfilled service requests. This would improve the accuracy of the time to completion prediction given to the citizen, and make the process of filing a service request much more reliable.