Predicting CS106 Office Hours Queuing Times

Brahm Capoor | Nick Troccoli | Michael Troute

Stanford University | CS229 | December 12, 2017

Abstract

We apply machine learning to predict the time a student at CS106 office hours will wait for help, and the time a helper will take to resolve the student's problem. We frame the prediction task both as a multi-class classification problem (applying logistic regression and neural networks) and a regression problem (applying neural networks). For classification, the neural network achieves the greatest wait time accuracy and logistic regression achieves slightly higher help time accuracy. For regression, the neural network predicted wait times with a mean squared error (MSE) of 1756 and help times with an MSE of 387.

Introduction

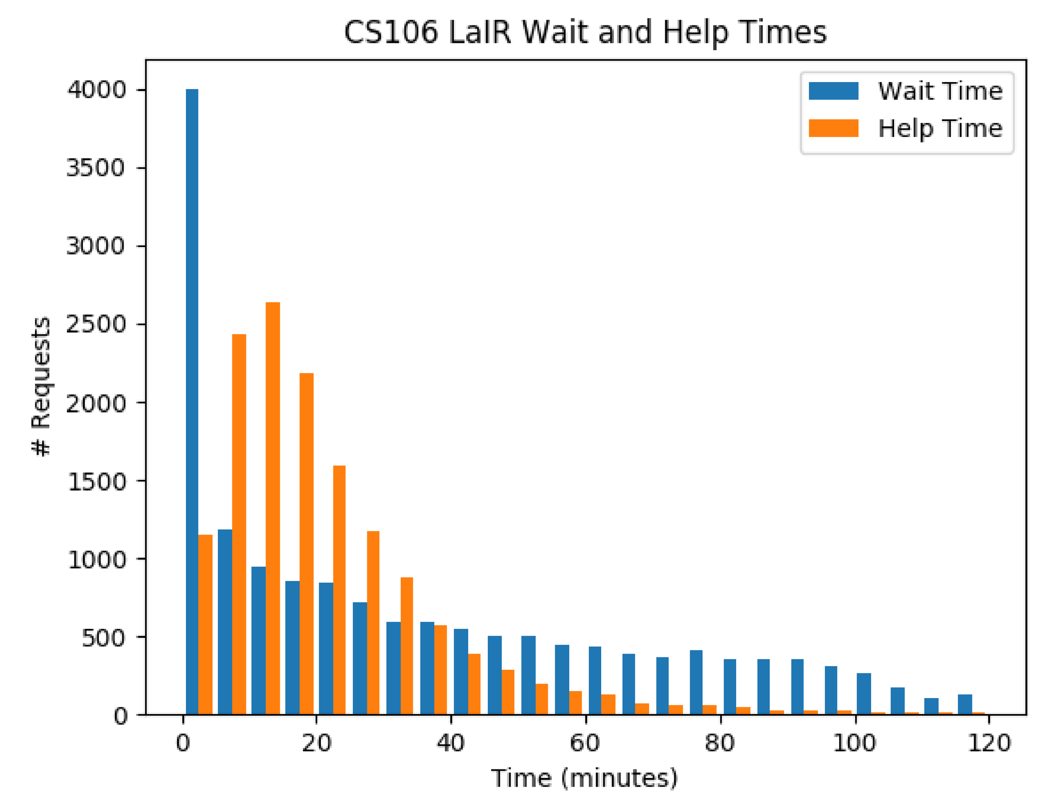
Each year, thousands of students enroll in Stanford's introductory CS106 computer science classes. Wait times for students at office hours ("the LaIR") can be significant, and are impacted by many factors, such as the student's question, day of week, staff on duty, and more. Accurately estimating wait and help time is important because students can better plan their time, and course staff can know in advance the time required to resolve outstanding requests. Further, analyzing the trained model could provide insights into what factors most impact help times.

Problem Statement

To predict CS106 office hours wait and help times by building machine learning models that featurize help requests and characteristics of the LaIR (such as helpers on duty).

Datasest

Students at the LaIR fill out a help form, which logs requests in a database. We exported data for the past 4 quarters (15,914 requests). Each request has an anonymized student identifier, their course, the problem description, the identifiers of the helpers on duty when the request was made, the request's date and time, and its wait and help time.



Vectorizers

We chose features from requests that we believed strongly influenced wait and help times, and created vectorizers that we apply to the requests before feeding them into our models. These vectorizers include: the request description, vectorized using TF-IDF, and its length, information about previous requests that day, the student's course, the helpers on shift at that time, the date and time of the request, and information about the student's previous LaIR visits.

Regression

We trained a neural network each to predict wait and help times; they are each comprised of 10 size-500 hidden layers with ELU activations, trained with Stochastic Gradient Descent (SGD) with a batch size of 25. Our learning rate optimization results are below.

Learning Rate vs. Training MSE

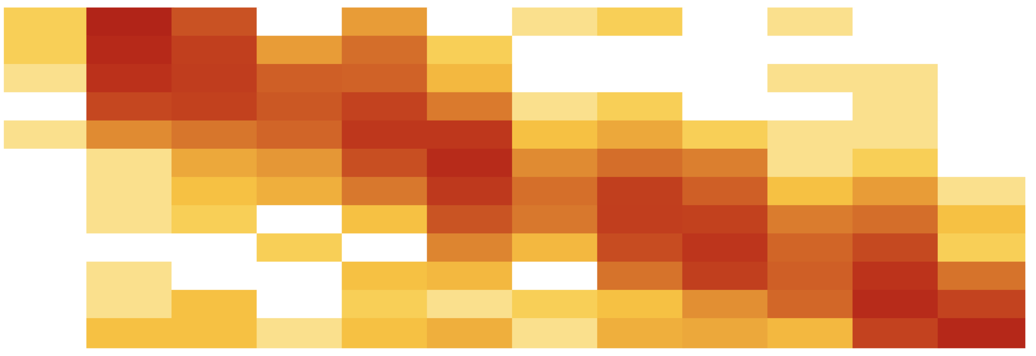
Learning Rate	Wait Model	Help Model
0.005	2243.29	446.23
0.01	1850.49	388.84
0.02	1756.73	387.14
0.03	1756.09	387.15
0.04	1756.09	387.17

With the best learning rates found above, after 100 epochs we obtained a MSE for help time of 342.27 and an MSE for wait time of 1743.75.

Classification

We trained neural network and linear regression models with various bucket sizes for times. We used an "equiwidth" approach with buckets of equal time size, and an "equidepth" approach with buckets containing the same number of requests. The confusion matrices below show the potential advantages of the equidepth approach to combat dataset skew.

Equiwidth Confusion Matrix for Neural Network Wait Model



Equidepth Confusion Matrix for Neural Network Wait Model

The results of our experimentation with bucket sizes are shown in the following figures, assuming a maximum time of 120 minutes:

Bucketing vs. Accuracy % for Logistic Regression

	# Buckets	Help Dev. Acc.	Wait Dev. Acc.
Equiwidth	24	19.353	32.956
	12	33.416	43.763
	8	44.922	53.377
	6	59.091	59.566
Equidepth	24	4.581	16.4
	12	9.553	31.794

Bucketing vs. Accuracy % for Neural Network

	# Buckets	Help Dev. Acc.	Wait Dev. Acc.
Equiwidth	24	18.679	35.595
	12	33.736	49.545
	8	43.146	58.216
	6	57.919	65.567
Equidepth	24	4.581	16.588
	12	9.091	33.459

Comparison

We believe overall our classification approach outperforms our regression approach. The regression model predicts help times, on average, within $\pm$ 20 minutes while the best classification model correctly predicts 60% of help times inside a 20 minute bucket. The regression model predicts wait times within $\pm$ 42 minutes while the best classification model correctly predicts 65% of wait times inside a 20 minute bucket. We also tested other regression algorithms such as linear regression, which tended to perform abysmally owing to the nonlinear nature of our data. So far, our results show that logistic regression slightly outperforms Neural Networks in predicting help times whereas neural networks outperform logistic regression in predicting wait time. However, we believe with additional tuning of architecture and hyperparameters, Neural Networks can surpass logistic regression for predicting help times as well.

Conclusion

Classification is more successful at predicting LaIR wait and help times than regression; with further parameter and architecture tuning, Neural Networks will likely be the most successful algorithm overall.

Future Work

We see three avenues of further work. First, we plan to adjust our classification loss function to better reflect gradation of error across various classes for a particular request. This is because our buckets reflect time, and so buckets that are closer to truth are less incorrect than buckets that are further from the truth. Secondly, we seek a more rigorous way to compare the performance of regression and classification models. Finally, we hope to perform feature selection to identify features that most strongly indicate wait and help times.