Charlottesville Parking Ticket Predictor

Repo - https://github.com/JulianKrese/DS4002-CS3-Ticket-Predictor

Context

The year is 2024 and you're a student at UVA and just entered your 2nd Year. You ran the race and secured a great lease off grounds, but unfortunately it doesn't include parking. None of your friends have cars and you're going to need a car to enjoy all Charlottesville has to offer, as well as the more mundane tasks like getting groceries. No problem, there's plenty of street parking! Or is there? It's a mad scramble to find spots as many other students have the same idea as you. Unfortunately, at some point you'll have to briefly park in an illegal zone to wait for a real spot to open up. Taking chances like these means feeling the stinging pain of a ticket eventually.

Coming from a data science background, wouldn't it be helpful if you could actually put the skills you've learned to use? You realize if you could get a hold of some quality data, you could train a model to predict your odds of getting a ticket for a given location and time. Looking online, you notice that Charlottesville actually publicly stores anonymized ticket data at the City of Charlottesville Open Data Portal. With this data, you should be able to train a model to help you avoid tickets– hooray!

Deliverables

One issue with creating a parking ticket predictor is that infrastructure changes. Roads appear and disappear, and the same happens with parking spots. Given that the data is from a long period of time, you can't expect the locations to remain the same. After considering this, you think of even more factors at play– the time of day for tickets being given, the amount of workers ticketing, special events such as football games, and the list goes on. You realize you will need to create multiple models trained on multiple segments of time, hoping that one time period will generalize the best to the current trends in ticketing. You will want to test on the most recent data for the most accurate trends currently, so you should train models on all times before. Once you have created all your models, you can evaluate how well each one performs and start using the best to help yourself save money and time.