Bike Rental

Marketing Campaign Analysis

Course CSE 450 Brad Strange, Jacob Ferris, Landon Davis, James Hall, Keaton Kesler

This document shows the process and the reasoning behind our results and there impact. It goes over a collection of bike rental data. By the diffrent features and model we are able to predict the likelyhood of when someone will rent and understand what conditions. Which will help in running this Business in Washington DC.

Model

Model Features

The features that we chose to go with were everything besides the following: 'casual', 'registered', 'dteday', 'season'. The reason we chose these features is because we felt for our model to get the most performance it should take all of the features. The model "creates" it's own features while it works with the data, so we also felt feature engineering was mostly unneeded. We combined the casual and registered features to create a 'total' feature. We were ask to predict the amount of people so we took that as total. We dropped 'dteday' because the encoding with the model failed. We then recognized the possible strength of the date, so we instead chose to separate the columns into 'day', 'month', 'year'. We wanted to drop season because it seemed to make no real change to the data. The season can somewhat be interpreted through the temperature anyways.

Model Chosen

We chose a neural network model so that we could have the model teach itself about the data and to learn how features interact and our importance so that our model could give us a more accurate feel for the data.

Model Testing

We are confident that our model will explain around 79% (0.79 r squared) of the variance in future data.

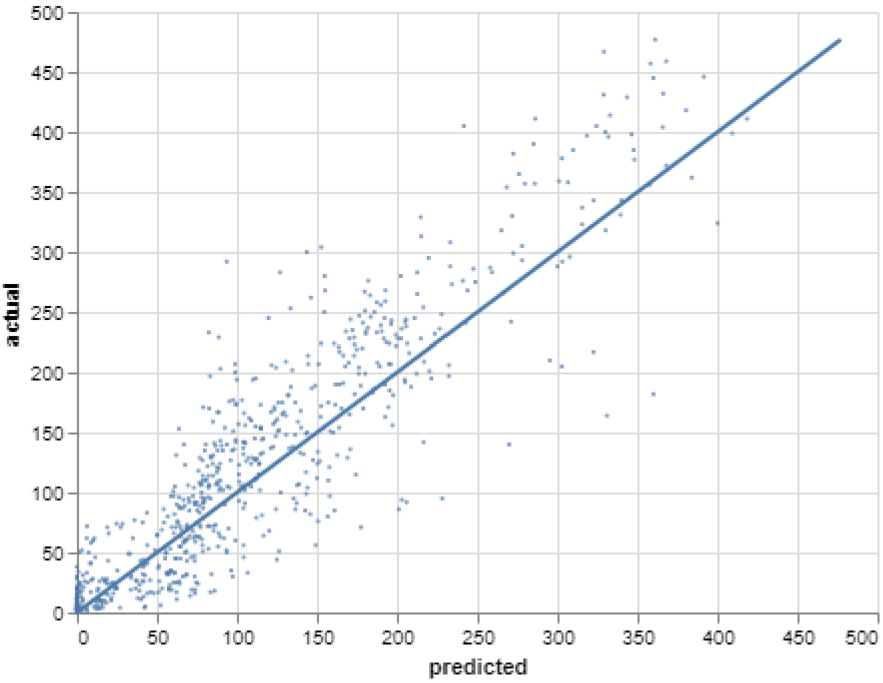
Because we were concerned that our model needed to be applicable to future data, and not simply other data points within the data range, we tested our ability to create models within the dataset that were extrapolated to future data points.After finding the hyperparameters with the highest interpolation r2 (0.928 validation), we used the hyperparameters to fit the model to the first 11 months of the first year (2011). This gave us an average r squared of 0.79.

An earlier version of the model hyperparameters got an average r squared of 0.69 on the extrapolation test, and was extremely inconsistent. After reviewing the code and thinking about the problem, we realized that the season contained the first months and the last days of the year, and dropped it from the dataset. Although allowing the test to influence the model has a possibility of biasing our test

results, we believe that the difference between the two is highly significant and that our later results are an accurate representation of the model’s ability to extrapolate.

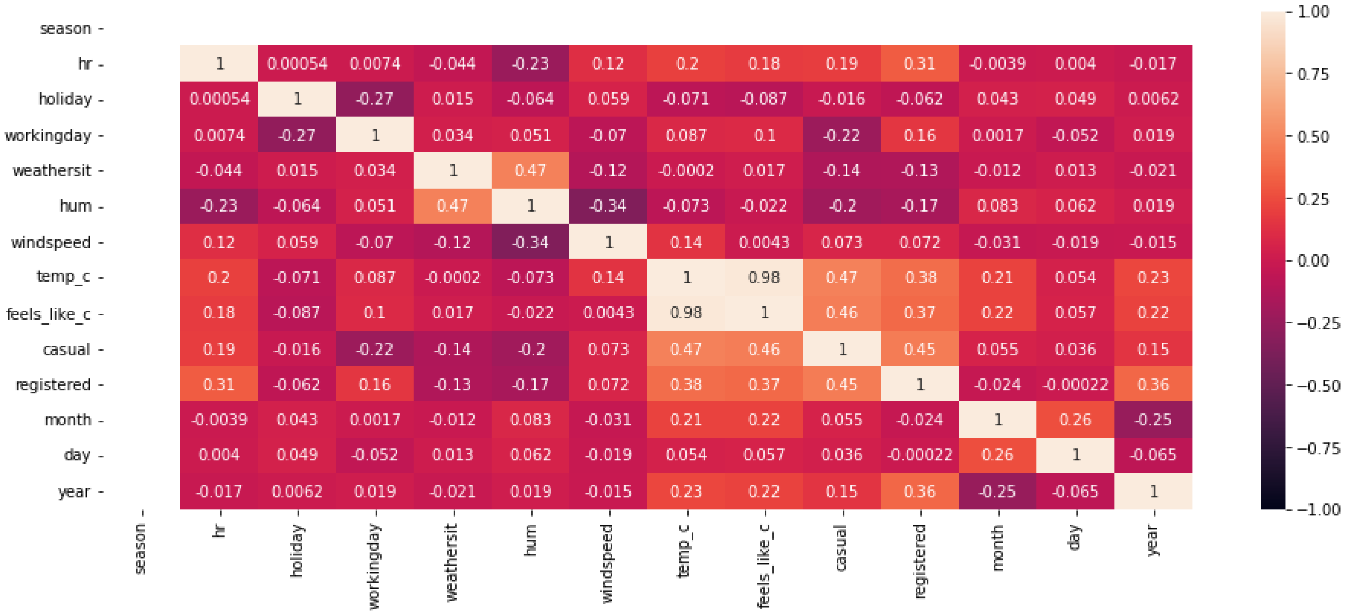
We used data points from the entire dataset as validation data, and this has the possibility of biasing our hyperparameters to fit the test extrapolation month. However, the difference between our validation errors were small, and often more determined by the model run than the hyperparameters. There was also only a small correlation between the validation set and the testing set.

Overall, we feel that our r2 estimate of 0.79 is more likely to be biased low than high. When we tested the model, it did not have any training data for the month of December. But our overall model does, and has more data overall.

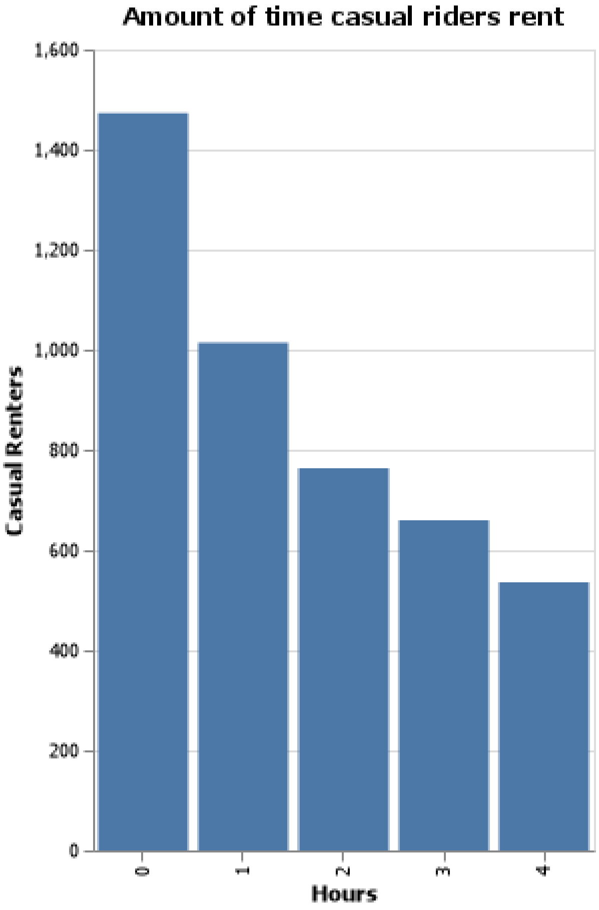


Here we can see our predicted vs actual to show our model prediction. We can see our model goes through the main meat of the data. As you can see our model actually predicts lower than actual for most of the data which isn’t actually a bad thing. It means our model predicts on the lower end of sales so that you can plan around a lower estimate or close to a “worst case” for revenue in the month. This means that you won't have a month where you earn less than our prediction.

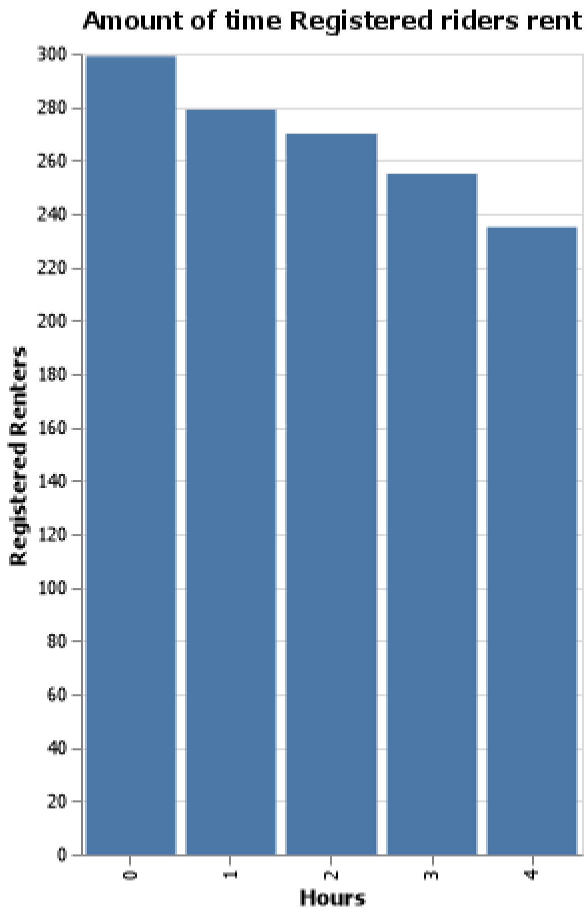
Visualizations



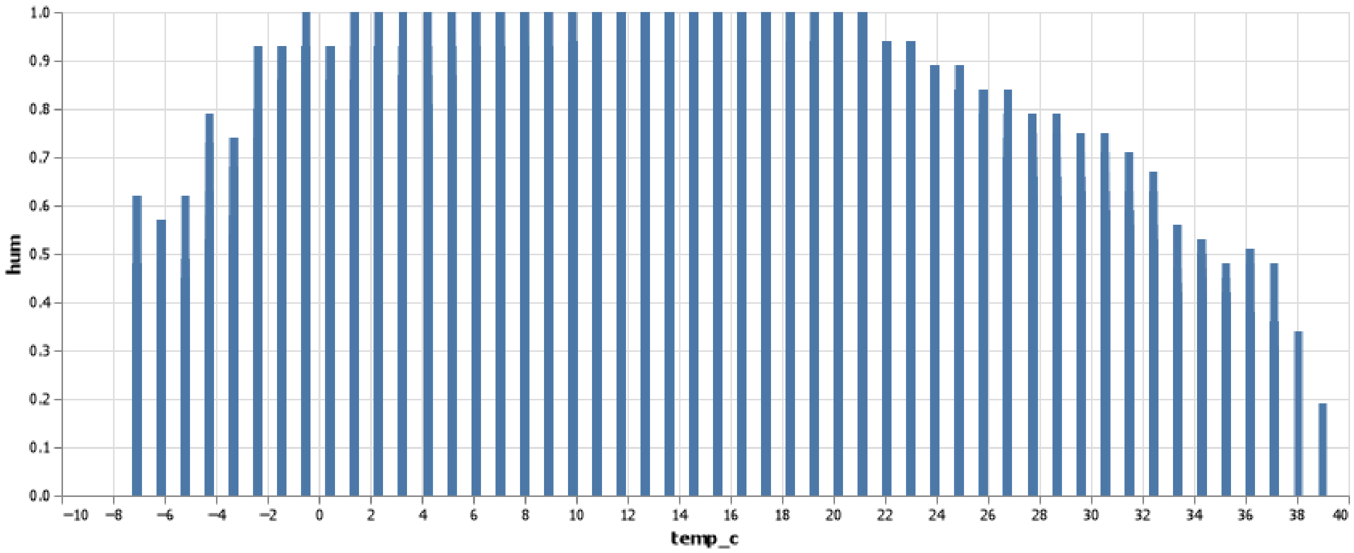
This heat map allows us to visualize all of the features and compare them to know which ones are more weighted and important for the model to use.



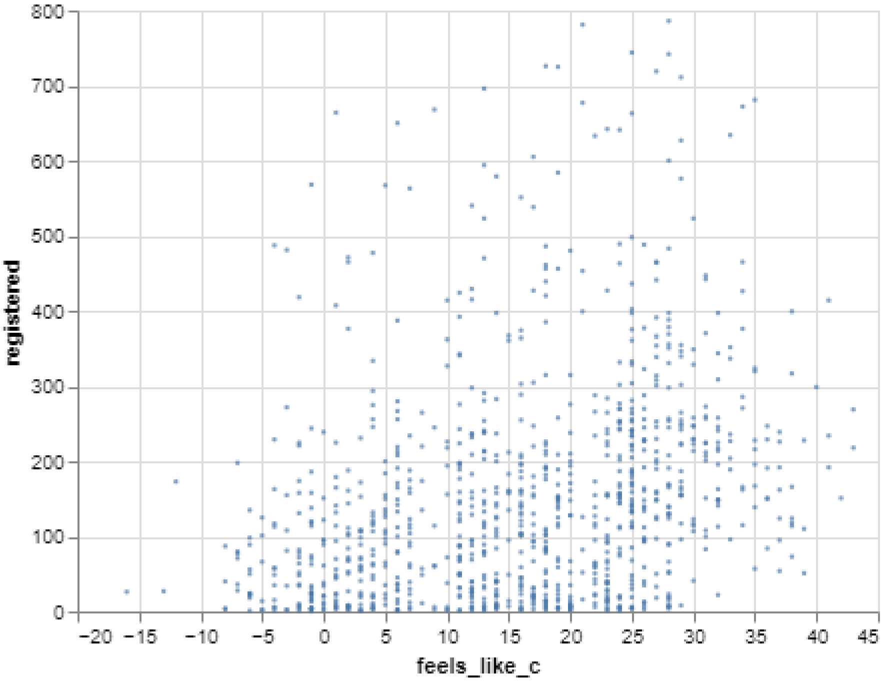
This chart shows us how long the adverage casul rider rents for. As we can see the majority of casual riders rent for less then an hour which comes out to almost being 1/3 of all casual renters and it steadly declines the more hours is past.



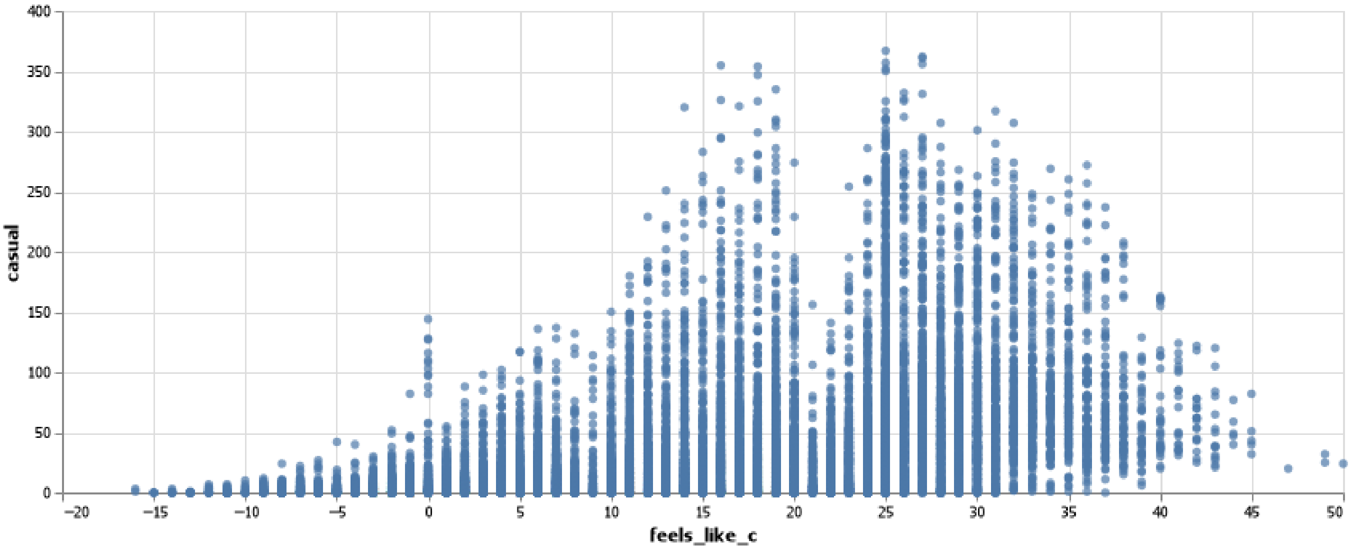
This chart shows how many hours a registered renter rents for on adverage. We see that again most riders rent for under an hour but its alot closer and a steadier decline the more hours that past. compared to casual riders which had more renters but the drops where more significant.



This chart shows the adverage humidity percentage compared to the tempature outside. We see that the humity rises the more it leaves freezing temperature into cooler temps but then once it gets into hot and very hot temperatures the humidity goes down.



This chart shows the ammount of registed renters depending on the feel like temperatures. We see more riders out between 10 and 30 degrees celsius. After and before those temperatures the ammount of riders drops.



We see again this chart shows the ammount of casual renters depending on the feel like temperatures. We see more casual riders are out between 10 and 30 degrees celsius. Hitting its peak at 25 degrees celsius. Then with the after and before temperatures the ammount of riders drops. So we see the importance of the data which can relate to the time of year when people are out renting bikes.

Discussion Responses

Below are our final answers to the case discussion questions:

Which of the following hyperparameters do you feel has the most potential for model improvement?

- **Learning rate and optimizer selection. We feel the learning rate helps us the most in tuning our model the best in how well it can find the global minimum of the loss function.**

How do you think we should handle the temperature features?

- **We decided the temperature features were good as is and didn’t feel a need to adjust them for any reason and used them without any binning or feature engineering.**

What approach do you think you're going to take to find the optimal learning rate?

- **We used loop functions to run our model through many iterations to see what learning rate gave us our best r-squared value on our testing data set.**

What are you planning to use for the loss function?

- **We used mean-squared error. This ensures that the average of the predicted values will equal the average of the actual values.**

We would like use AI to predict the likelihood of damage based on user profile data, such as name, birthday, sex, or address, so that we can add an insurance premium to the rental cost. We are concerned that there may be ethical/legal implications here, what would you recommend?

- **We believe that it is not unethical to use certain features of renters to set varying insurance premiums as we see examples of this in things like car insurance and other forms of liability laws and could be used to help make some more income for the company.**

APPENDIX A (PYTHON CODE)

Colab Notebooks:

- https://colab.research.google.com/drive/1JPvWaLzAi9t6bM1yfED07Lq13WtfWbMA#scrollTo=bcun3A69Ncx0
- https://colab.research.google.com/gist/jamesbhall423/3c598fafb3f6ea94211fde50f4509dc3/starter_bikes.ipynb