PDPilot User Study

Introduction and Interview Slides

Introduction

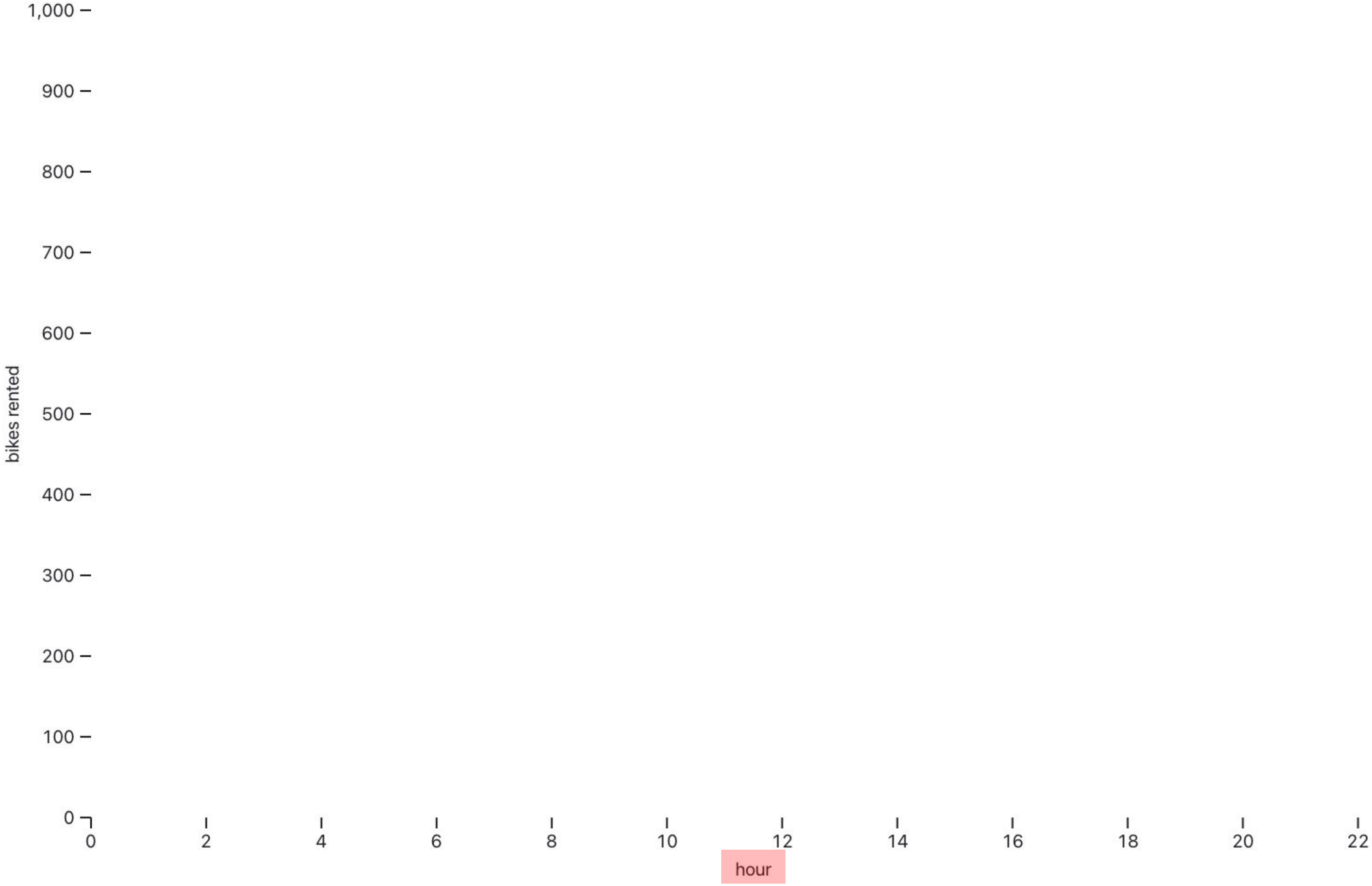
- We have developed an interactive tool for exploring partial dependence plots (PDPs) and individual conditional expectation (ICE) plots in Jupyter notebooks.
- We are evaluating this tool with machine learning practitioners.

Agenda

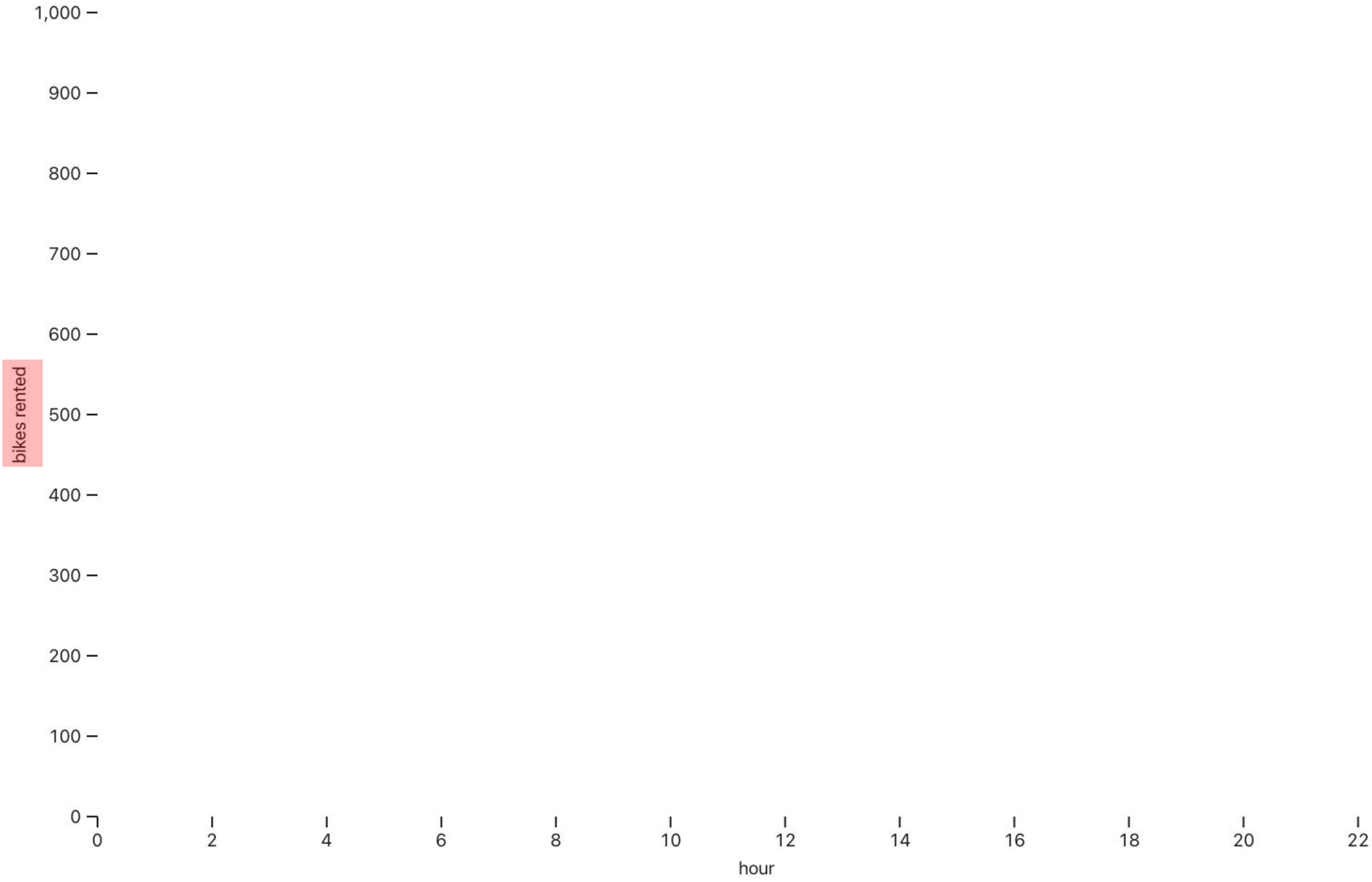
- PDP/ICE plot review (~15 min.)
- Tool Tutorial (~15 min.)
- Tool Use Verification (~20 min.)
- Model exploration (~40 min.)
- Interview (~15 min.)

Review

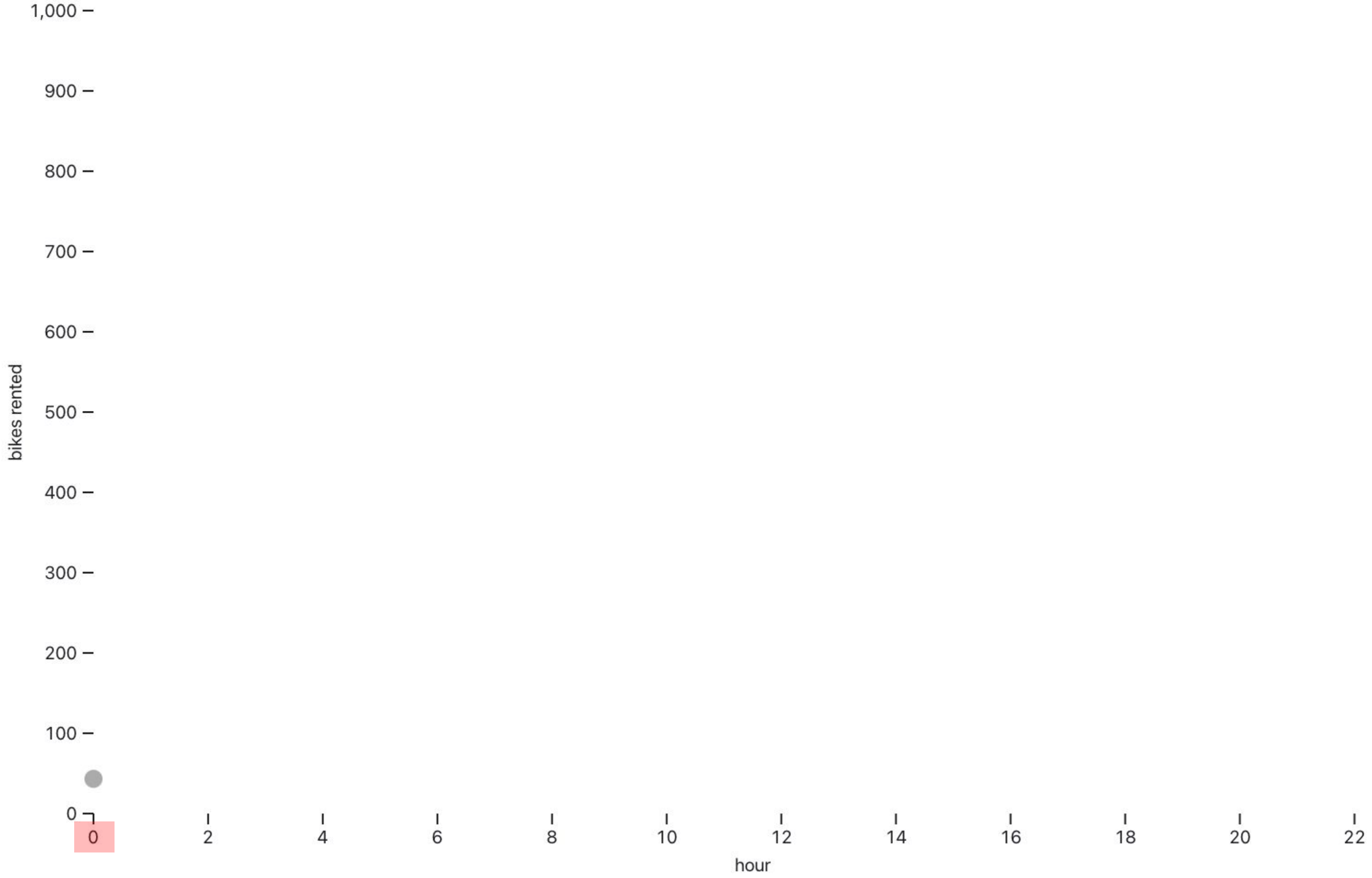
- Machine learning on tabular data.
- Partial dependence plots (PDPs) and individual conditional expectation (ICE) plots are common model explainability techniques.
- They show how a feature or pair of features impacts a model's predictions.
- Terminology:
    - “One-way”: the plot shows one feature.
    - “Two-way”: the plot shows two features
- Example of how to calculate ICE plots and PDPs:
    - Bike rental dataset
    - The ML task is to predict the number of bikes that are rented at a given day and time.
        - Example features: hour of the day, temperature, humidity, whether or not it is a working day



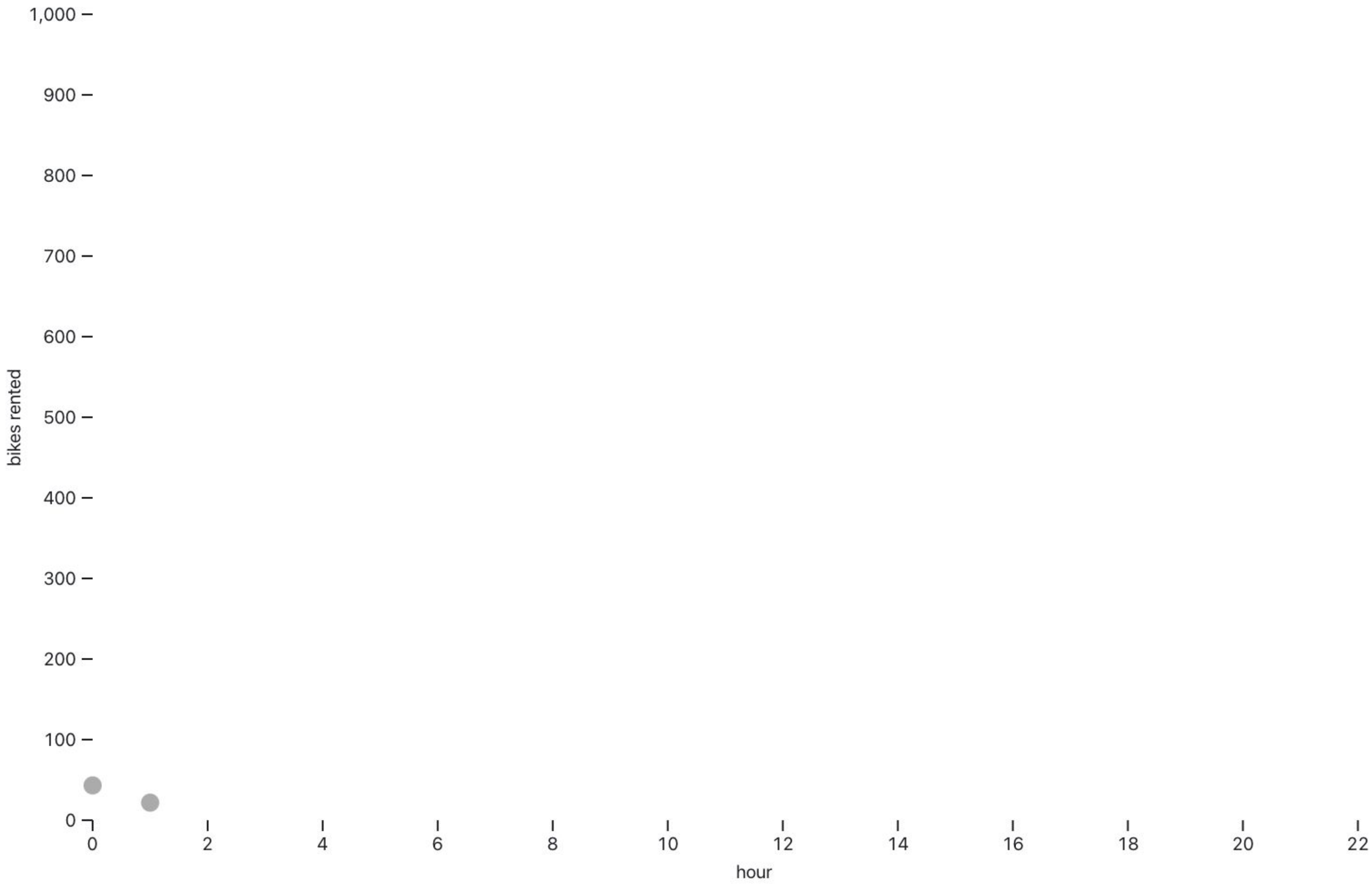
The x-axis shows the feature (hour of the day).



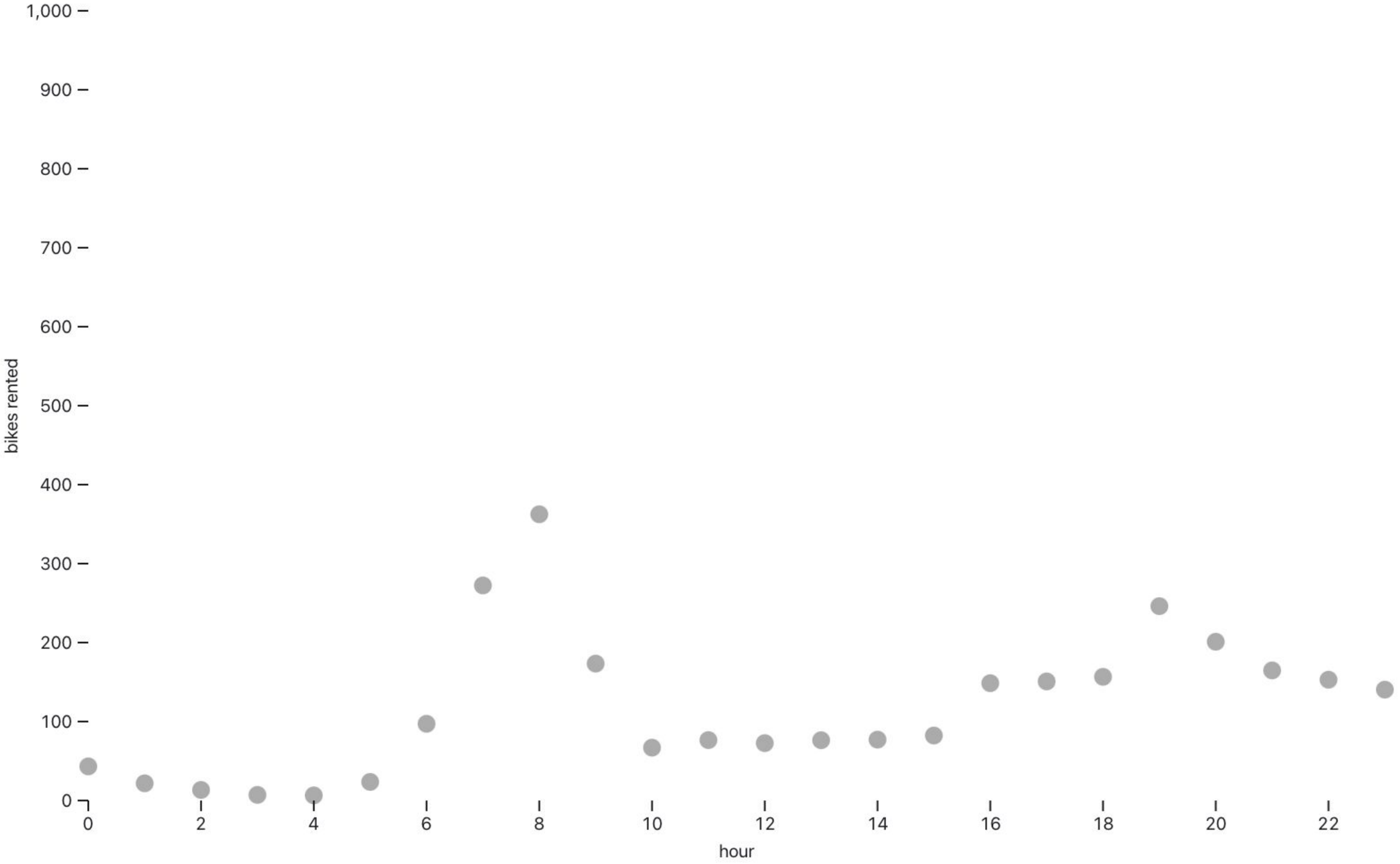
The y-axis shows the target or prediction (number of bikes rented).



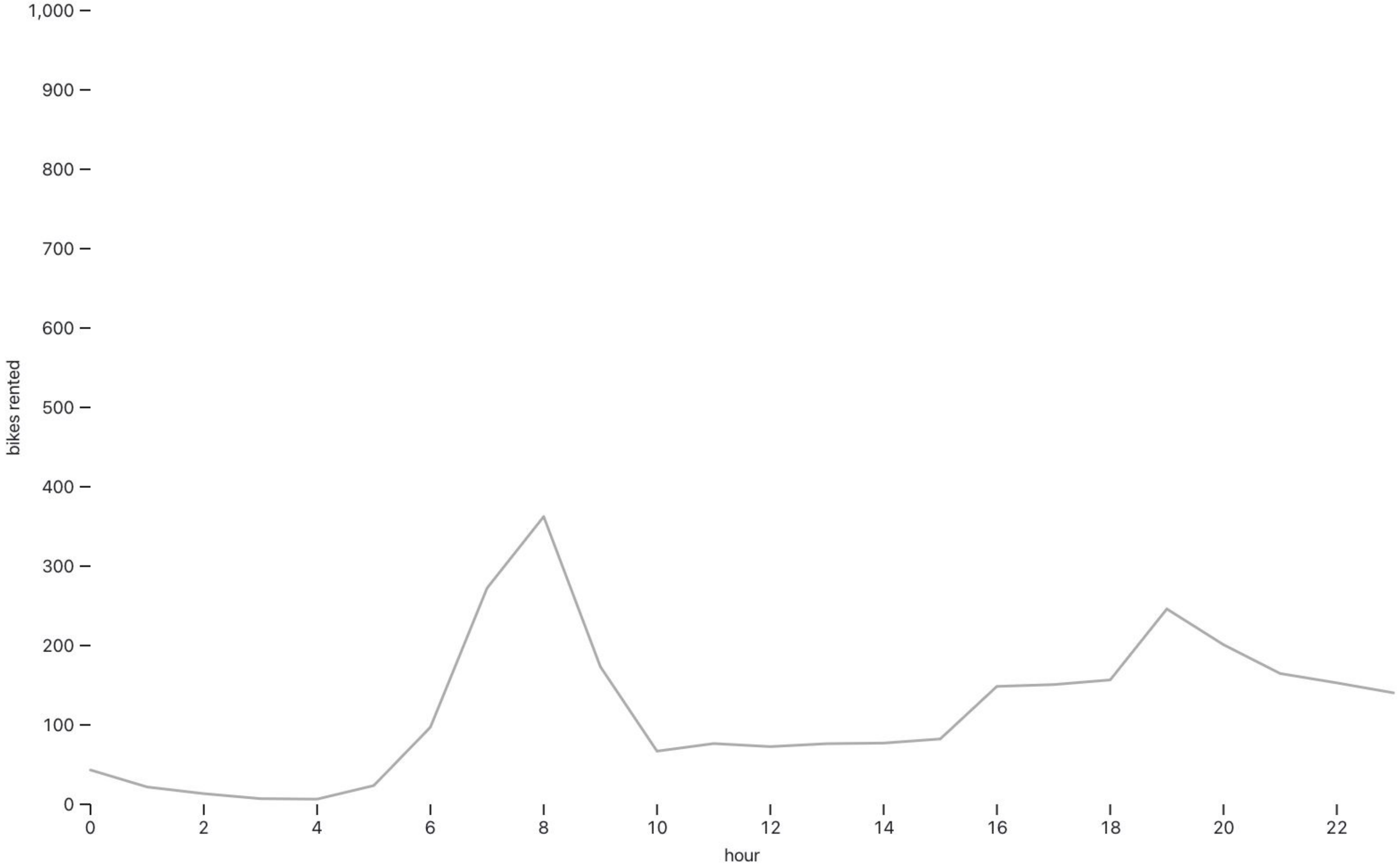
We select an instance from our dataset and set its hour to 0. We then get the model’s prediction.



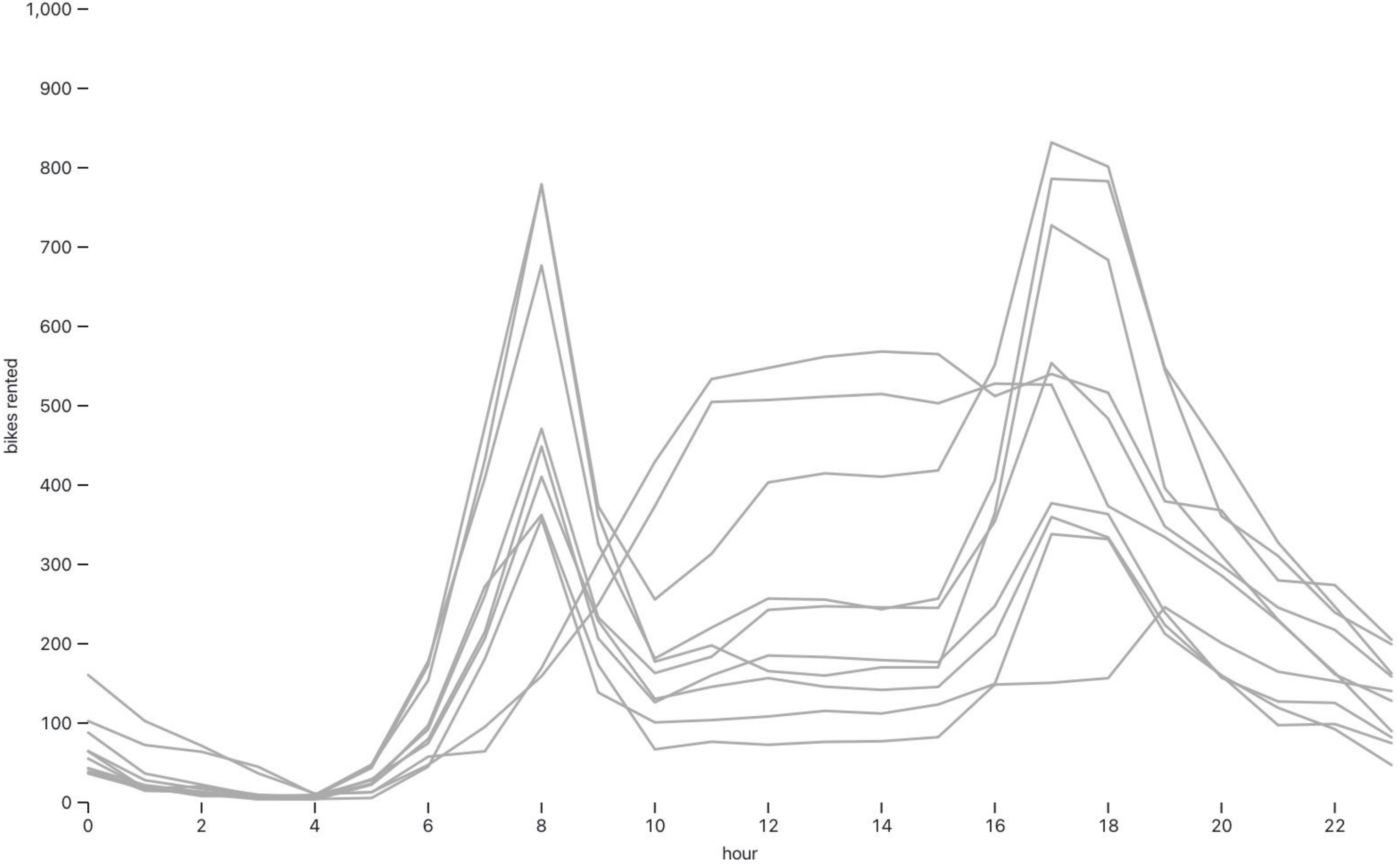
Using the same instance, we next set its hour to 1 and again get the model’s prediction.



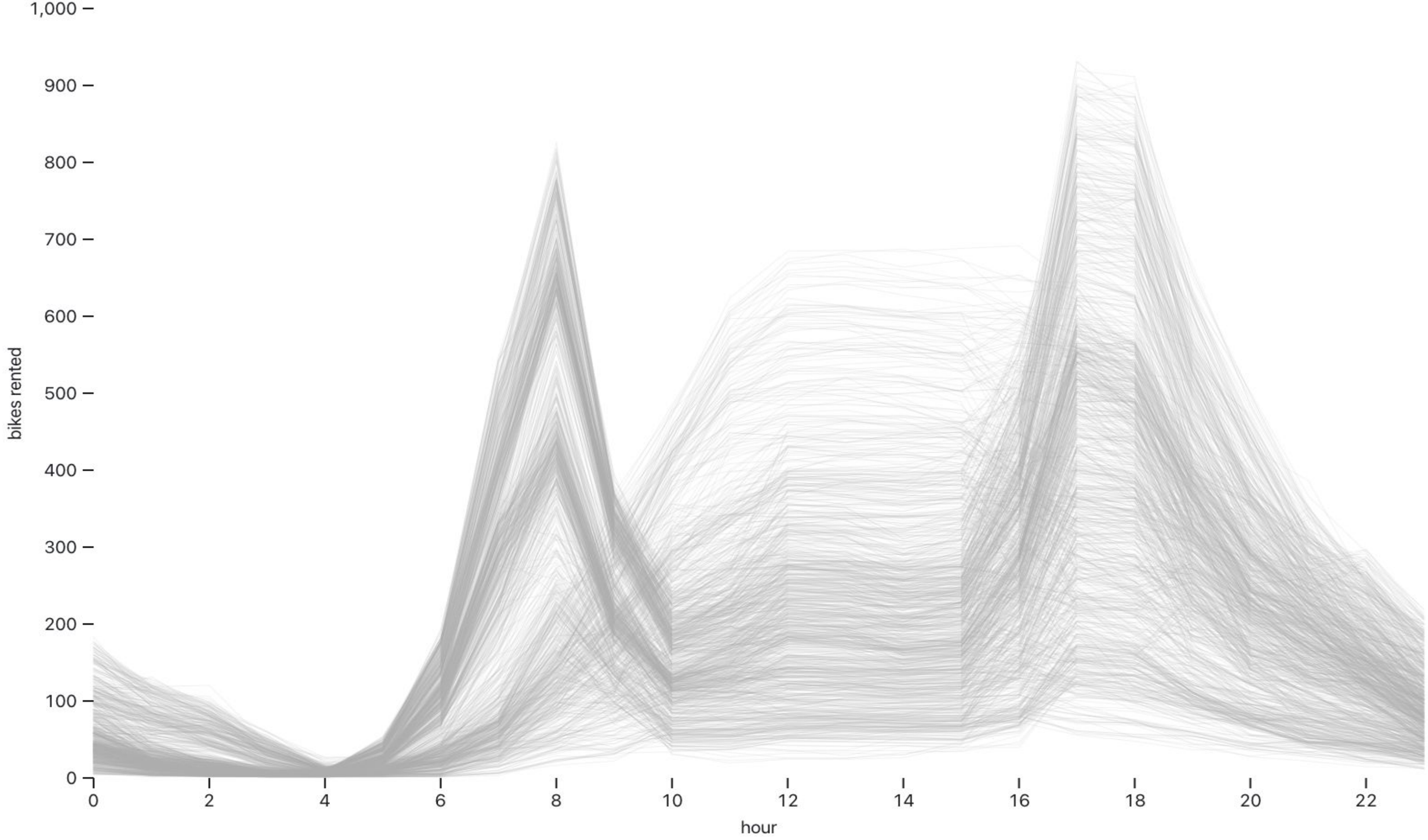
We repeat this for the different values of hour, using the same instance.



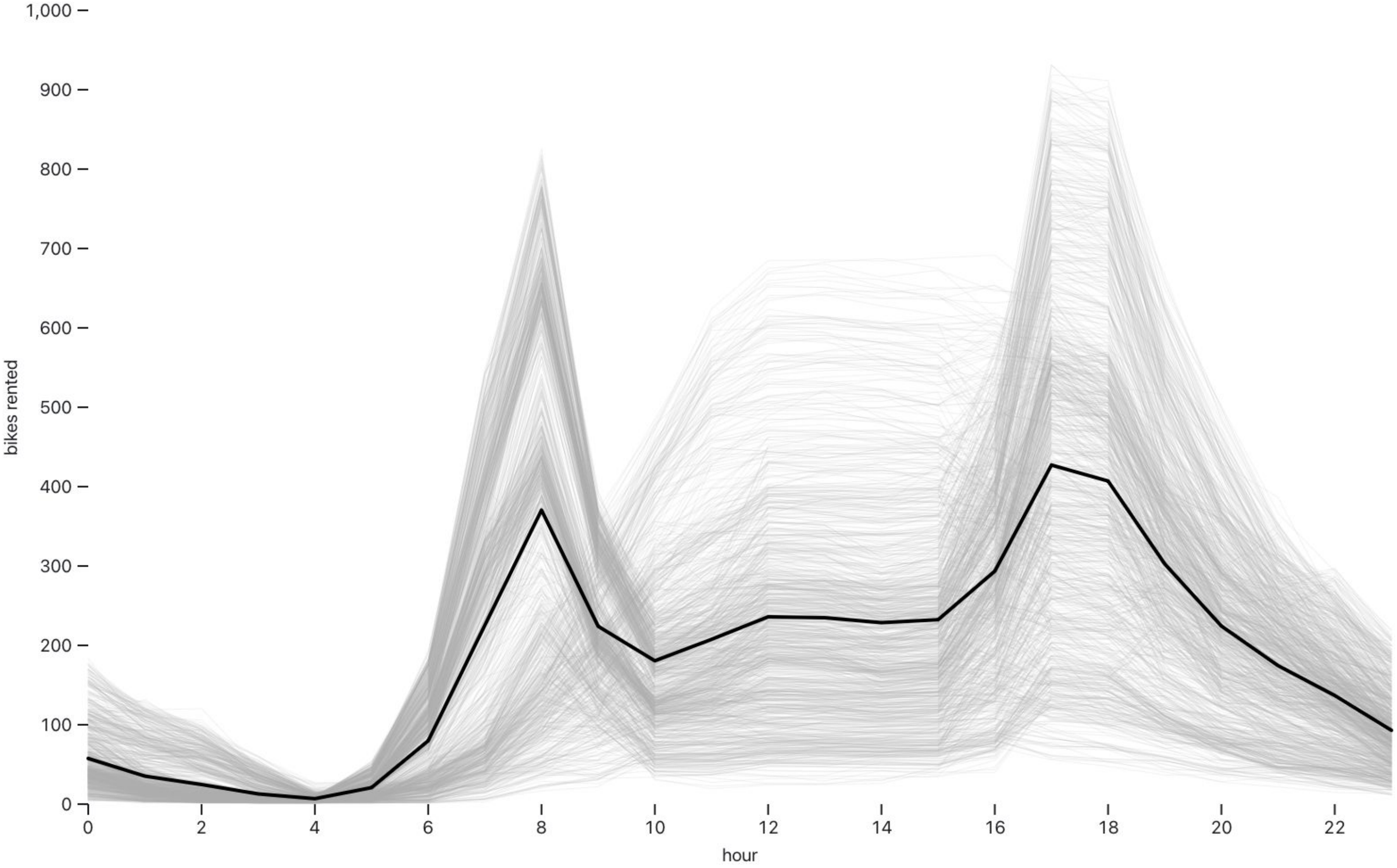
We can connect these dots with a line to create an ICE line for this feature and instance.



We can repeat that process for multiple instances.



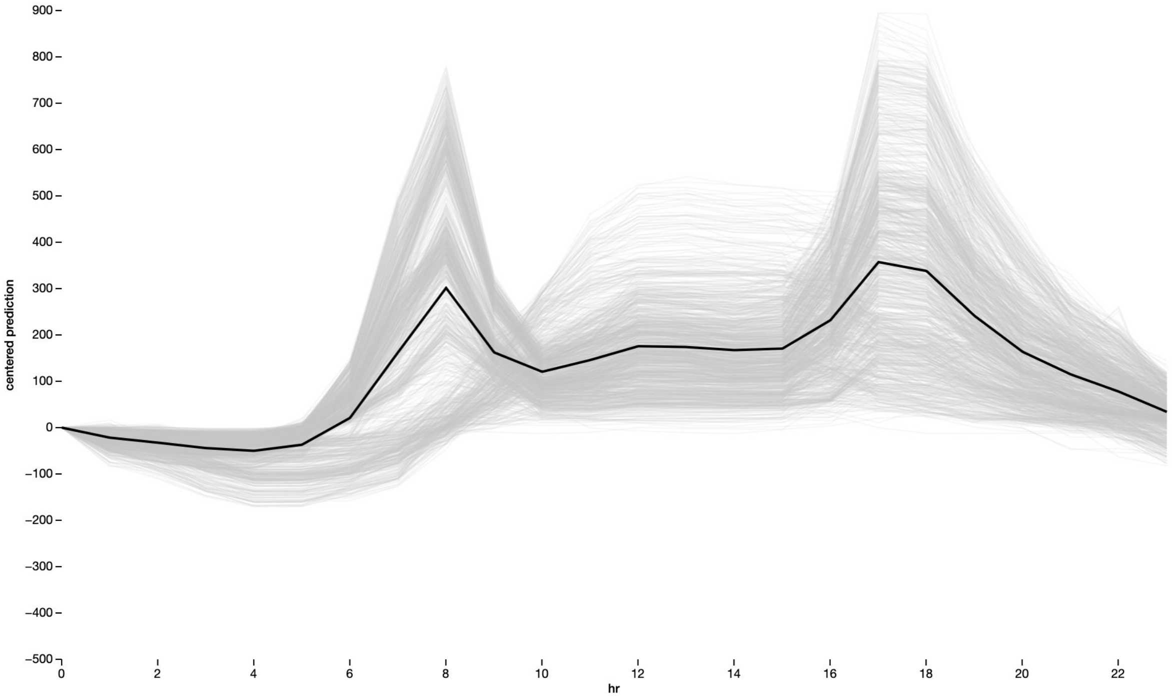
We can repeat that process for multiple instances.

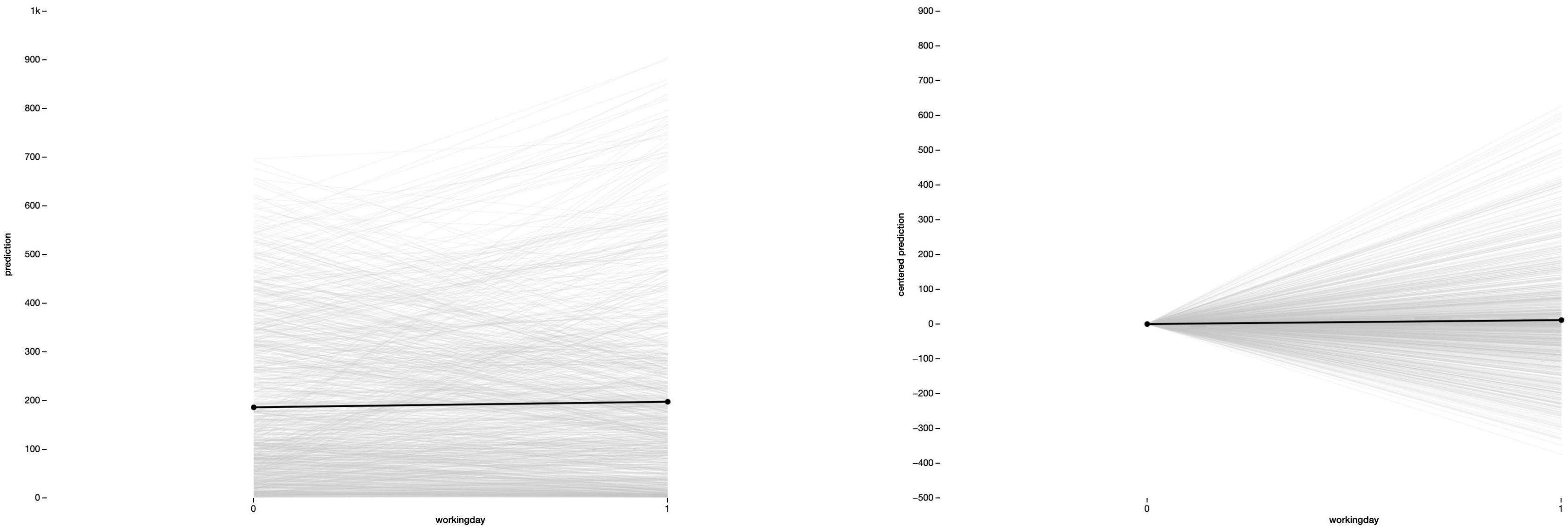


We can calculate the average prediction at each hour to get the partial dependence plot.

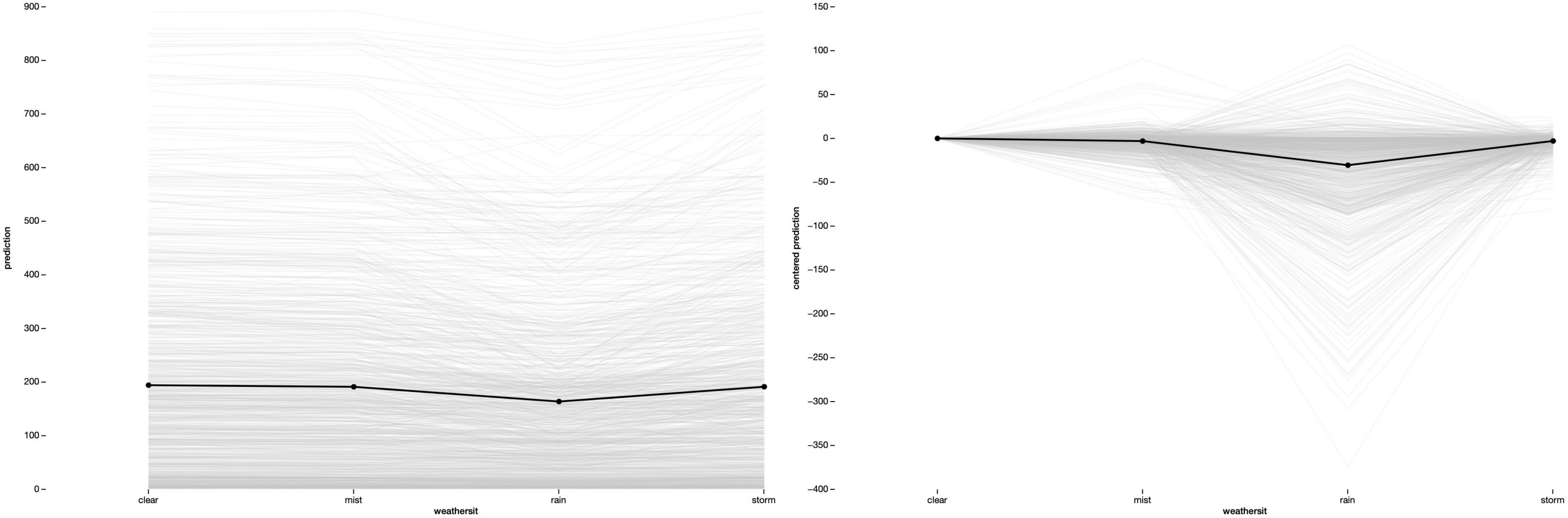
Centered ICE Plot

- All lines start at y = 0
- Normalizes ICE lines
- Useful for comparing the shapes of ICE lines
- Y-axis now shows difference in prediction from hour 0.





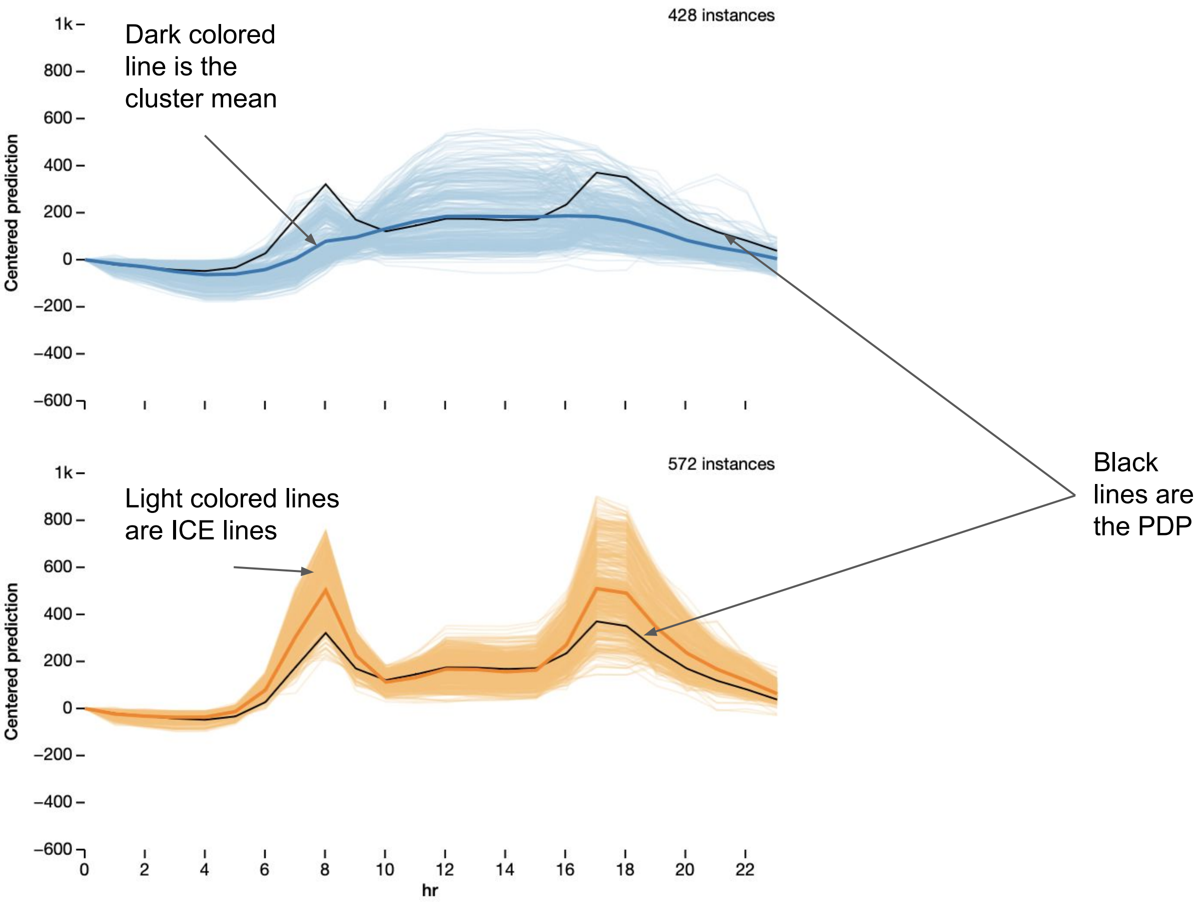
Standard vs. Centered ICE Plot for the working day feature



Standard vs. Centered ICE Plot for the weather situation feature. The plot is centered based on the first category.

Clustering ICE Lines

- Clustering the lines in a centered ICE plot by their shape can reveal subsets of instances with different behavior for the given feature.

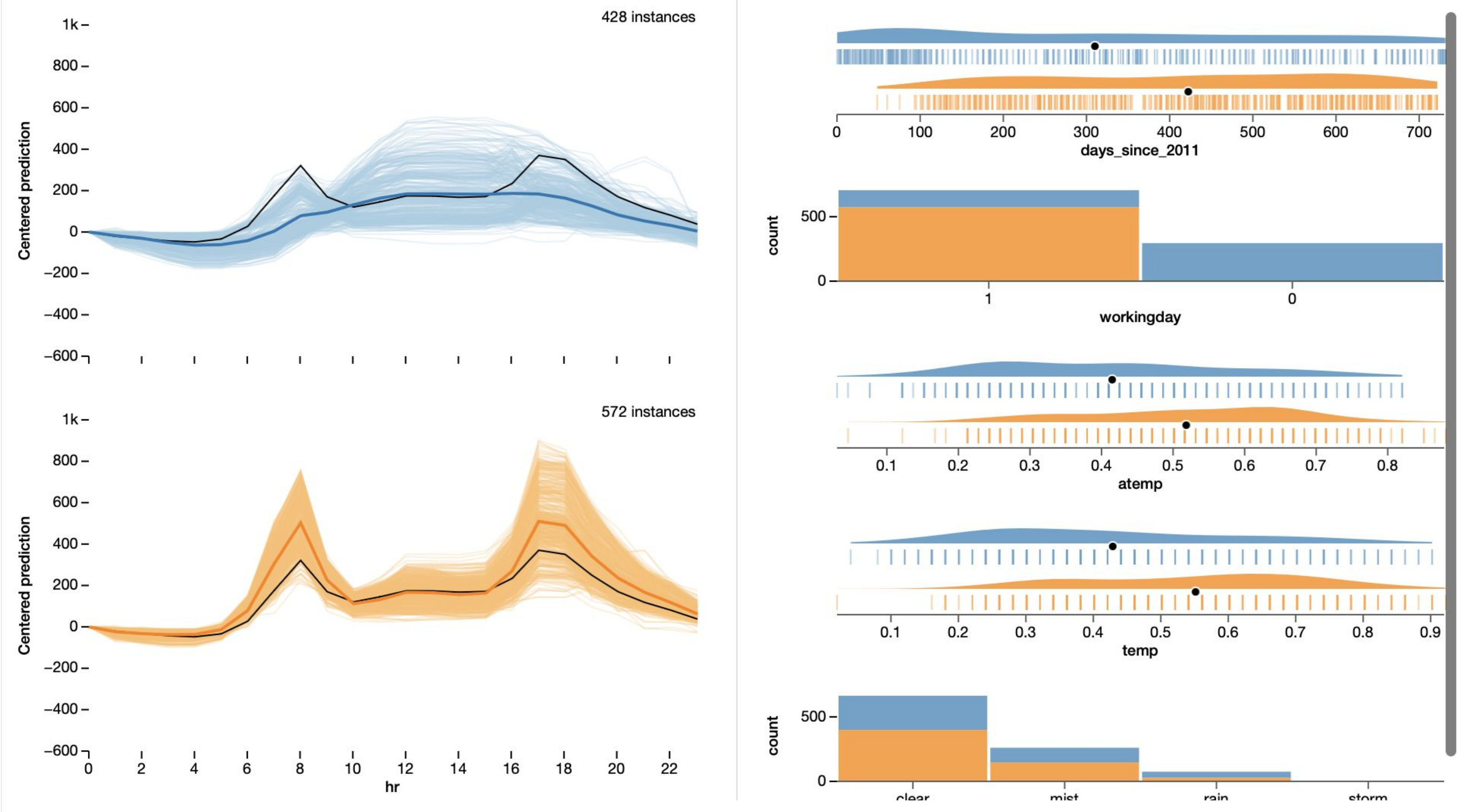


Dark colored line is the cluster mean

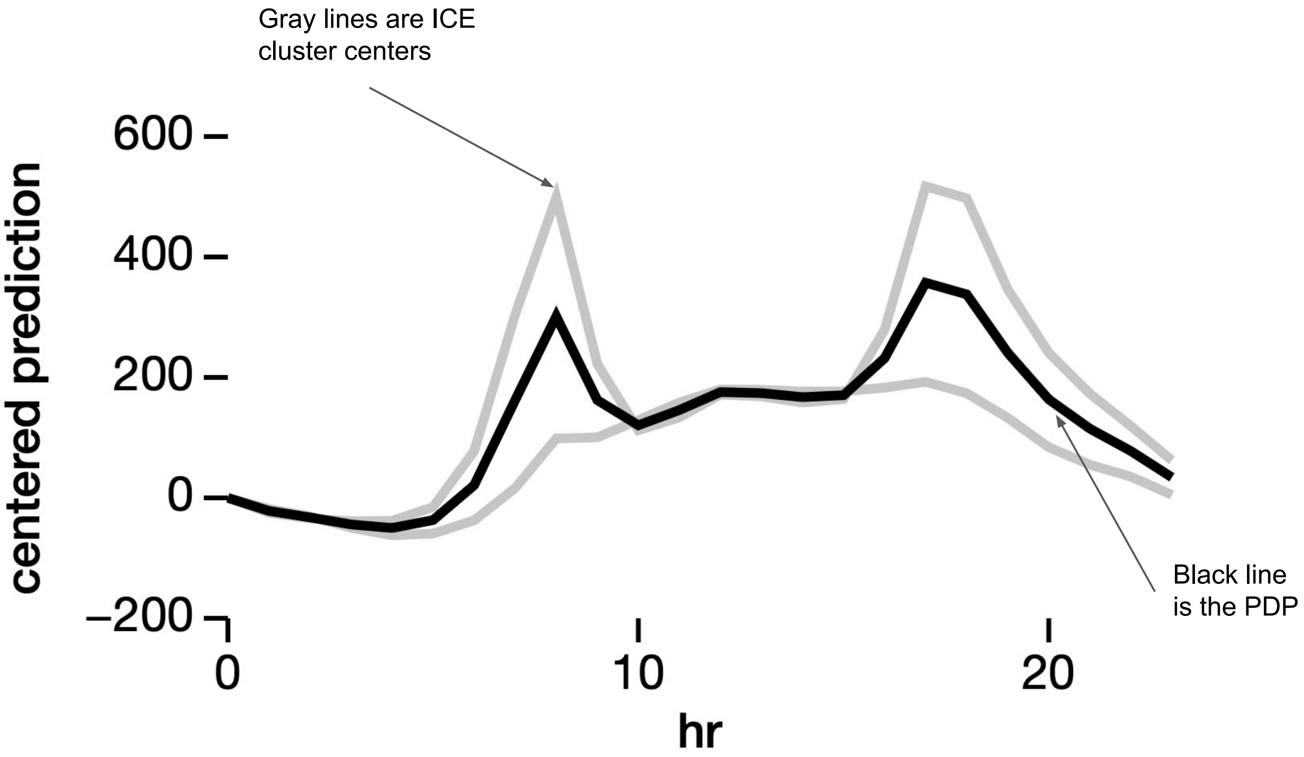
Light colored lines are ICE lines

Black lines are the PDP

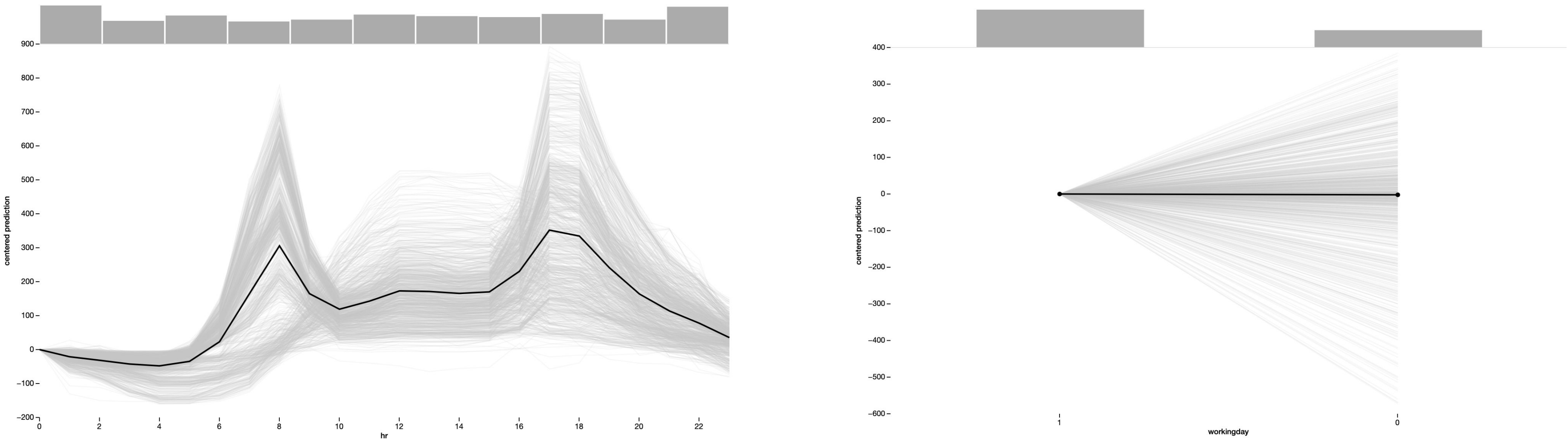
We can show each cluster in its own plot.



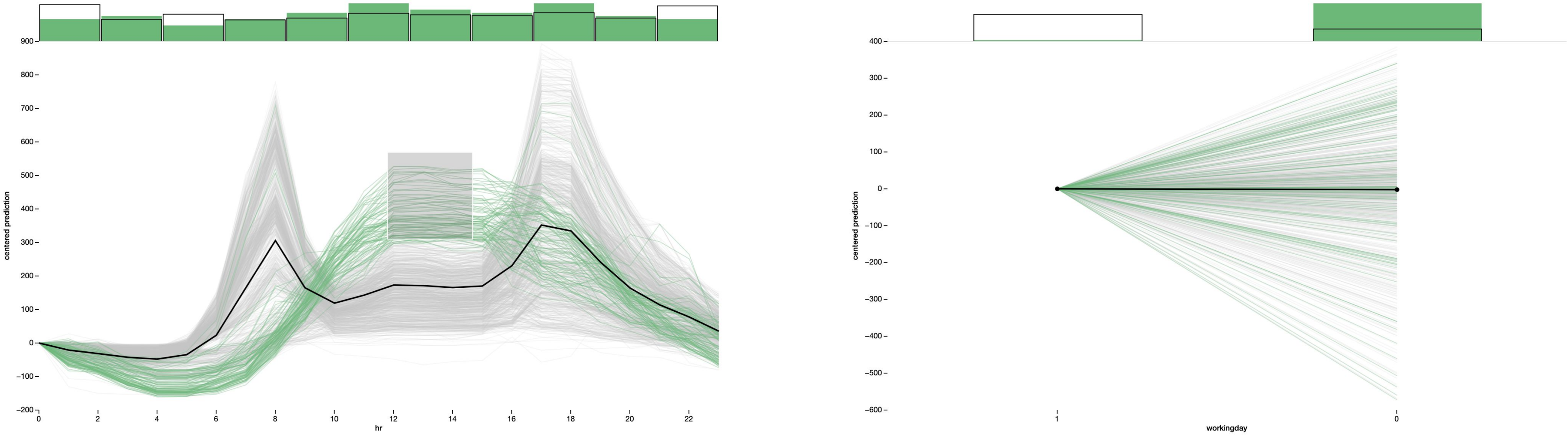
To help explain the types of instances in each cluster, we can compare the feature distributions for the instances in each cluster for a subset of features.



We can show also show the cluster centers in one plot.

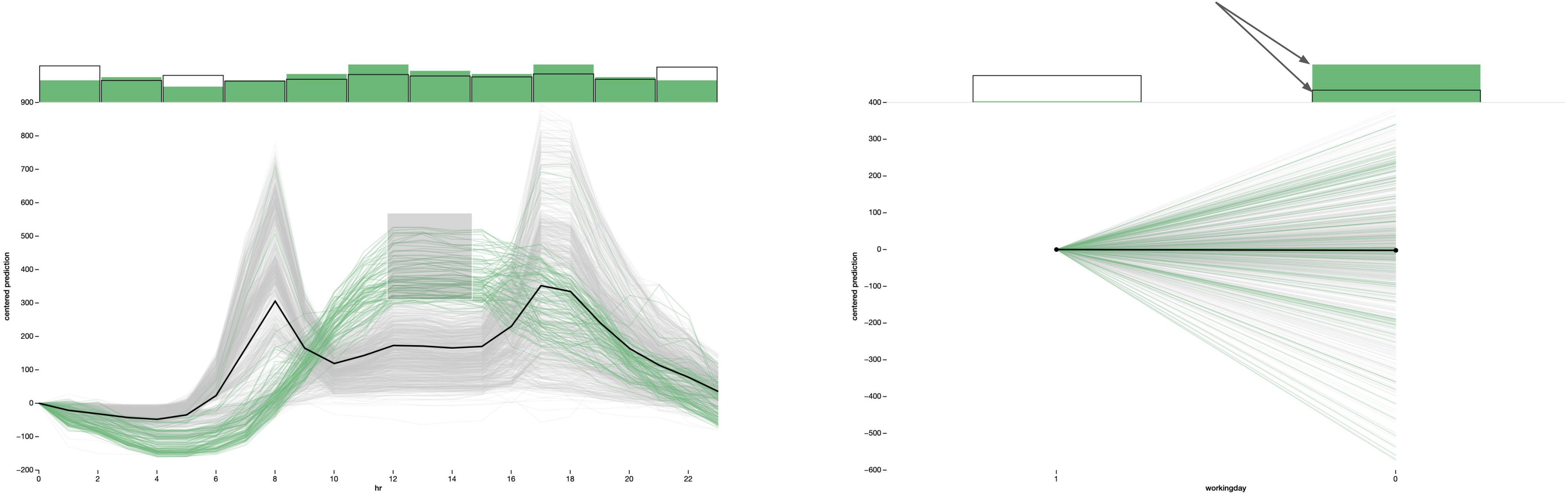


We can show feature distributions using histograms or bar charts above the plots. These charts show the distribution of the original values for a feature in the dataset.

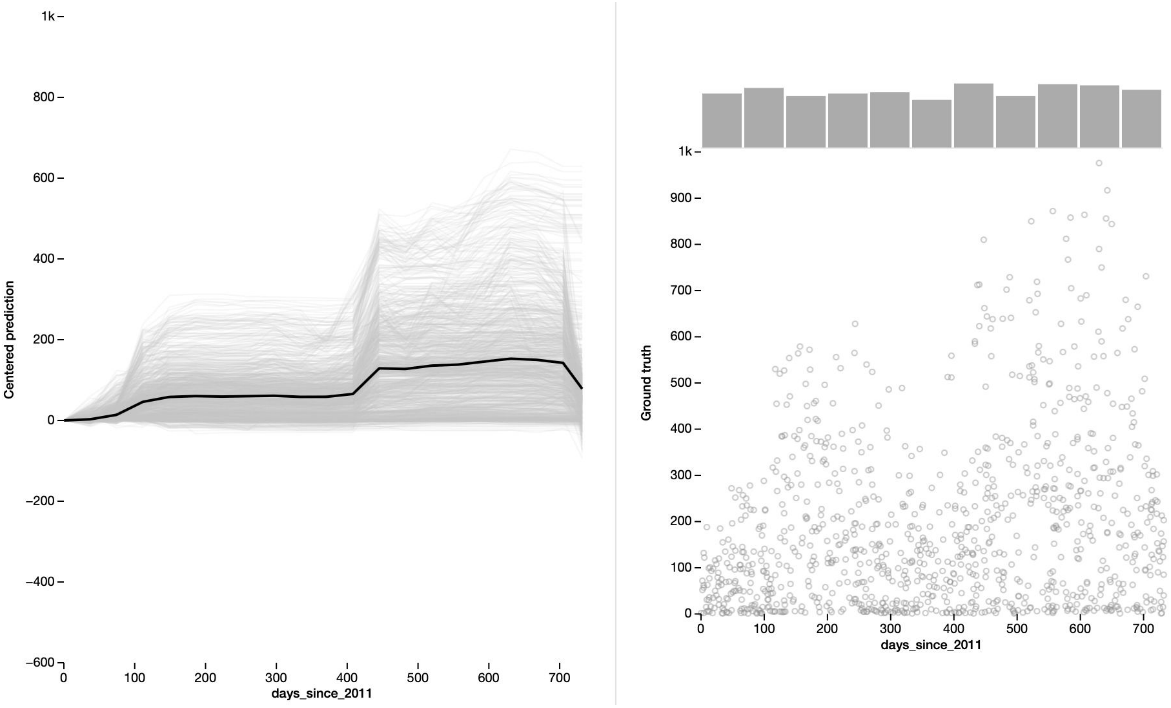


We can brush lines in one plot and highlight the lines for those instances in all other plots.

The percentage of highlighted instances that are non-working days is higher than the percentage of all instances that are non-working days.

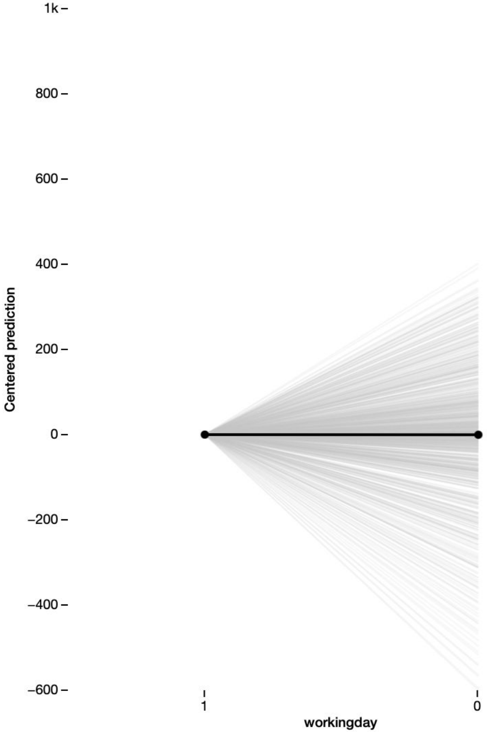


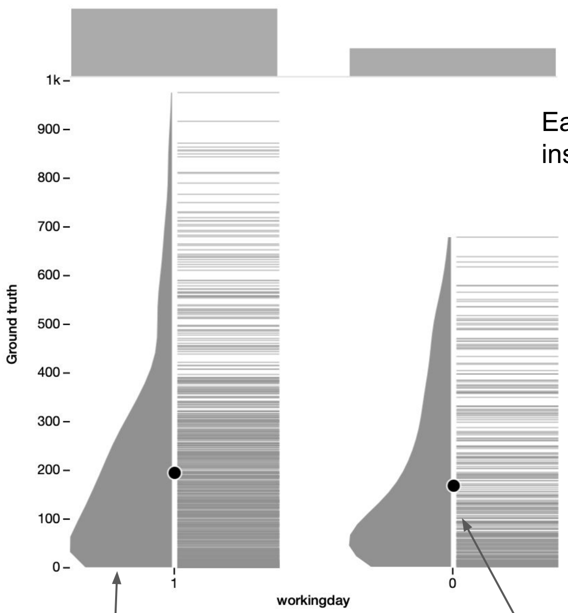
Each feature has two overlaid distribution plots. The green bars show the distribution of the highlighted instances. The transparent bars with black outline show the distribution of all instances.



For a given feature, we can also show the relationship between the feature’s values and the ground truth labels. When the feature is quantitative, this is visualized with a scatterplot.

Histogram or bar chart shows the distribution of feature values.



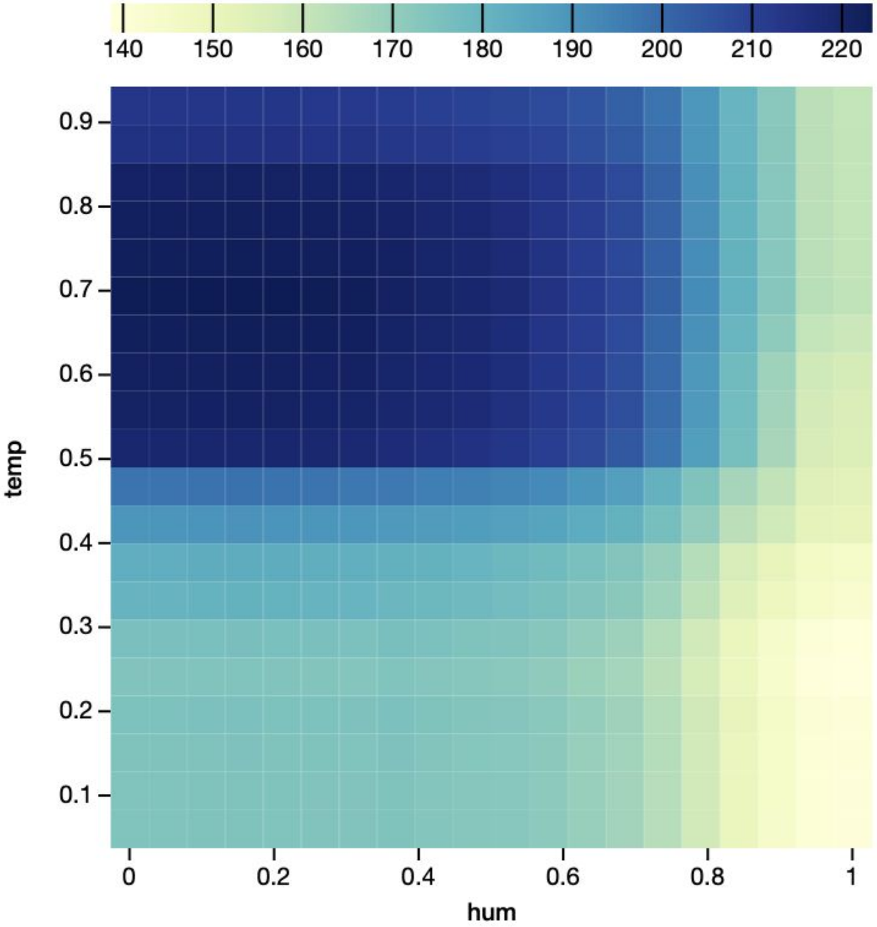


Each tick mark represents once instance in the dataset.

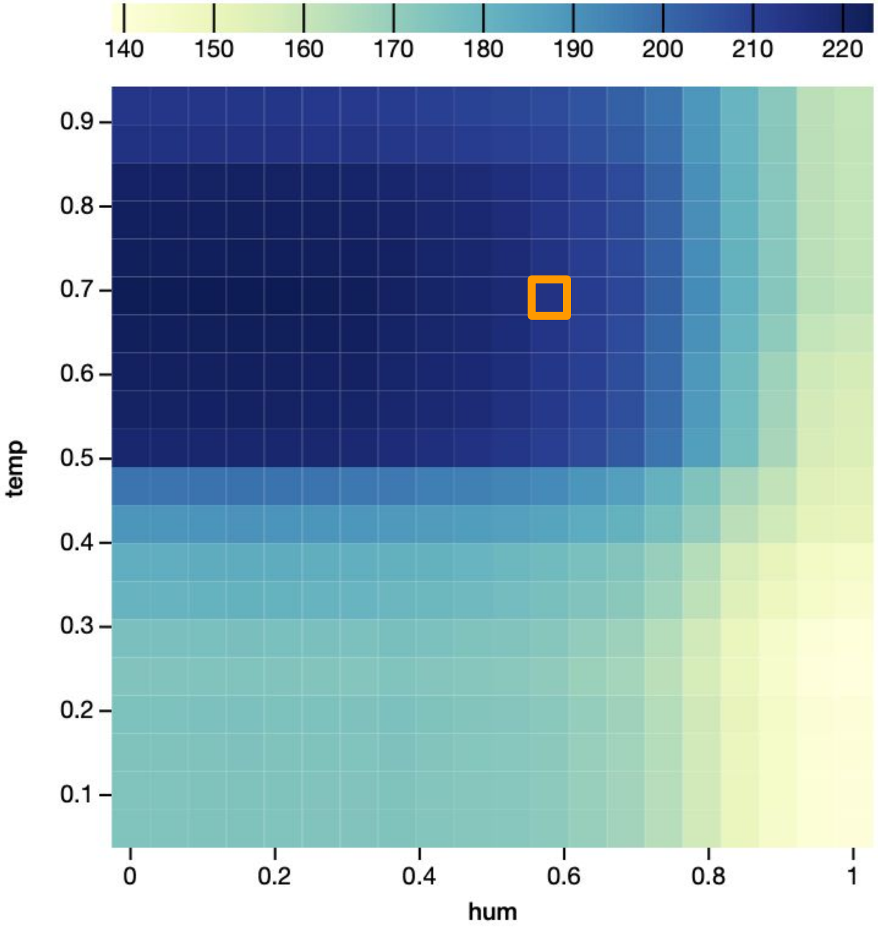
The density plot shows the distribution of ground truth values.

The circle shows the mean ground truth label for the given feature value.

When the feature is categorical, this is shown through raincloud plots.



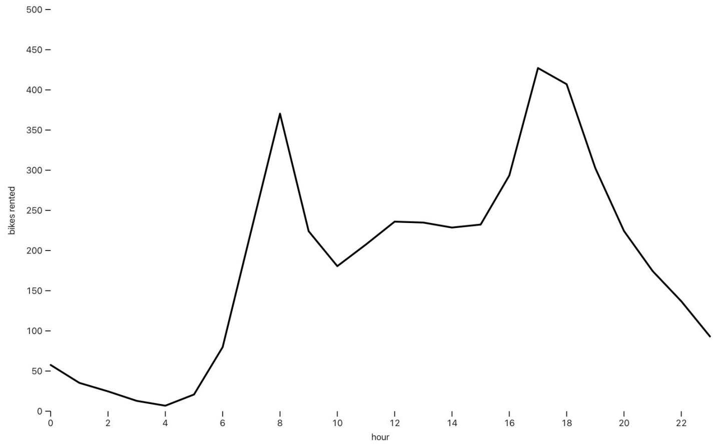
Here we show the two-way PDP for normalized temperature and humidity.

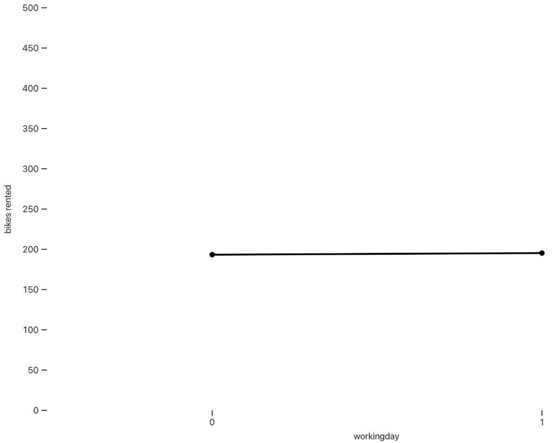


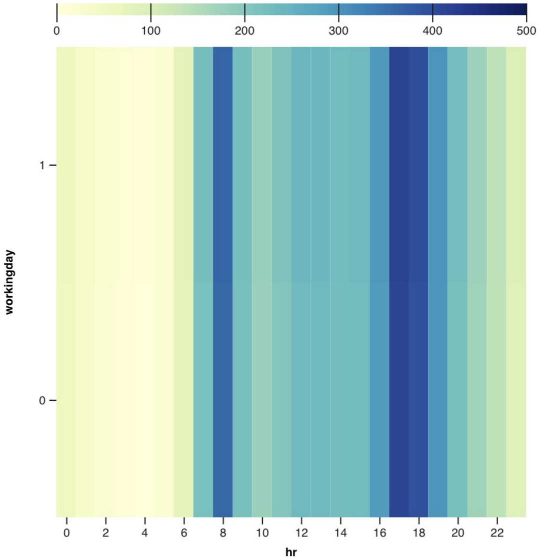
To calculate the value for the orange cell, we set all instances to have a temperature of ~0.7 and a humidity of ~0.6 and get the model's average prediction.

Interactions

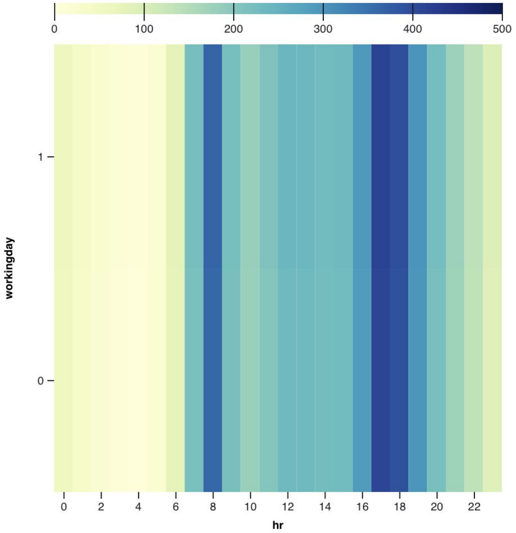
- Do the hour and working day features interact? If so, how?
- Using the one-way PDPs for the hour and working day features, we can calculate what the two-way PDP should look like assuming no interaction between those two features.
- We can then compare the estimated two-way PDP assuming no interaction to the actual two-way PDP.
- The difference between the two lets us detect and visualize the interaction between the two features.



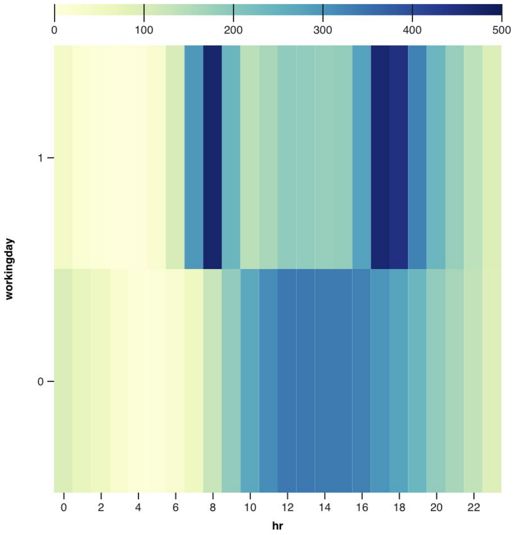




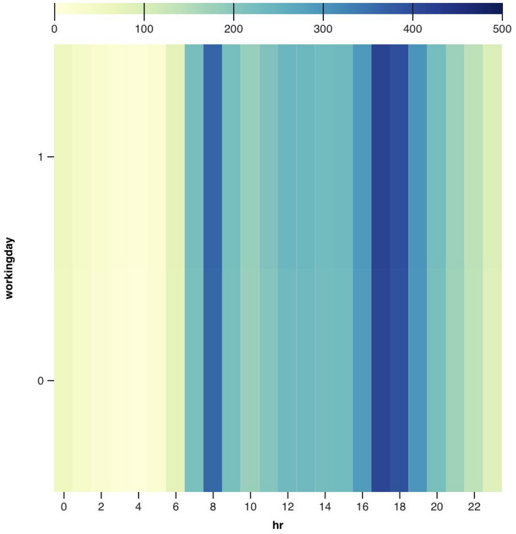
Based on the one-way PDPs for hour and working day, this is what we expect the two-way PDP to look like if there is no interaction between these features.



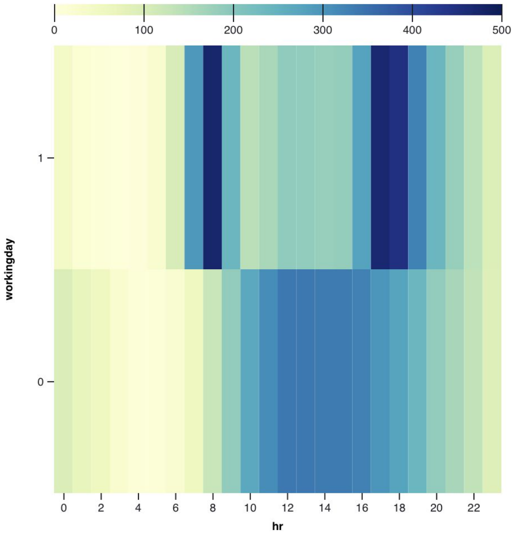
Expected PDP
with no interaction



Actual PDP

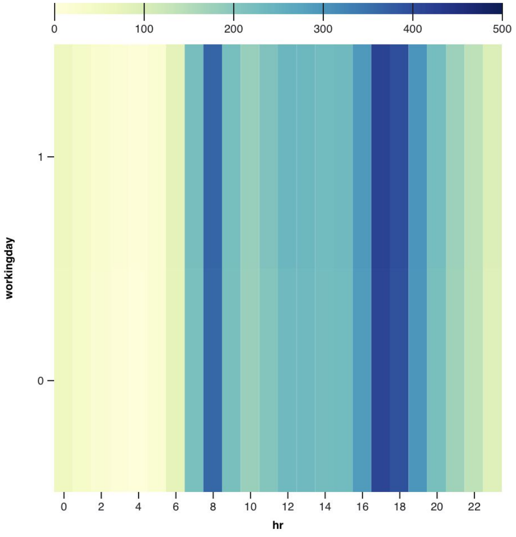


Expected PDP
with no interaction



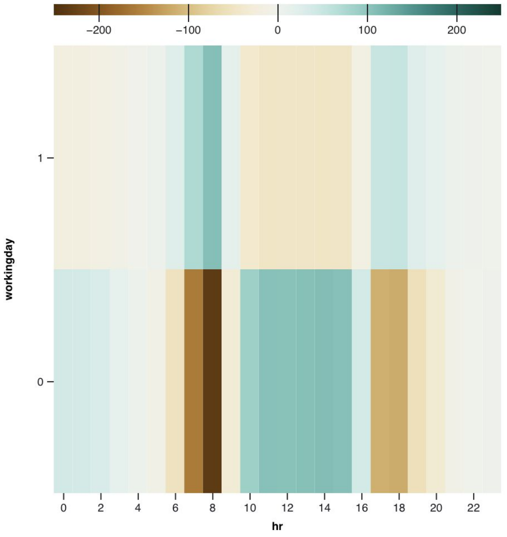
Actual PDP

−

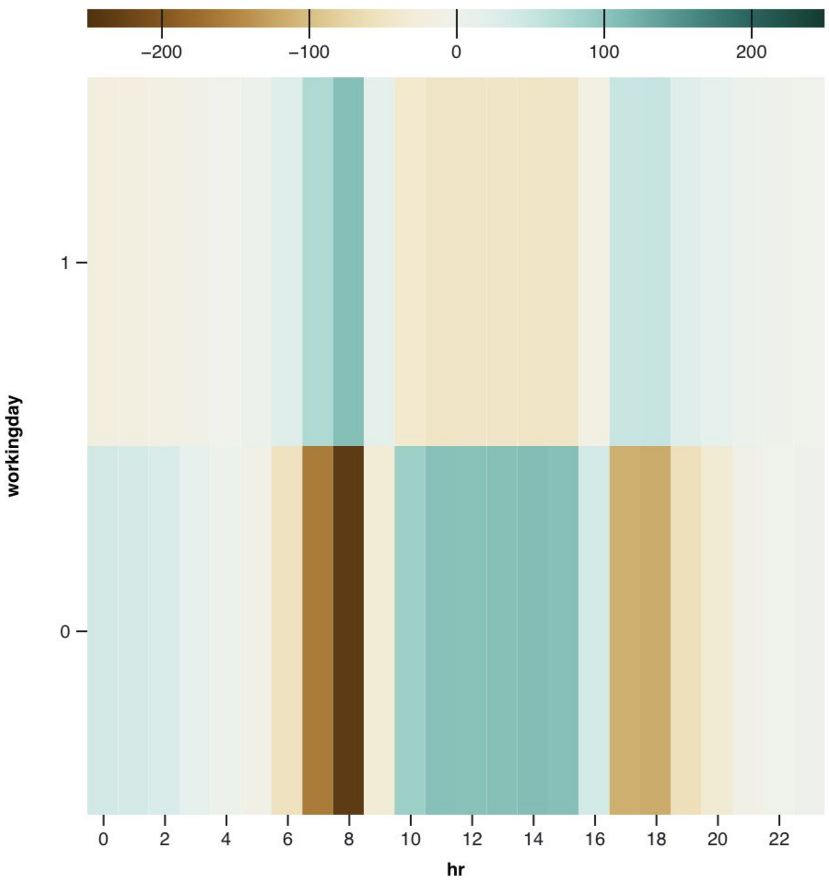


Expected PDP
with no interaction

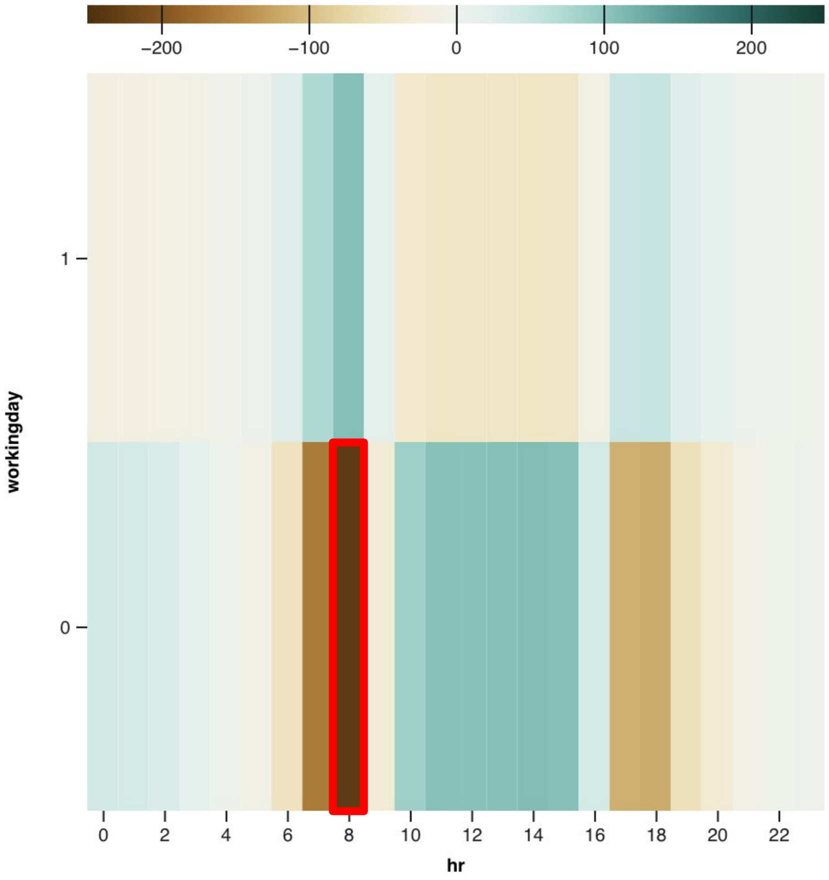
=



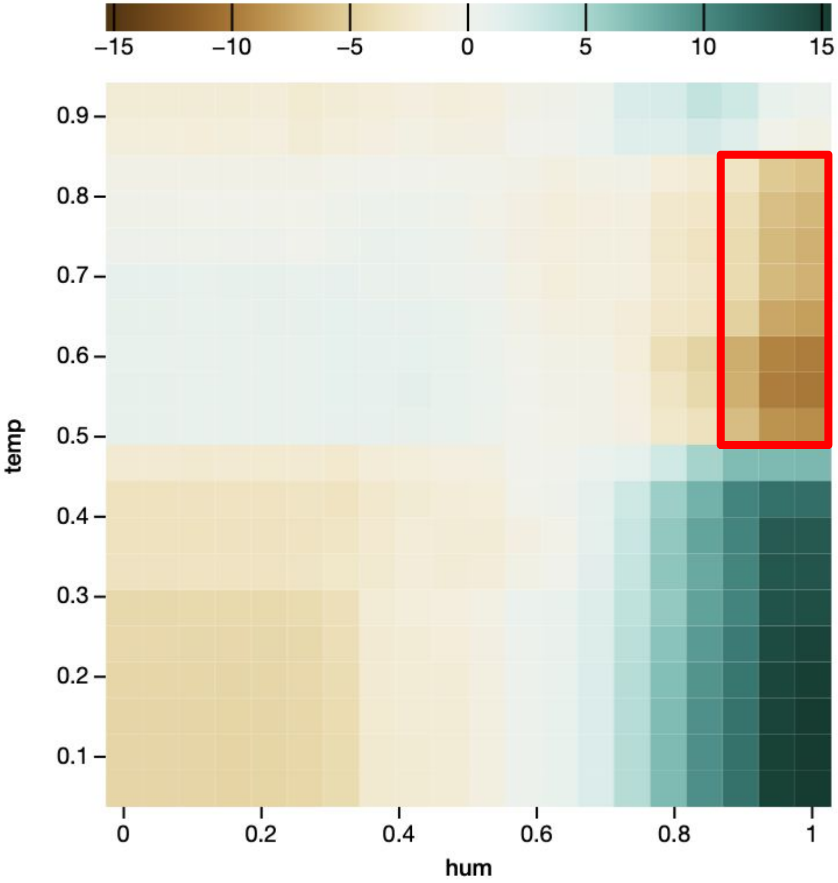
Interactions plot



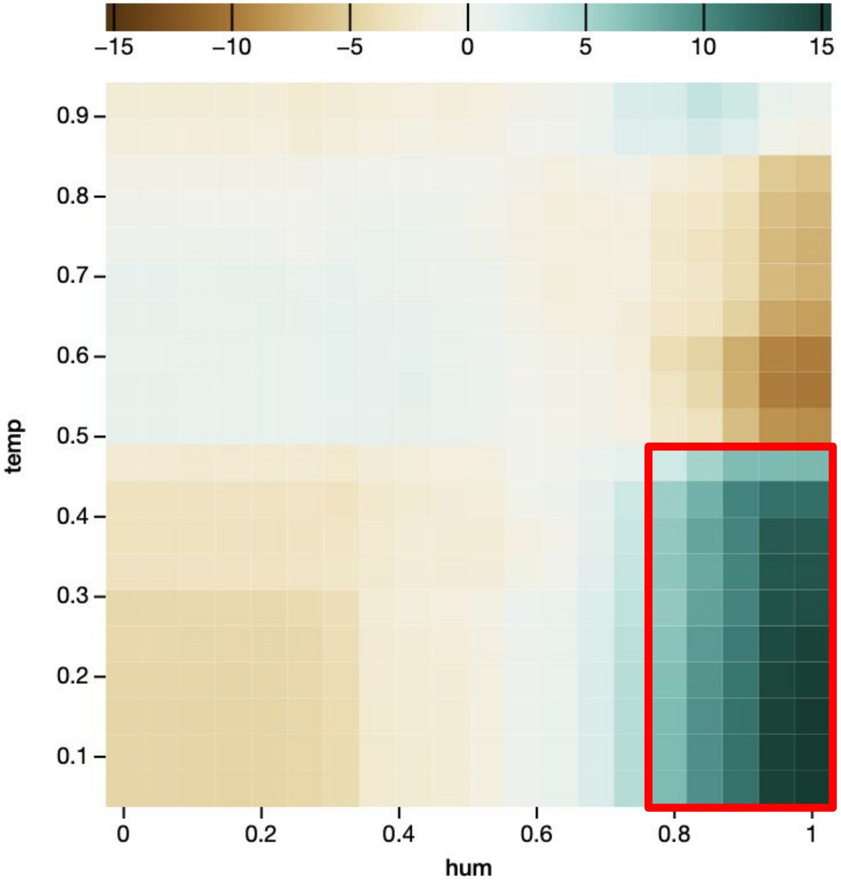
The color scale indicates whether the interaction between the two features makes the model's average prediction for the given values **lower** or **higher**.



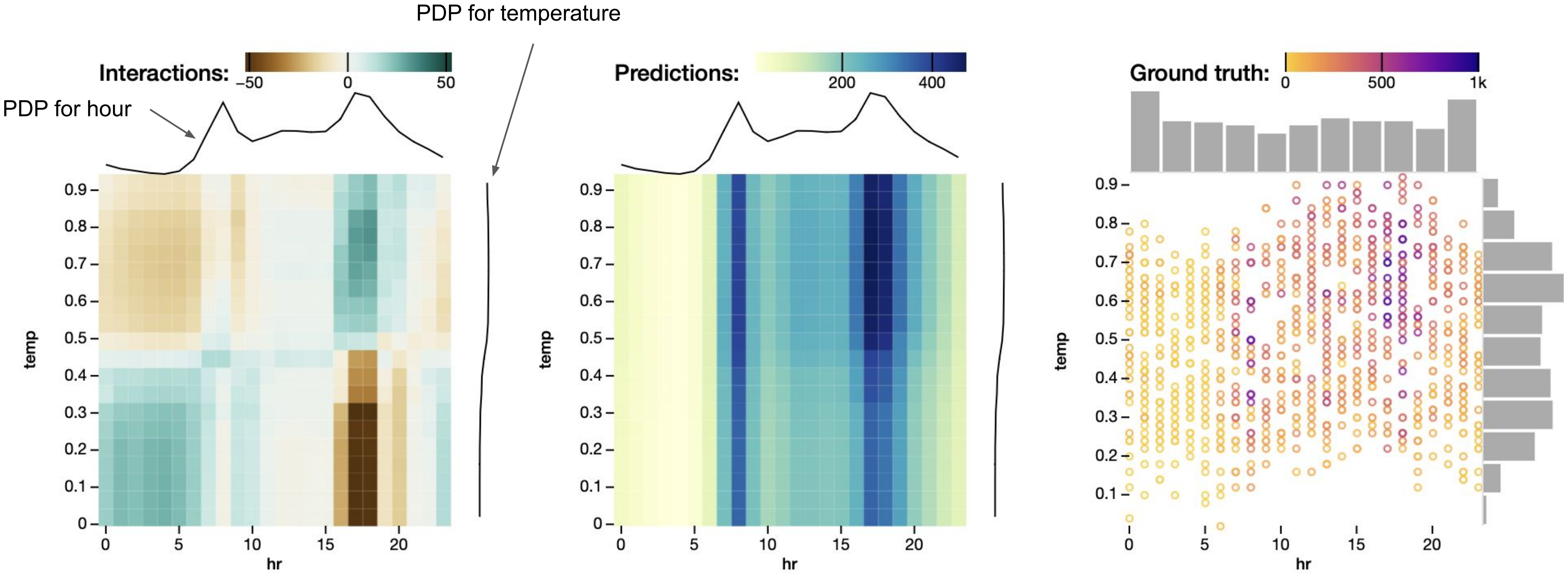
At 8am on non-working days, the interaction effects lower the model’s predictions by about 200 bikes, on average.



When there is high humidity and high temperature, the interaction effects lower the prediction by about 5 to 10 bikes, on avg.



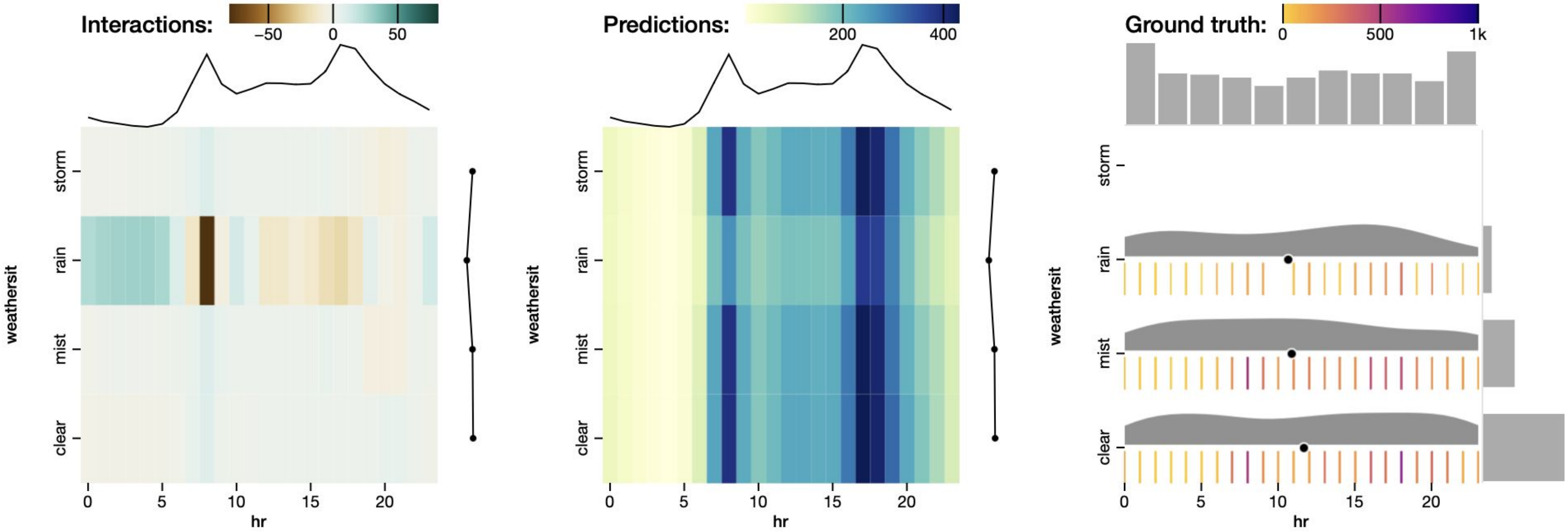
When there is high hum. and low temp, the interaction effects increase the model’s prediction by about 5 to 15 bikes, on avg.



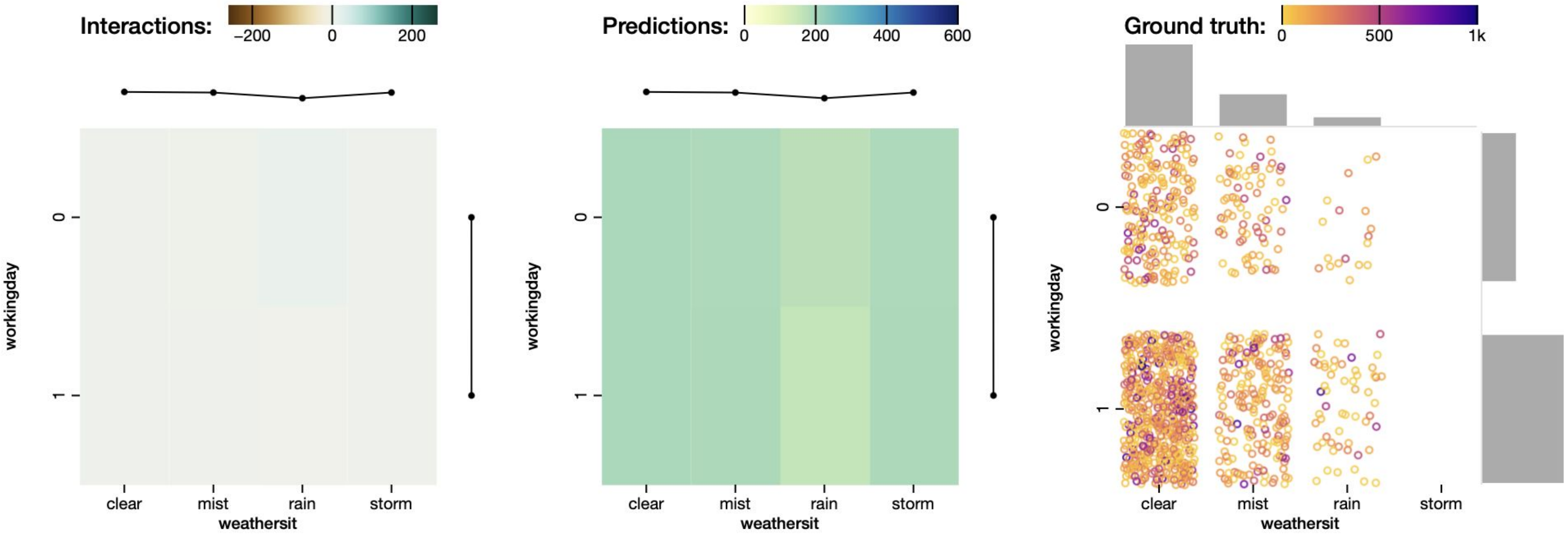
PDP for hour

PDP for temperature

For a given pair of features, we can also show the relationship between the features values' and the ground truth labels. When the features are quantitative, this is visualized with a scatterplot.



When one feature is quantitative and one is categorical, this is visualized with raincloud plots.



When both features are categorical, this is visualized with a categorical scatterplot.

Interview

- What is your current approach to using PDP or ICE plots?
    - Do you use both PDP and ICE plots?
    - Do you use both one-way and two-way plots?
    - What tasks do you use them for?
    - How do you determine which plots to look at?
    - Are there any pain points in your approach?

- How did PDPilot support (or not support) your model analysis?
    - Were there any questions that you were unable to answer?
    - Were there any tasks you were unable to perform?

- Were the visualizations useful?
    - Were any of them unclear?

- What impact did the different rankings have on your model analysis?
    - Which rankings did you find to be the most useful?
    - Which rankings did you find to be the least useful?
    - Are there any additional ways that you think would be helpful to rank the plots?

- How did analyzing subsets or clusters of instances impact your analysis?
    - Was the clustering useful?
    - Was the highlighting useful?

- Were the filtering capabilities useful to your analysis?
    - Are there any additional ways to filter the plots that you think would be helpful?

- How well did the tool enable you to analyze feature interactions?

- What are PDPilot’s biggest weaknesses or limitations?
- Are there any improvements or additional capabilities that you would want PDPilot to have?

- Do you have any other feedback about PDPilot?

Conclusion

- Thank you!
- We will send you your compensation soon.
- PDPilot is open source and installable through pip.
- Let us know if you are interested in using PDPilot for your work or with your own data and model.
    - We are happy to answer any questions or help you get set up.
    - We can make changes to PDPilot to support your needs.
    - We are interested in hearing about your experience, findings, and feedback.