Your Paper

You

July 2, 2023

1 Model Training

In order to identify optimal models for our datasets, we employ a multi-step approach using several techniques. All models are implemented in PyTorch, with hyperparameter tuning facilitated by Ray.

The following steps are applied consistently to train models addressing RQ1:

1. We first establish a hyperparameter search space, which includes the following parameters:

   - **Batch size**: For the toy dataset, we consider values in the range [128, 256, 512, 1024, 2048]. For the remaining datasets, we use the range [8, 16, 32, 64, 128]. The toy dataset has a different range due to its larger size.
   - **Learning rate**: We use a loguniform distribution between 1e-7 and 1e-2.
   - **Epochs**: For the toy dataset, we consider values in the range [10, 25, 50, 75]. For the other datasets, we use the range [25, 50, 100, 150, 300, 500].
   - **Hidden layers**: We select a random integer between 1 and 4.
   - **Hidden units**: For each layer, we consider values in the range [16, 32, 64, 128].
   - **Dropout**: We select either True or False.
   - **Dropout array**: If dropout is set to True, we select a uniform value between 0.1 and 0.8 for each layer.
   - **BatchNorm**: We select either True or False to determine whether a BatchNorm1d layer is included at the beginning.

   Next, we train 100 models with hyperparameters sampled from this search space. We employ 5-fold cross-validation to calculate the R^2 score and loss for each model. The ASHAScheduler is utilized for early stopping in cases where the R^2 score is particularly low.

2. We then select the hyperparameters corresponding to the highest R^2 score and proceed to fine-tune the learning rate, batch size, and epochs using scikit-optimize. Specifically, we employ the gp_minimize function, which performs Bayesian optimization with a Gaussian Process. We run 100 iterations for this optimization process.

3. Armed with these optimized parameters, we train a new model, incorporating early stopping to prevent overfitting. The resulting model is saved as our final optimized model.

We have also established a consistent train and test set across all models of the same dataset, both with and without confounding variables. This ensures a fair comparison when evaluating the performance of different models.

1.1 Bike rental

The optimal architecture that emerged from this process consists of three hidden layers with 128, 128, and 64 units respectively. More details on this architecture can be found in Table 1. Further information regarding the other hyperparameters is provided in the appendix (see Appendix **??**). The performance of the model is summarized in table 2.

Table 1: Neural Network Structure

Layer (type)	Output Shape	Param #
Linear-1	[-1, 128]	1,920
ReLU-2	[-1, 128]	0
Linear-3	[-1, 128]	16,512
ReLU-4	[-1, 128]	0
Linear-5	[-1, 64]	8,256
ReLU-6	[-1, 64]	0
Linear-7	[-1, 1]	65
	Total params:	26,753
	Trainable params:	26,753
	Non-trainable params:	0

Table 2: Train and test Losses with R^2 Score where the R^2 Score is calculated on the test set.

Metric	**Value**
Train RMSE Loss	575.0354
Test RMSE Loss	701.2170
R^2 Score	0.8523403716393165

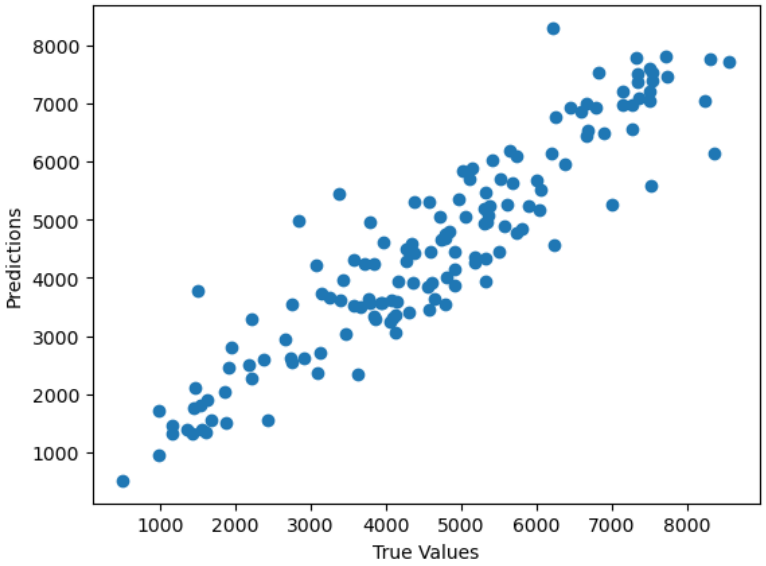

Figure 1: This figure shows the true value and the prediction of the model.

1.2 Bike rental confounder

The optimal architecture that emerged from this process consists of three hidden layers with 32, 32, and 128 units respectively. More details on this architecture can be found in Table 3. Further information regarding the other hyperparameters is provided in the appendix (see Appendix **??**). The performance of the model is summarized in table 4. We set up a specific confounding scenario by excluding all season-related variables. As the results indicate, this change led to less optimal outcomes. In 8 out of 10 trials, we observed similar performance. However, the remaining two trials showed markedly different results, either performing significantly better or worse.

Table 3: Neural Network Structure Confounder

Layer (type)	Output Shape	Param #
Linear-1	[-1, 32]	352
ReLU-2	[-1, 32]	0
Linear-3	[-1, 32]	1,056
ReLU-4	[-1, 32]	0
Linear-5	[-1, 128]	4,224
ReLU-6	[-1, 128]	0
Linear-7	[-1, 1]	129
		Total params: 5,761
		Trainable params: 5,761
		Non-trainable params: 0

Metric	**Value**
Train Loss	838.7948
Test Loss	875.8113
R^2 Score	0.7770114486751998

Table 4: Train and Test Losses with R^2 Score

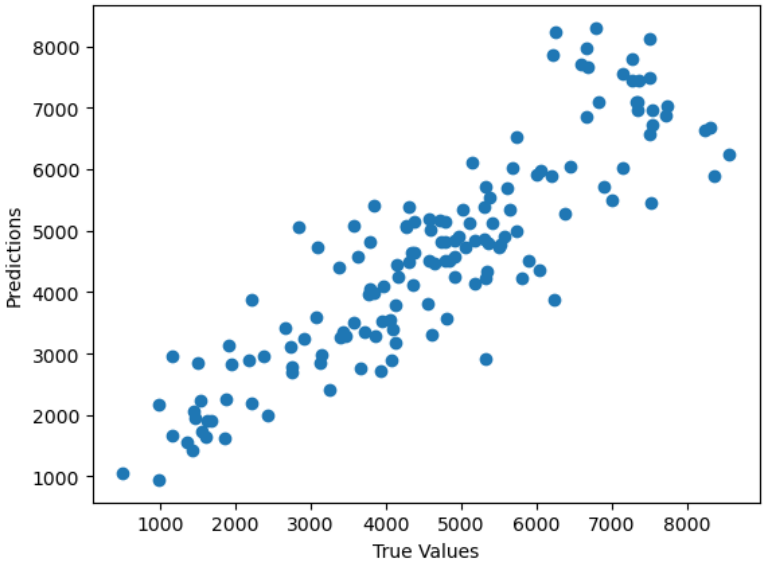

Figure 2: This figure shows the true value and the prediction of the model where we a season as a confounding variable.

1.3 Admission

The optimal architecture that emerged from this process consists of three hidden layers with 128, 16, and 32 units respectively. More details on this architecture can be found in Table 5. Further information regarding the other hyperparameters is provided in the appendix (see Appendix **??**). The performance of the model is summarized in table 6.

Table 5: Neural Network Structure

Layer (type)	Output Shape	Param #
BatchNorm1d-1	[-1, 7]	14
Linear-2	[-1, 128]	1,024
ReLU-3	[-1, 128]	0
Linear-4	[-1, 16]	2,064
ReLU-5	[-1, 16]	0
Linear-6	[-1, 32]	544
ReLU-7	[-1, 32]	0
Linear-8	[-1, 1]	33
	Total params:	3,679
	Trainable params:	3,679
	Non-trainable params:	0

Table 6: Train and Test Losses with R^2 Score Calculated on the Test Set.

Metric	**Value**
Train Loss	0.0705
Test Loss	0.0650
R^2 Score	0.709402011795968

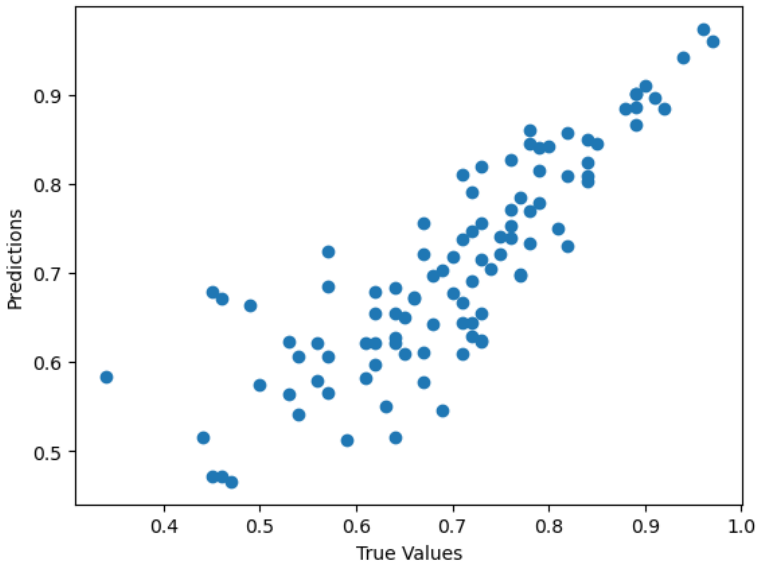

Figure 3: This figure shows the true value and the prediction of the model.

1.4 Admission confounder

The optimal architecture that emerged from this process consists of only one hidden layer with 128 units. More details on this architecture can be found in Table 7. Further information regarding the other hyperparameters is provided in the appendix (see Appendix **??**).

The performance of the model is summarized in table 8.In this confounding setup, we choose to remove the CGPA variable which now plays the role of a hidden confounder.

Table 7: Neural Network Structure

Layer (type)	Output Shape	Param #
BatchNorm1d-1	[-1, 6]	12
Linear-2	[-1, 128]	896
ReLU-3	[-1, 128]	0
Linear-4	[-1, 1]	129
	Total params:	1,037
	Trainable params:	1,037
	Non-trainable params:	0

Metric	**Value**
Train Loss	0.0662
Test Loss	0.0738
R^2 Score	0.66212

Table 8: Train and Test Losses with R^2 Score

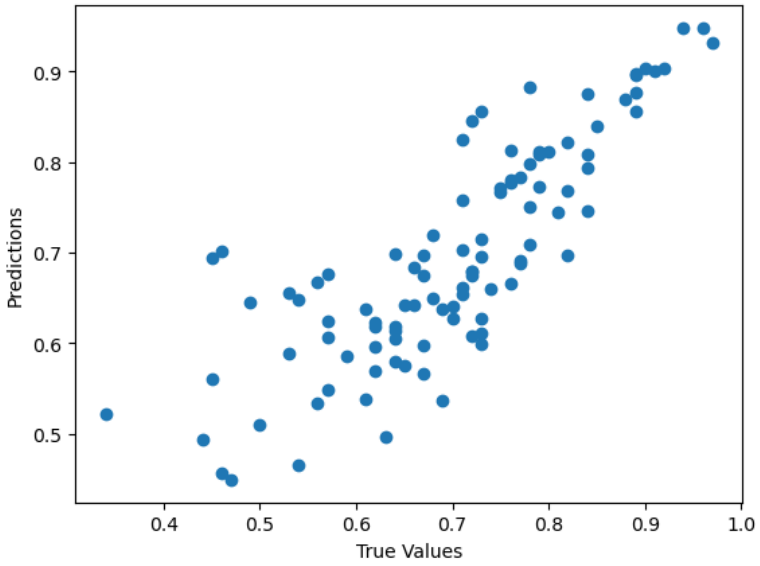

Figure 4: This figure shows the true value and the prediction of the model where we have CGPA as a confounding variable.

1.5 Toy-Data

The optimal architecture that emerged from the described process for the toy dataset consists of three hidden layers with 128, 64, and 32 units respectively. More details on this architecture can be found in Table 9. Further information regarding the other hyperparameters is provided in the appendix (see Appendix **??**).

The performance of the model is summarized in table 10.

Table 9: Neural Network Structure

Layer (type)	Output Shape	Param #
Linear-1	[-1, 128]	896
ReLU-2	[-1, 128]	0
Linear-3	[-1, 64]	8,256
ReLU-4	[-1, 64]	0
Linear-5	[-1, 32]	2,080
ReLU-6	[-1, 32]	0
Linear-7	[-1, 1]	33
	Total params:	11,265
	Trainable params:	11,265
	Non-trainable params:	0

Metric	**Value**
Train Loss	1.9447
Test Loss	2.1685
R^2 Score	0.998

Table 10: Train and Test Losses with R^2 Score

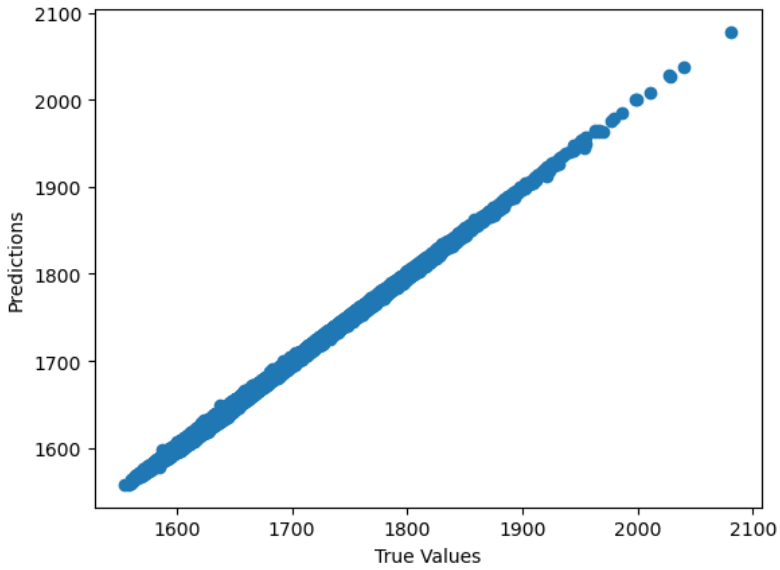

Figure 5: This figure shows the true value and the prediction of the model using all variables.

1.6 Toy-Data Confounder

The optimal architecture that emerged consists of three hidden layers where the first layer has 64 units and the second and third layers both have 32 units. More details on this architecture can be found in Table 11. Further information regarding the other hyperparameters is provided in the appendix (see Appendix **??**).

The performance of the model is summarized in table 12.In this confounding setup, we choose to remove the variable $x1$ which now plays the role of a hidden confounder.

Table 11: Neural Network Structure

Layer (type)	Output Shape	Param #
BatchNorm1d-1	[-1, 5]	10
Linear-2	[-1, 64]	384
ReLU-3	[-1, 64]	0
Linear-4	[-1, 32]	2,080
ReLU-5	[-1, 32]	0
Linear-6	[-1, 32]	1,056
ReLU-7	[-1, 32]	0
Linear-8	[-1, 1]	33
	Total params:	3,563
	Trainable params:	3,563
	Non-trainable params:	0

Metric	**Value**
Train Loss	10.0235
Test Loss	7.4154
R^2 Score	0.978

Table 12: Train and Test Losses with R^2 Score

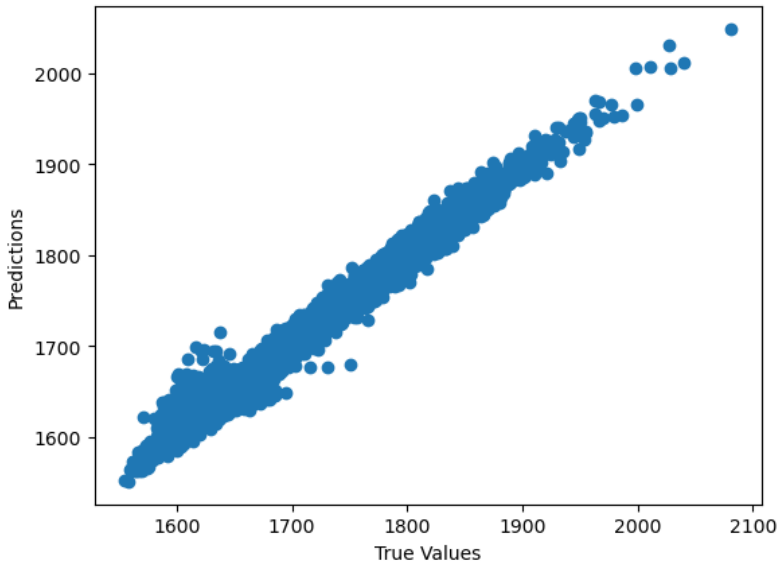

Figure 6: This figure shows the true value and the prediction of the model where we have $x1$ as a confounding variable.

2 ICE-Plots

Individual Conditional Expectation (ICE) plots offer an intuitive way of visually representing how a specific prediction changes as a given feature changes. Developed by Goldstein et al. in 2017, these plots are designed to counter a limitation of the Partial Dependence Plots (PDP), which only show the average effect of a feature without focusing on individual instances. An ICE plot, on the other hand, provides a detailed view of the relationship between the prediction and the feature for each instance separately. This is done by changing the feature of interest and holding every other feature fixed and plotting the resulting prediction of the changed data point.

The mathematical definition of the ICE plot can be formally expressed as follows:

Given a dataset, where each instance is represented as a pair (x_i^S, x_i^C) for $i = 1, ..., N$, where x_i^S is the variable of interest that we want to change to different values and x_i^C is part of the instance which is fixed. We now calculate $\phi(x_j^S, x_i^C)$ for $j = 1, ..., N$ which will be the ICE line for the sample i.

Another way in which the ICE line could be computed is to calculate the range of the feature x^S as $min_i(x_i^S)$ and $max_i(x_i^S)$ and split the space evenly between them and evaluate the model on this grid points then. In this thesis, we will however use the first approach which only uses observed values as stand-ins.

Here is an example of what an ICE plot looks like:

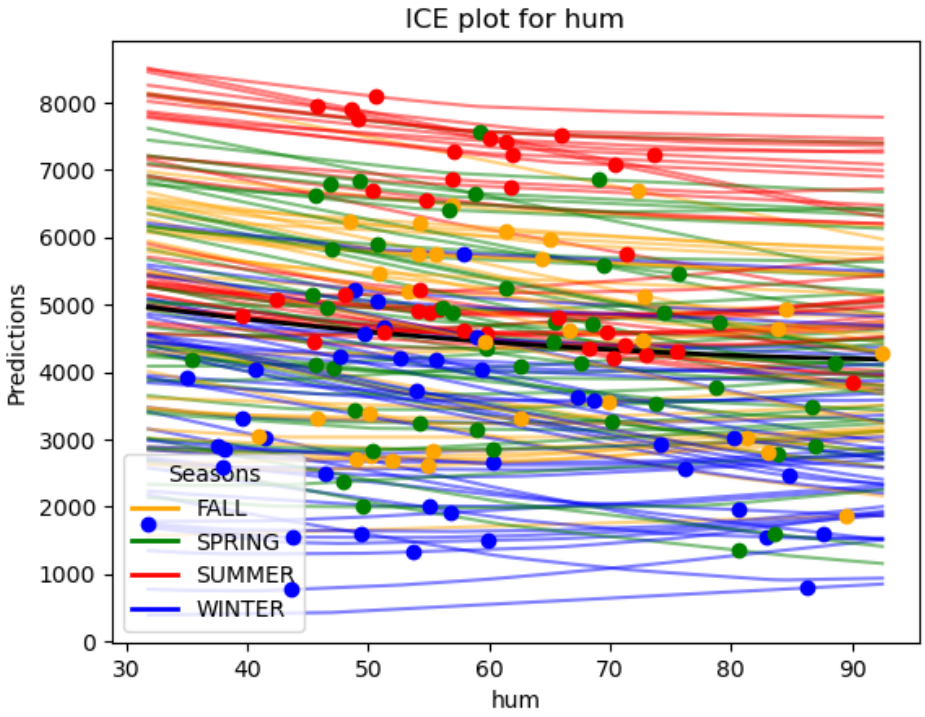

Figure 7: This is a typical ICE plot. The black line is the average of all lines (PDP line). The dots represent the original value of the instance.

2.1 Interpretation

Interpreting an ICE plot involves understanding the variations in the lines and what these imply about the model's predictions.

Trend and direction: The direction of the lines (upward, downward, horizontal) can indicate how changes in the feature affect the prediction. For instance, an upward trend suggests that an increase in the feature value leads to an increase in the predicted outcome while a horizontal line indicates that there is not much interaction and the outcome does not depend much on this variable.

Slope of the lines: The slope of the lines provides information about the rate of change in the prediction as the feature value changes. A steep slope indicates a high sensitivity of the prediction to changes in the feature value.

Variation across instances: Differences in the lines across instances (i.e., heterogeneity of the lines) can suggest interactions between the feature of interest and other features. If lines for different instances follow vastly different paths, it may be an indication that the effect of the feature of interest is not consistent across instances, potentially due to interactions with other features.

2.2 Advantages

The advantages of ICE plots are that they are very intuitive to understand. They also give an insight into the behavior of individual data points compared to PDP. ICE plots can uncover heterogeneous

relationships that PDPs might obscure.

2.3 Disadvantages

While ICE plots offer an intuitive and robust way to visualize the impact of a feature on model predictions for individual instances, they do come with some limitations.

Firstly, the nature of ICE plots restricts us to examining one feature at a time. This could be limiting when trying to understand the interplay between multiple features in high-dimensional datasets.

Secondly, when dealing with large datasets, the visualization may become cluttered due to the sheer volume of lines plotted. This high density of data points can make it challenging to discern clear patterns or interpret the plot.

Finally, the approach used to create the grid for model evaluation can introduce potential issues. By using a uniform distribution within the feature range to establish the grid, we risk generating invalid data points that do not align with the original distribution of the feature. This could lead to an inaccurate representation of the feature's influence and thus distort the interpretability of the model's behavior.

3 Model interpretation

3.1 Bike rental

The first interpretation we will do is on the prediction-truth plot. In figure 10 we see both plots and observe that the full model is better than the confounding model (more linear appearance). One thing which pops out is that the confounder model seems to overestimate some days in summer as seen in the top right of the plot 9.

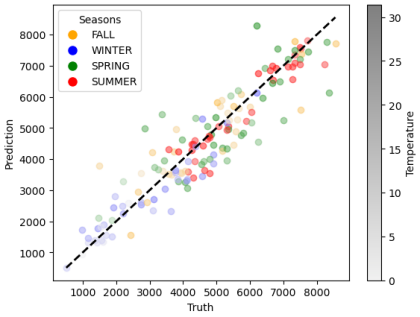

Figure 8: This image shows the prediction-truth plot of the full model. Notably, the model exhibits a good performance during the summer months, as demonstrated in the image. The depicted trend line is seen to closely align with the actual data points, signifying accurate predictions during this period.

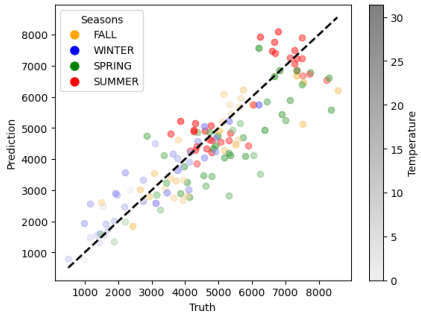

Figure 9: This image contrasts the performance of the full model with that of the confounding model during the summer months. Unlike the full model, the confounding model consistently overpredicts temperatures on hot summer days. Nearly every point on the plot that deviates above the line represents a time the model overestimated the count. This tendency to overestimate could be a result of the model not having a way to account for seasonal variations, such as the absence of a specific attribute for the summer season.

Figure 10

3.1.1 ICE-Plots

In the following section, we will present a series of Individual Conditional Expectation (ICE) plots for examination and discussion. Our analysis will start with an exploration of one of the most significantly

correlated features in our study - the temperature.

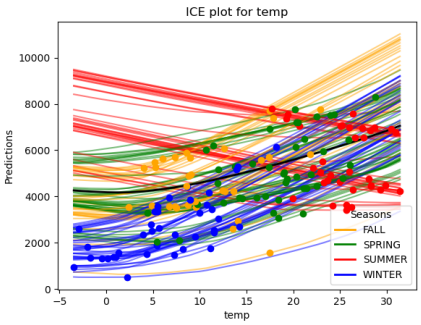

Figure 11: Here we see the ICE plot for the full model. It is interesting to see that the model learned a negative trend of the temperature in summer and a positive trend for all other seasons. This also shows that the model is invalid for data points outside of the distribution of the training data since it does not make sense that the company would rent the most bikes on a summer day at -5 degrees. This downward trend helps the model to not overestimate very hot summer days thus producing a better performance.

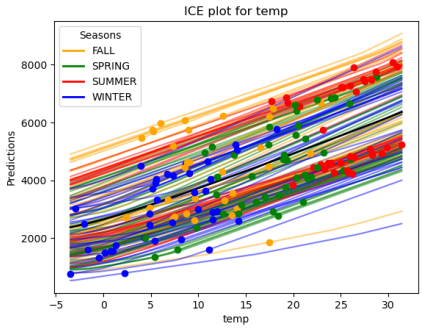

Figure 12: The subsequent image presents the Individual Conditional Expectation (ICE) plot derived from the confounding model. Notably, an upward trend is observed for each season. However, this seems impractical. As temperature exceeds 30 degrees, it is reasonable to expect a decrease in bike rentals due to the excessive heat making cycling a less appealing activity.

Figure 13: These are the two ICE plots for the bike-rental dataset, the temperature is the only variable allowed to change while holding all other variables fixed. The average trend across all individual lines is represented by the black line in each plot.

Since we see an upward trend (of the average line) we can infer that the temperature is an important variable of the model in both cases. As described the season attribute seems to allow for a more fine-grained learning of the model to differentiate between the 4 seasons.

The next attribute we look at is the year. Since we know that we have a quite strong upwards trend of rented bikes from 2011 to 2012 we want that to be seen in the ICE plots 16.

The plots indicate that the full model captures the importance of the year better. But this is not the full story. We saw that the importance of variables can be distributed to different highly correlated features. The same thing happens here if we also look at the ICE plots of the variable days_since_2011. Since we can accurately calculate the year from this all the information of the year is also available in days_since_2011. From that point of view, this variable might be an even better variable. The ICE plots 19 together with 16 show this importance shifting perfectly.

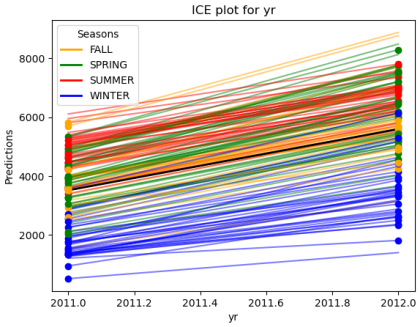

Figure 14: In this ICE plot we see a relatively good mapping from 2011 to 2012. Broadly speaking we map for example winter season 2011 to winter season 2012 and so on. The average black lone indicates also that we increase the count when changing from 2011 to 2012 which aligns with what we know to be true in reality.

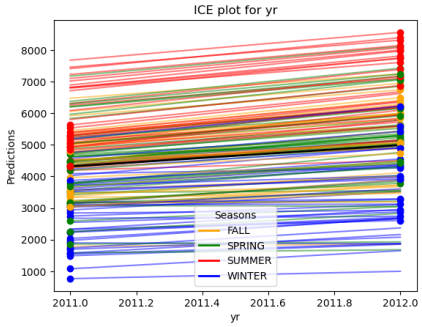

Figure 15: The ICE plot for the confounding model on the other hand shows a worse mapping and a very low upwards trend from 2011 to 2012 compared to the full model. We observe that if we change the year from a hot summer day in 2012 to a hot summer day in 2011 we will end up way higher than what we would predict for a hot summer day in 2011

Figure 16

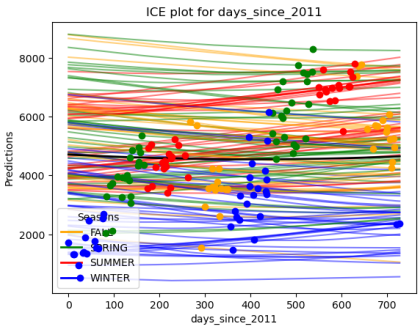

Figure 17: This shows the ICE plot of the full model.

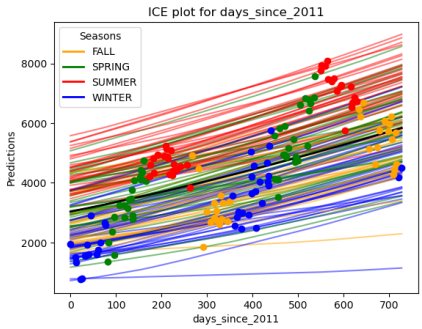

Figure 18: This shows the ICE plot of the confounder model.

Figure 19: In comparing the two charts, it's evident that the 'full model' displays mostly flat lines with no uniform trend for Individual Conditional Expectation (ICE), while the 'confounder model' shows an upward trend. We also see a reversal in the ICE plots for the year, with the confounder model now mapping summer 2011 to summer 2012 roughly, but the full model does not. Alongside the ICE plot for the year, it seems logical to suggest that the full model places greater emphasis on the year to understand the trend, whereas the confounder model gives more weight to the 'days_since_2011' factor

3.1.2 Feature-importance / SHAP values

In this section, we will discuss and present SHAP values as a tool for interpreting individual data points to gain an understanding of the impact of different features.

Initially, we'll focus on the day with the highest and lowest number of rented bikes. By utilizing the SHAP waterfall plot, we'll make a comparison against the baseline, which is the mean of all data points. (Note that we only use the test set for this analysis.)

The baseline has the following values for each feature :

Feature	Value
yr	2011.510204
temp	15.322261
hum	60.256324
windspeed	13.208245
days_since_2011	361.142857
season_FALL	0.210884
season_SPRING	0.292517
season_SUMMER	0.238095
season_WINTER	0.258503
holiday_HOLIDAY	0.020408
workingday_WORKING DAY	0.700680
weathersit_GOOD	0.639456
weathersit_MISTY	0.346939
weathersit_RAIN/SNOW/STORM	0.013605

Table 13: Baseline summary

Predictions from our models of the baseline indicate 1690.68 for the full model and 2766.26 for the confounding model. Importantly, please note that the input for the confounding model does not incorporate season variables.

From the test data set, the maximum number of bikes rented in a single day reached 8555, while the minimum dropped to 506.

To better visualize the impact of various features on these extreme days, we turn to SHAP analysis. Figure 90 presents the SHAP waterfall plot for the day with the highest number of bike rentals. This graph allows us to see how each feature contributes towards pushing the model's prediction away from the baseline.

When we compare Figures 90 and 92, we can see that the confounder model predicts fewer rentals, possibly because it doesn't include seasonal data. This is highlighted by how the full model boosts its prediction using season data. Both plots hint at a year-on-year increase in rentals. This suggests that the growth in rentals might be due to factors like the rising popularity of bike rentals, rather than just seasonal or weather factors.

The next two figures 22 and 23 show the SHAP waterfall plot for the day with minimum Bike rentals, one for each model. We see that both the weather situation and the temperature play a big role in both of them. Interestingly the reduction of bikes because of the year is now bigger in the confounding model than in the full model even though we saw that the trend is stronger in the full model. This might be the case because the full model has a lower baseline prediction than the confounding model.

Reflecting on the baseline data, it's evident that temperature had a more significant impact on the worst day with lower rentals than on the highest rental day. This is because the baseline temperature (15.3°C) is much closer to the temperature on the highest rental day (17.5°C). In contrast, the temperature on the lowest rental day is substantially lower, at just 2.2°C.

SHAP Waterfall Plot for the Day with Maximum Bike Rentals (full model)

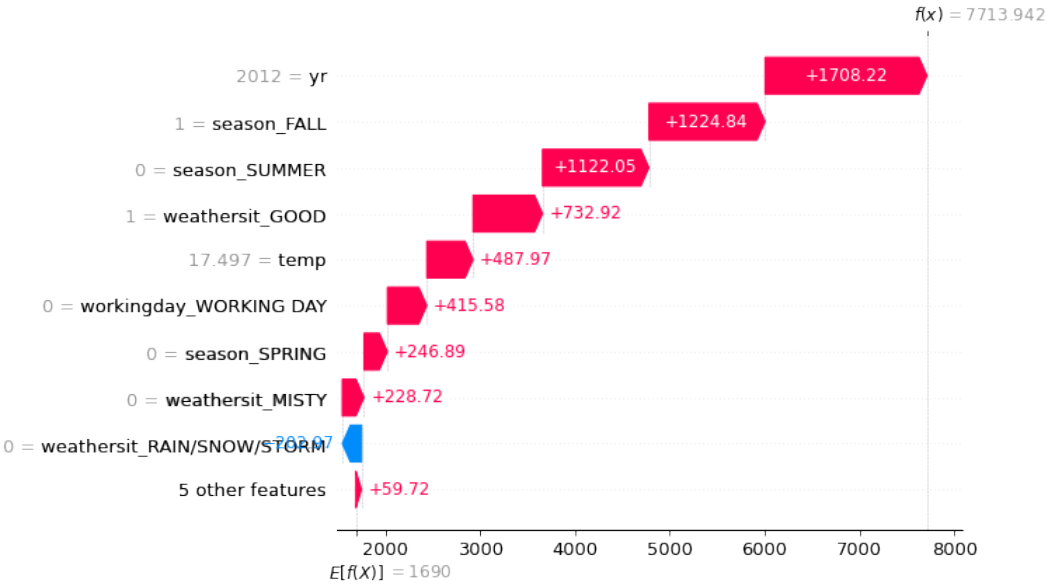

Figure 20: We can see that the year is the most important factor for this data point as it adds 1708.22 rented bikes to the total prediction. This is expected since we saw a high trend when we looked at the ICE plots. We also observe that we have two season attributes which add a lot of bikes.

SHAP Waterfall Plot for the Day with Maximum Bike Rentals (confounder model)

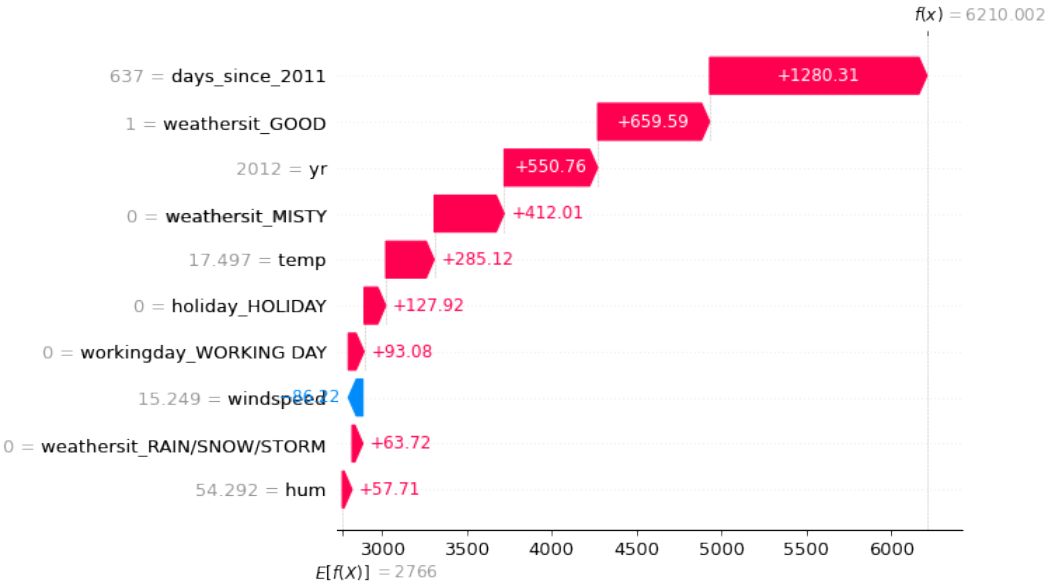

Figure 21: We see that here day_since_2011 is the most important attribute which is also expected since we saw earlier that the confounding model puts more importance on this variable instead of the year. Adding both year and d.s.2011 we see that they are not far off between the two models.

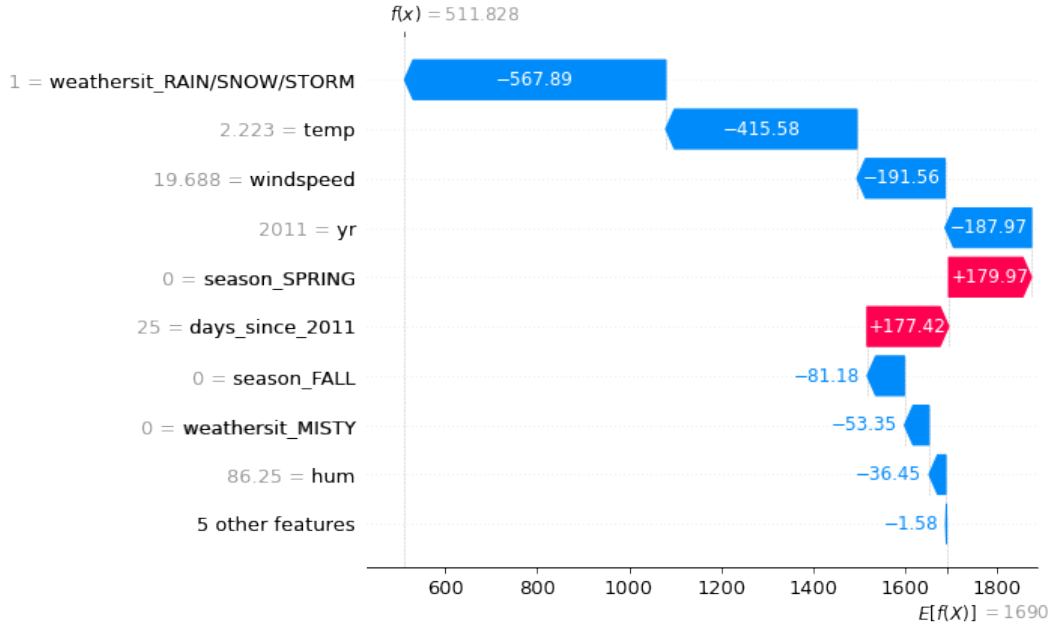

Figure 22: Becasue of the bad weather the number of rented bikes is very low. The model also removes a lot of bikes because of the near-freezing temperature.

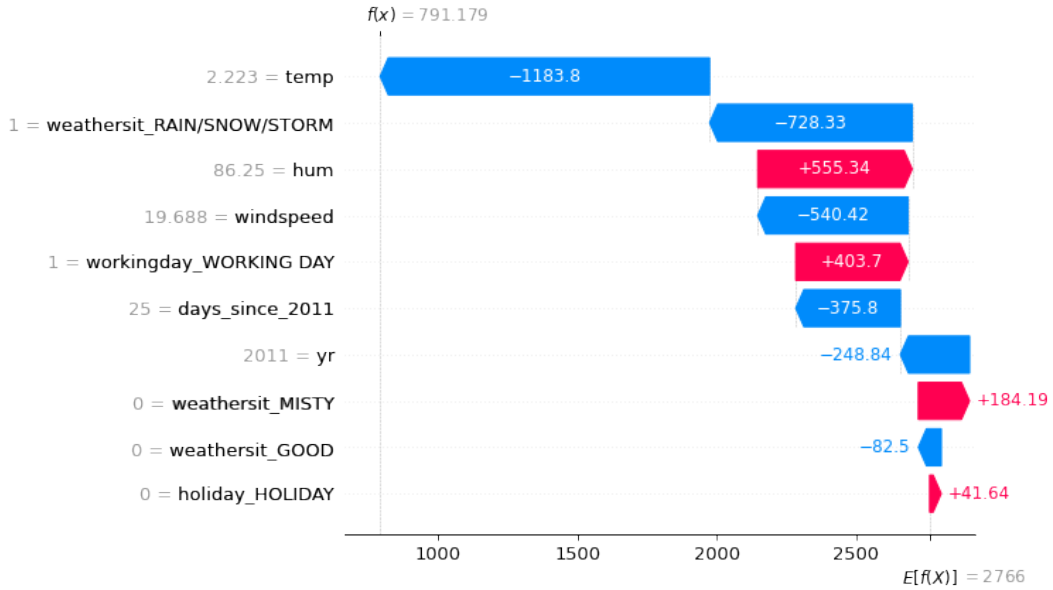

Figure 23: We observe the temperature as the most impacting feature for this data point. Again the d.s.2011 variable is stronger than the year for the confounder model.

Next, we look at a random data point from each season and compare it to the baseline which is again the mean of each season.

Table 14 displays the baseline for each season. It is the average over all data point in this season.

Attribute	Fall	Spring	Summer	Winter
Year	2011.516	2011.512	2011.457	2011.553
Temperature	10.94	18.00	25.50	6.49
Humidity	63.03	62.07	59.64	56.51
Windspeed	12.03	12.95	11.23	16.27
Days since 2011	505.42	313.05	385.17	275.74
Holiday	0.032	0.023	0.000	0.026
Working day	0.613	0.674	0.743	0.763
Good Weather	0.645	0.581	0.743	0.605
Misty Weather	0.323	0.419	0.257	0.368
Rain/Snow/Storm	0.032	0.000	0.000	0.026

Table 14: Seasonal averages of different attributes

The predictions of the baselines for both models are provided in table 15.

Season	Full Model Prediction	Confounder Model Prediction
Fall	3513.96	2724.77
Spring	3963.55	2861.39
Summer	4921.54	4957.58
Winter	1572.58	1224.40

Table 15: Comparison of Predictions for Full Model and Confounder Model Across Seasons

We will start with winter and plot the SHAP waterfall plot for a random day in winter. The true count is 2177. Note that we do not observe very influential season variables which might lead to a similar prediction by the confounding model since it seems the prediction does not rely on season data.

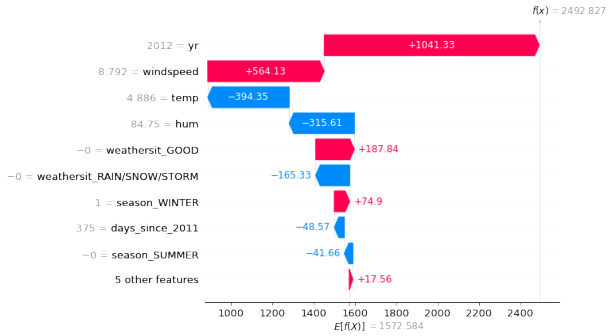

Figure 24: Again we see the year as the most important feature. The low wind speed compared to the baseline has also a big influence but the low temperature lowers the predicted count again.

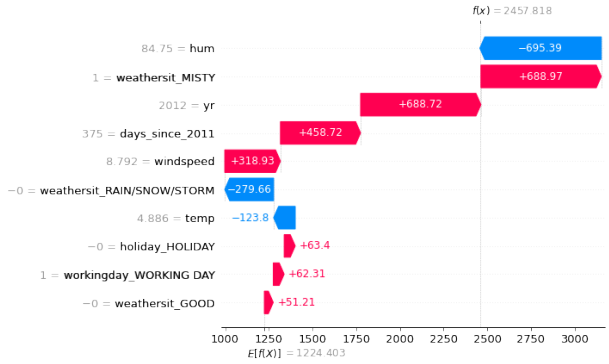

Figure 25: Surprisingly the humidity plays a huge part in this prediction. Compared to the baseline we have a much higher humidity which reduces the count. Again we have a strong trend of the year and d.s.2011 which added together is comparable to the full model.

Figure 26: These are the SHAP plots for a random day in winter for both models in comparison to the average winter day.

Next, we look at the Spring plots represented in figures 27 and 28. The true value for this day was 4803. Since this random day is in 2011 we see that the prediction gets lowered by a good amount in both cases. Again we see that for one model this is done via the variable year (full model) and

via days_since_2011 in the confounding model. In both cases, the good weather situation leads to an increase in the prediction of rented bikes for this day.

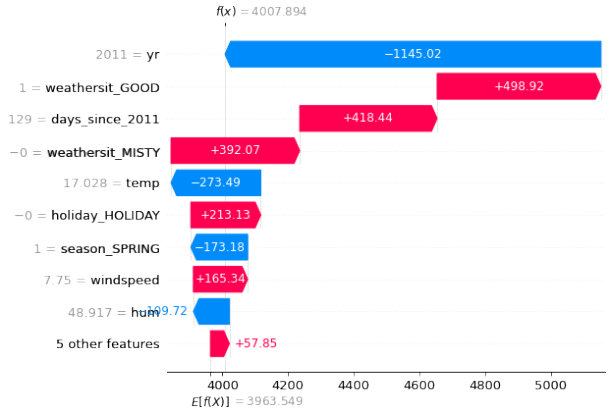

Figure 27: We see that year is the most important feature. The full model also takes the season into account and reduces the prediction by 173.18 bikes based on that.

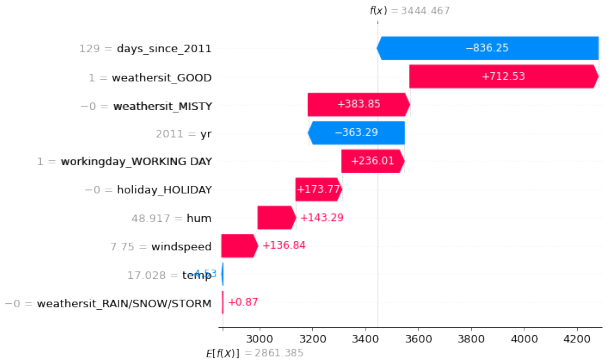

Figure 28: The confounding model also indicates that because of the low days_since_2011 value the prediction needs to be reduced from the average by a lot. Additionally, we observe that favourable weather conditions positively influence the predictions, reaffirming the pattern identified in the full model.

Figure 29: These are the two SHAP plots for a random day in Spring. Figure 27 represents the full model and figure 28 is the confounding model.

In the summer SHAP plots, we observe the already seen trend of the full model to lower the count the higher the temperature is in the full model. Because we are hotter than the average in summer we receive a lower count in contrast to the confounder model where we add a lot more rented bikes to our prediction. This leads to the overestimation we already saw. The true count is 4590. If the confounder model would not add all the bikes because of the temperature it would be much closer to the true value. This indicates again that the season is an important variable we want to model. The ICE plots already indicated this learned behavior.

The final plots in this chapter depict SHAP waterfall plots for a randomly selected day in autumn. While the plots may seem alike, each model emphasizes different features. Unfortunately, both models tend to overestimate the true count, suggesting a higher number than the actual figure of 5146.

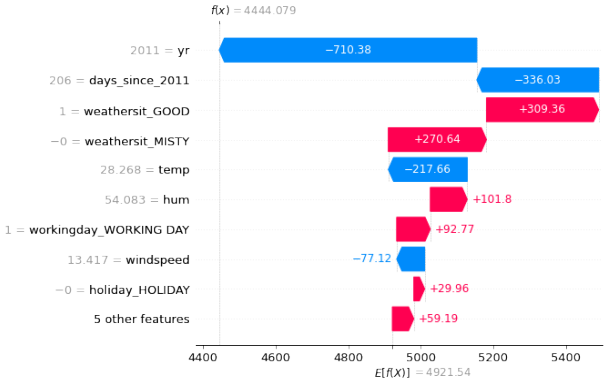

Figure 30: The most interesting aspect of this figure is the reduced prediction even though the temperature is high. In contrast to other plots (for example **??**) where we saw that an increase in temperature leads to an increase in prediction, the full model can differentiate between the seasons and indicates that for a very hot summer day, there will be fewer bikes rented because of the hot temperature.

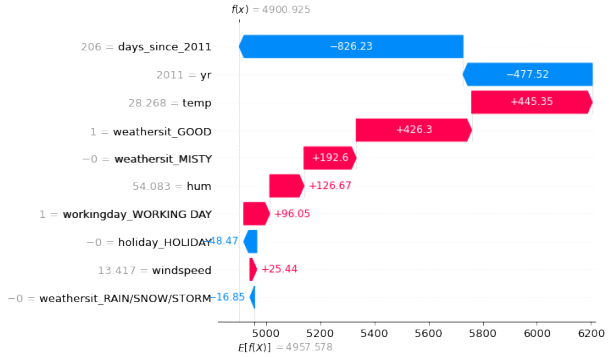

Figure 31: Here we see the exact opposite of the full model. The confounding model increases the prediction of rented bikes by 445.35 because of the temperature and thus overestimating the true count. This plot also shows clearly the switched roles for the year and days_since_2011 for the models.

Figure 32: These are the two SHAP plots for a random day in Summer. The left picture corresponds to the full model and the right one to the confounding model.

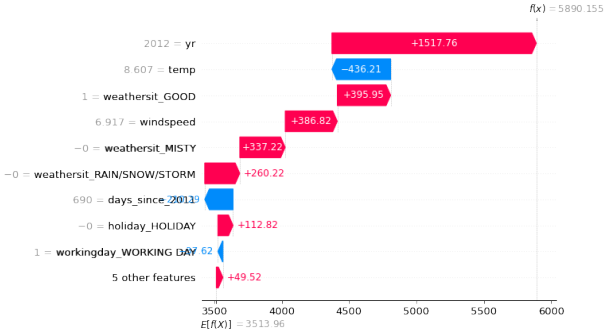

Figure 33: We see that again the year is very important. Because of the lower temperature than the average we see a reduction in the predicted rented bikes.

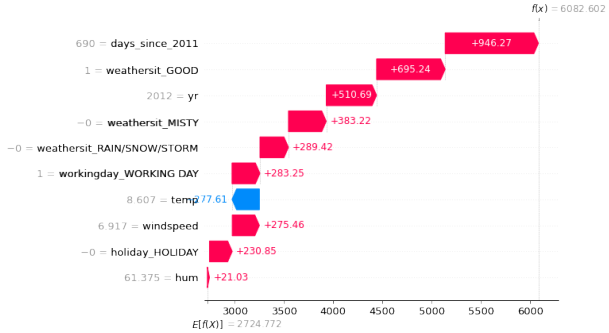

Figure 34: In this plot the weather situation plays a bigger role compared to the full model.

Figure 35: These are the two SHAP plots for a random day in autumn. The left picture corresponds to the full model and the right one to the confounding model.

Conclusion

In our analysis, we have identified varying levels of importance for different features in the models. Specifically, the full model heavily relies on the "year" feature, while the confounding model utilizes the "days_since_2011" feature. This observation is consistent across all plots and indicates a significant issue with multicollinearity, as these features are highly correlated. In fact, one feature can be precisely calculated using the other, as demonstrated by the equation:

$$year = 2011 + \mathbb{1}_{days\_since\_2011>356}$$

where $\mathbb{1}$ represents the indicator function.

Furthermore, the inclusion of the "season" feature versus its exclusion revealed notable distinctions in the ICE plots for temperature and the SHAP waterfall plots, particularly regarding the summer season. The full model successfully differentiated between different seasons and captured a downward trend for extremely hot summer days. This trend aligns more closely with real-world expectations, as biking tends to become less enjoyable in excessively hot weather. Conversely, the confounding model exhibited an ever-increasing trend across all seasons, resulting in a structural overestimation of hot summer days.

3.2 Admission

Again we will first look at the prediction-truth plot where we colored the dots based on the CGPA value. In figure 38 we see both plots and observe that the full model is better than the confounding model although the difference is not big. Looking at the values in the range of 0.7 to 0.9 we observe a bigger discrepancy in the confounding model than the full model. (figure 37). It is also obvious that a higher CGPA score leads to a better chance of admission. This is captured by both models. Remember that the confounding model has no access to the feature CGPA.

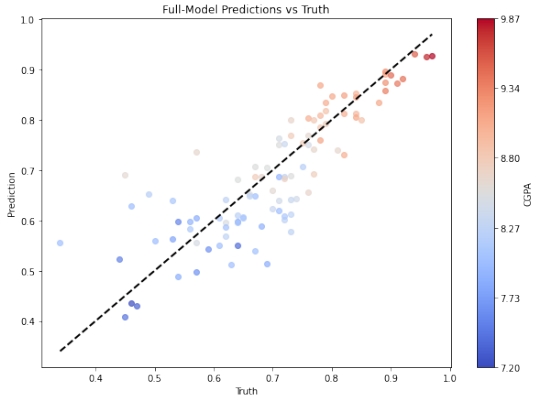

Figure 36: This figure shows the truth-prediction plot for the full model.

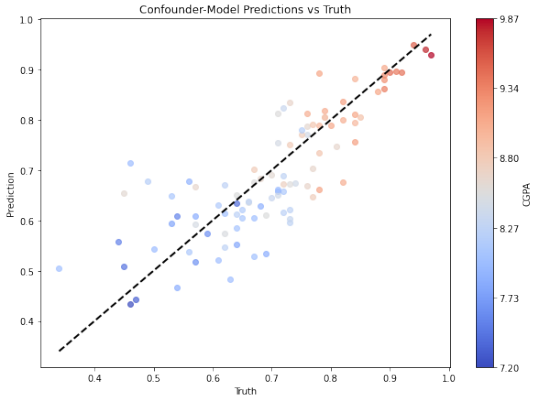

Figure 37: This figure shows the truth-prediction plot for the confounding model (Trainset does not include feature 'CGPA') .

Figure 38

3.2.1 ICE-Plots

The left side shows the ICE plots for the full model while the right side shows the plots for the confounding model.

Figures 39 and 40 present Individual Conditional Expectation (ICE) plots for the variable 'GRE Score'. In both plots, we observe a positive relationship between the GRE score and the likelihood of admission, signifying that a higher GRE score generally improves the chances of admission. However, this trend is more pronounced in the confounding model, indicating that the effect of the GRE score on admission probability may be more significant when other factors are held constant.

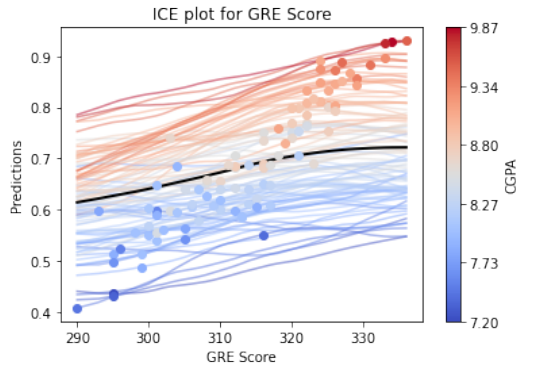

Figure 39: ICE plot for the feature GRE score of the full model.

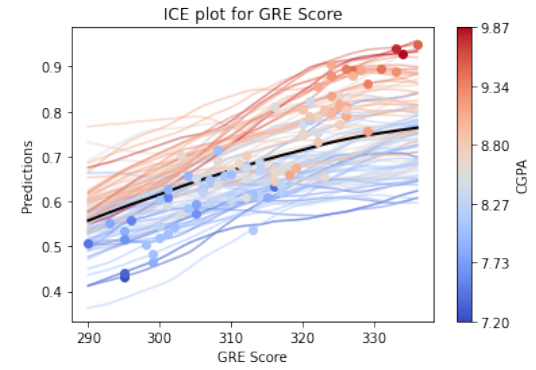

Figure 40: ICE plot for the feature GRE score of the confounding model.

Figure 41

The ICE plots for the TOEFL Score 44 are similar but for students with lower CGPA values the

confounding model indicates that a higher TOEFL score is more important than for the full model. This can be seen from the slopes of the blue lines. (Figure 43)

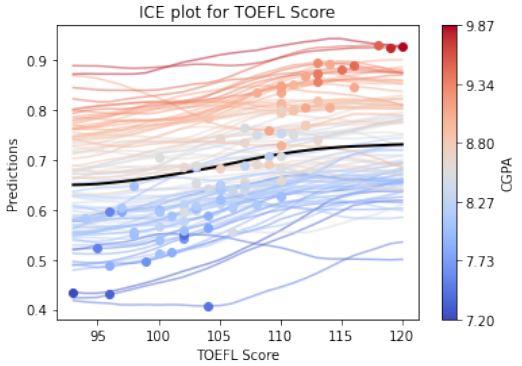

Figure 42: ICE plot for the feature TOEFL Score of the full model.

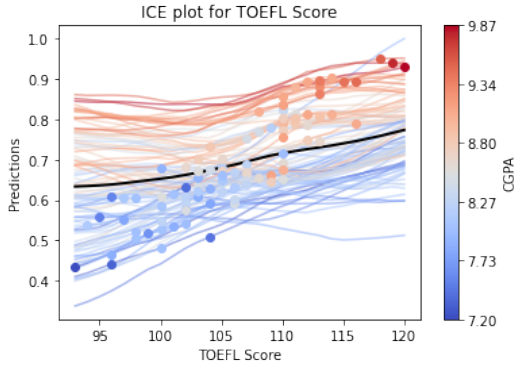

Figure 43: ICE plot for the feature TOEFL Score of the confounding model.

Figure 44

The plots for university ranking are almost constant for the full model, while the confounder model indicates a positive trend for high CGPA scores and a negative trend for low CGPA scores as seen in figure 46. From the ICE plot, we also see that a high CGPA score only exists in our training data with a university rating of 4 to 5. This can be seen by the dark red dots which are only at high ratings. At the same time, a low CGPA score comes mostly with a bad university rating (a lot of blue dots in lower rankings). This can be interpreted in the following way: High CGPA students apply more often for higher-ranking universities and lower CGPA students to lower-ranking universities. The ICE plot 46 suggests that the chance of a low CGPA student shrinks if the student applies for a higher-ranking university that aligns with reality. On the other hand, the chance also shrinks for a high CGPA student if the student applies for a lower-ranking university that does not align with reality. This can happen because as mentioned this case is out of the distribution of our training data so the network can do here whatever it wants without getting punished, thus it is important to not rely on this interpretation.

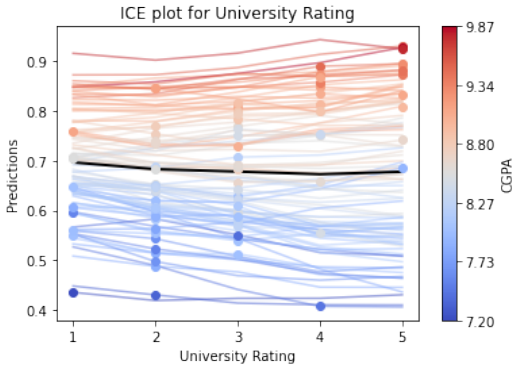

Figure 45: ICE plot for the feature University ranking of the full model.

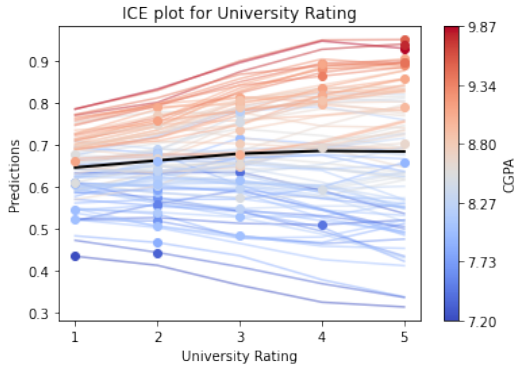

Figure 46: ICE plot for the feature University ranking of the confounding model.

Figure 47

For the variable LOR, we have again plots with a positive trend which is slightly stronger in the confounder model. The full model puts slightly more importance on the LOR if the CGPA score is low than when it is high.

The ICE plots for the "statement of purpose" (SOP) feature 53 reveal intriguing differences between the two models. In the main model, there is a slight overall increase in the chance of admission as the SOP improves. This trend is consistent for almost every individual line in the plot. However, the

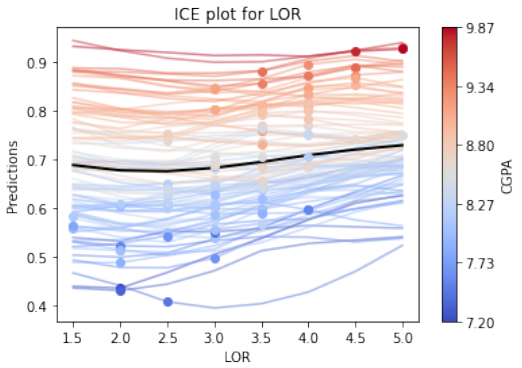

Figure 48: ICE plot for the feature letter of recommendation of the confounding model.

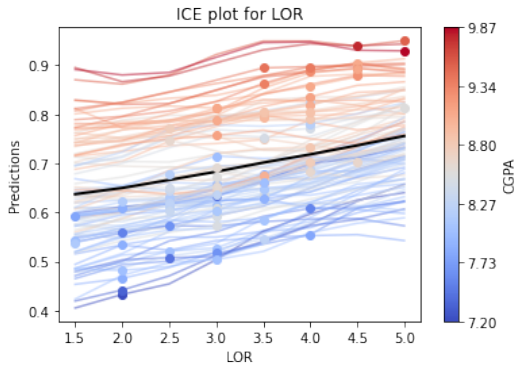

Figure 49: ICE plot for the feature letter of recommendation of the confounding model.

Figure 50

confounding model displays a distinct pattern: for high CGPA students, there is a sharp decline in the chance of admission when the SOP is not good. Conversely, low CGPA students do not experience an increase in the chance of admission when the SOP improves. These contrasting observations could be attributed to the fact that the evaluated data points are out of distribution. In other words, the results deviate from what one would expect, namely that a better SOP would certainly lead to a higher chance of admission.

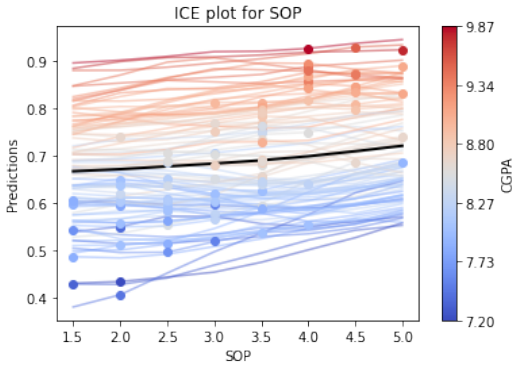

Figure 51: ICE plot for the feature statement of purpose of the full model.

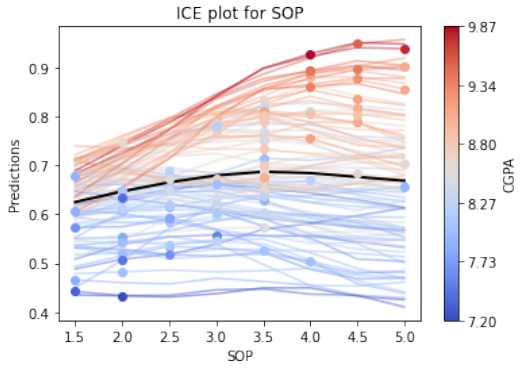

Figure 52: ICE plot for the feature statement of purpose of the confounding model.

Figure 53

The last ICE plot was generated by perturbing the feature for indicating research experience. Since this feature is binary we only evaluate the model with either 0 or 1 in that feature. For both plots 56 we see an increase meaning the chance of admission improves if a student has research experience.

3.2.2 Feature-importance

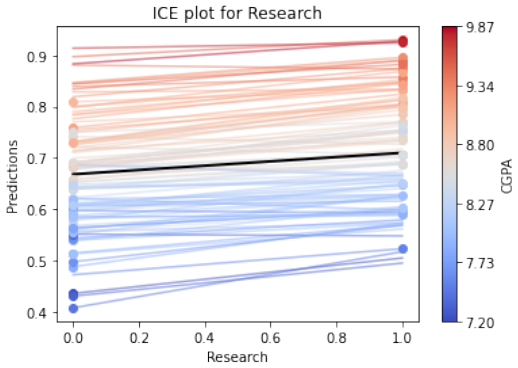

Figure 54: ICE plot for the feature research of the full model.

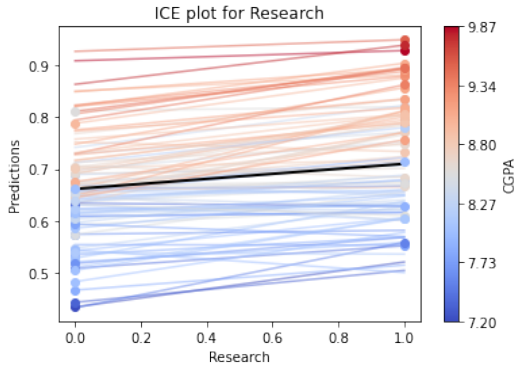

Figure 55: ICE plot for the feature research of the confounding model.

Figure 56

SHAP Waterfall Plot for the student with the highest chance of admission (full model)

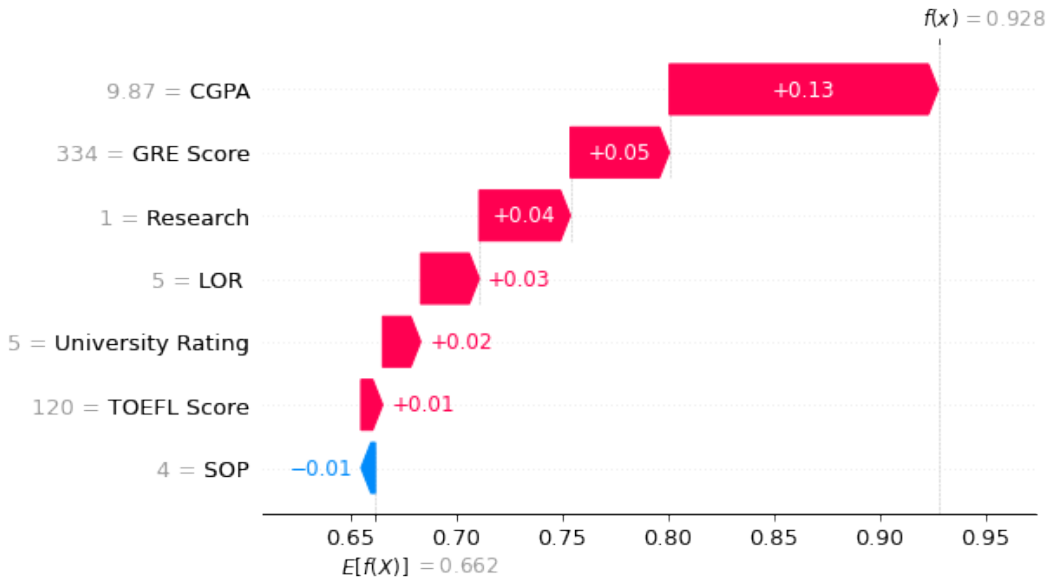

Figure 57

SHAP Waterfall Plot for the student with the lowest chance of admission (full model)

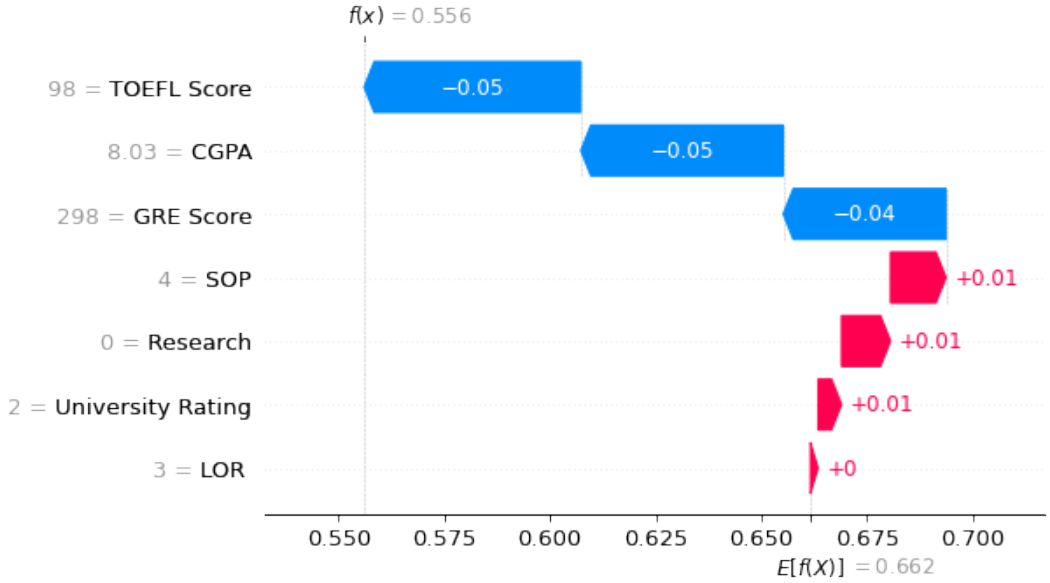

Figure 58

SHAP Waterfall Plot for the student with the highest chance of admission (confounding model)

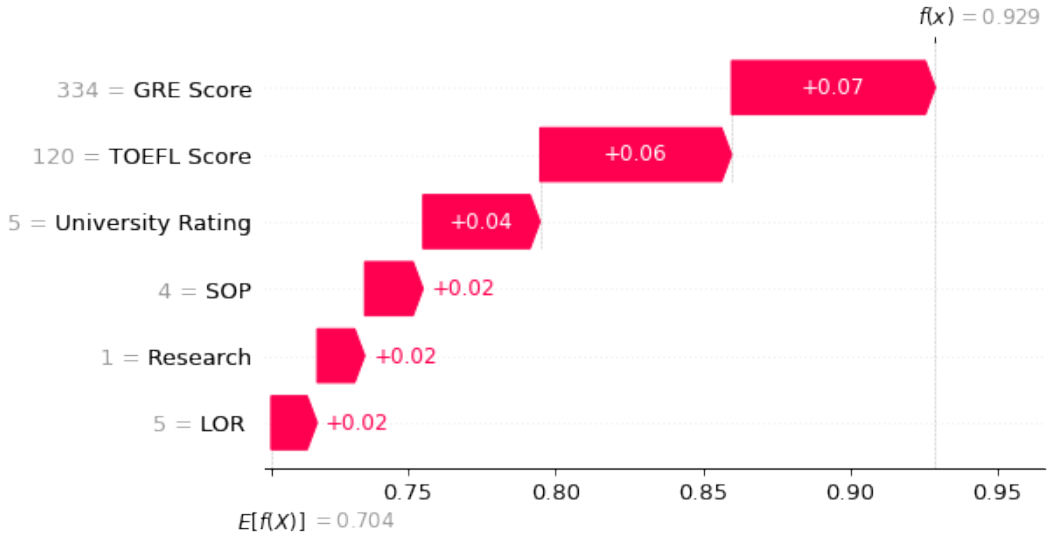

Figure 59

SHAP Waterfall Plot for the student with the lowest chance of admission (confounding model)

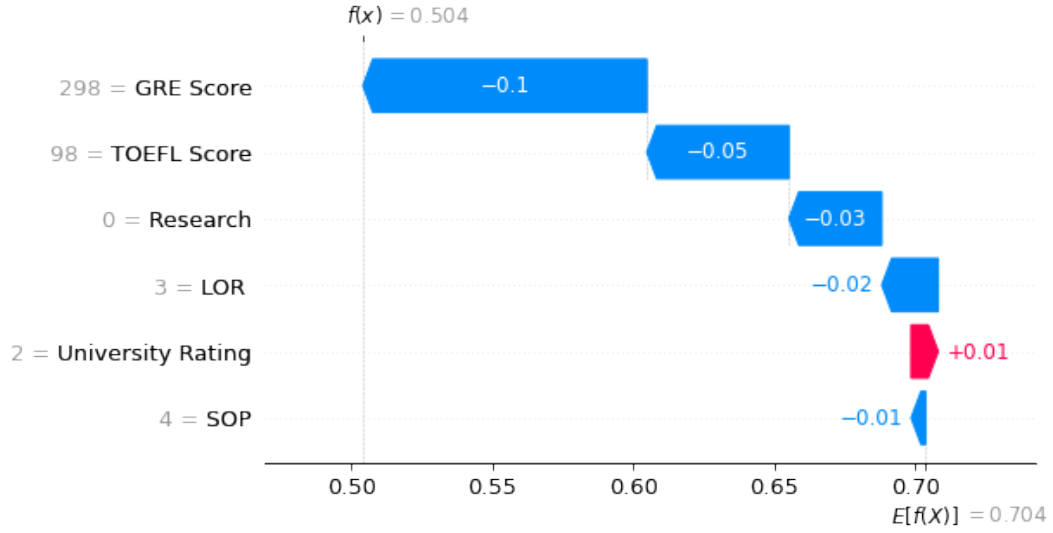

Figure 60

SHAP feature importance (full model)

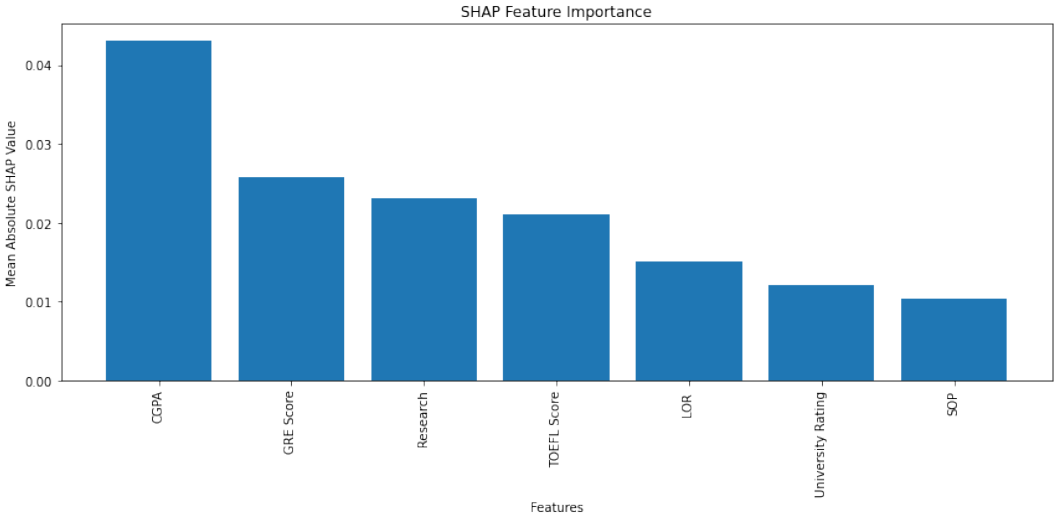

Figure 61

SHAP feature importance (confounding model)

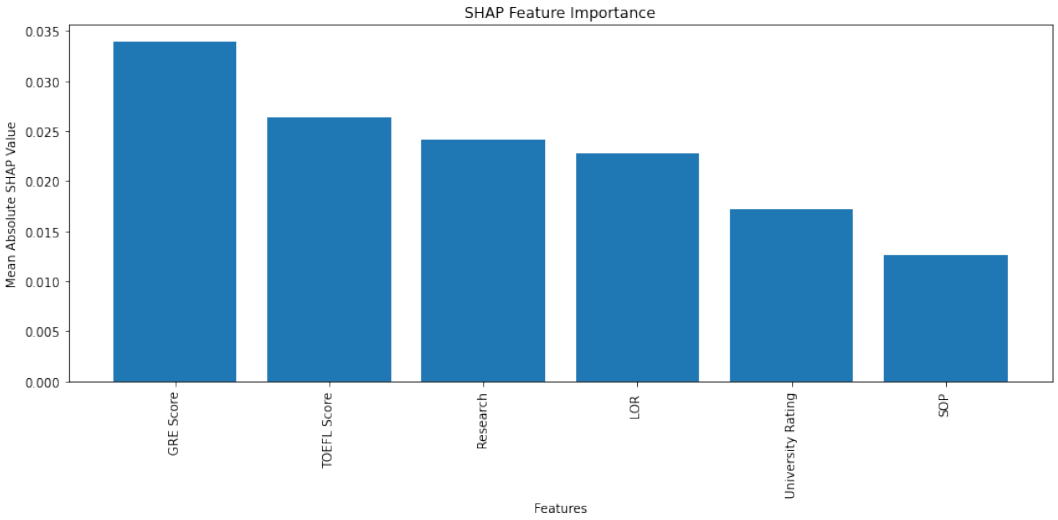

Figure 62

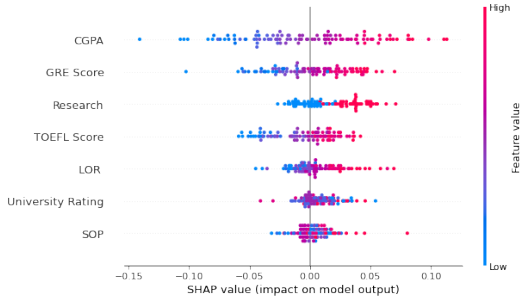

Figure 63

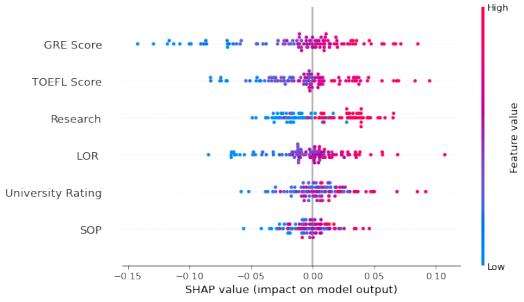

Figure 64

Figure 65

Conclusion

3.3 Toy-Data

This time we use variable $x1$ as a confounder, hence the prediction-truth plot is color coded accordingly. It's important to note that for the creation of these visualizations, we have utilized a subset of 300 randomly selected samples from the unseen test set. This strategy ensures clarity and avoids the over-saturation of data points that could potentially obscure the visualization, thus providing a more interpretable and meaningful graphical representation. In figure 68 we see both plots and observe that especially for very high and very low values of $x1$ the prediction of the confounding model lacks accuracy.

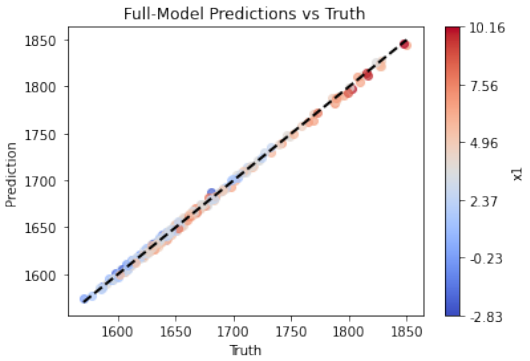

Figure 66

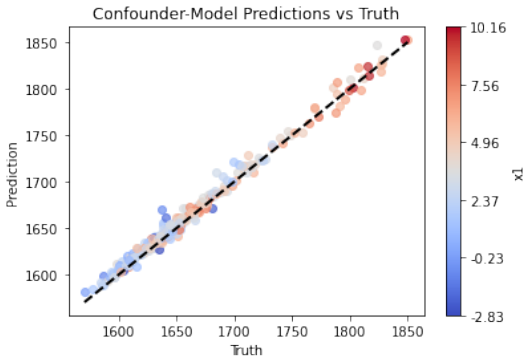

Figure 67

Figure 68

3.3.1 ICE-Plots

For our toy dataset, we are in a fortunate position where we know the underlying truth. This allows us to create ICE plots that directly relate to this truth. By comparing these plots to those generated by our models, we can better understand how our models are performing and where they may be making mistakes.

We start by looking at the ICE plots for feature $x0$. Since this is just a Gaussian random variable with mean 1 and variance 1 and simply added to y, we expect to see just straight lines with a small positive slope. This is the case for the ground truth model as seen in figure 71. In the case of the full model, we notice that the ICE plots (69) exhibit primarily linear trends around the original values. However, where the model is evaluated at points outside the training distribution, we observe lines that either rise or fall. This serves as a clear reminder that when our model is extrapolating beyond the distribution of the training set, its behavior can become unpredictable and arbitrary. The confounding model, as seen in figure 70, shows an upward trend which is also captured by the average line. This is probably related to the fact that $x0$ is adjusting for $x1$ not available since this pair has the highest correlation.

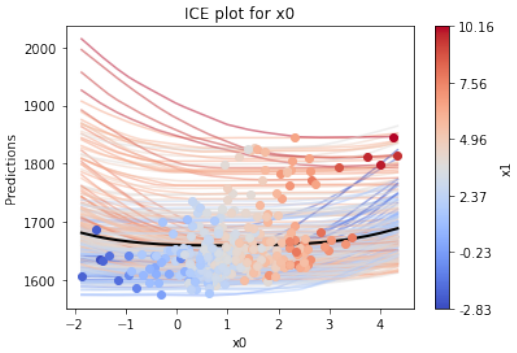

Figure 69: This is the ICE plot for variable $x0$ of the full model.

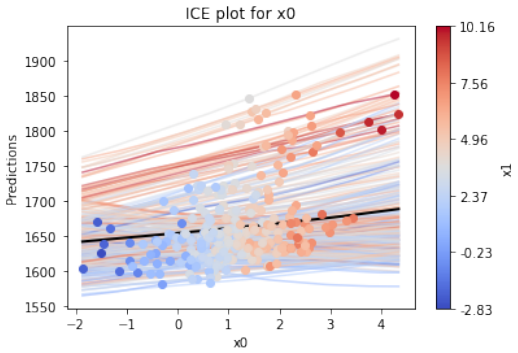

Figure 70: This is the ICE plot for variable $x0$ of the confounding model where $x1$ is left out.

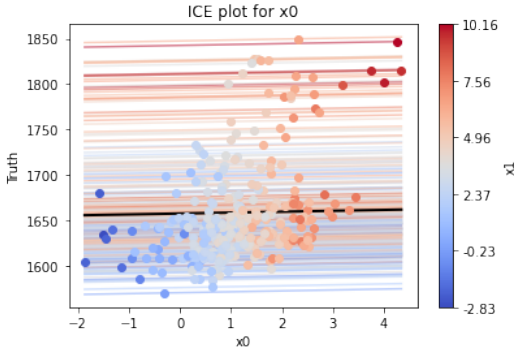

Figure 71: This is the ICE plot of the ground truth function.

Figure 72

The next variable under consideration is $x2$, which interacts with $x1$ to influence y. This relationship is represented in the equation as a product of $x1$ and $x2$, magnified by a factor of 10, that is, $10X_1X_2$. The full model (73) is nerly identical to the truth ICE plot in figure 75. The confounding model also presents some variations; however, it is consistent with the primary trend. Specifically, we observe that as the values of $x1$ increase, the model predictions correspondingly show a rising trend. Conversely, for lower values of $x1$, the model predictions either remain constant or exhibit a declining trend.

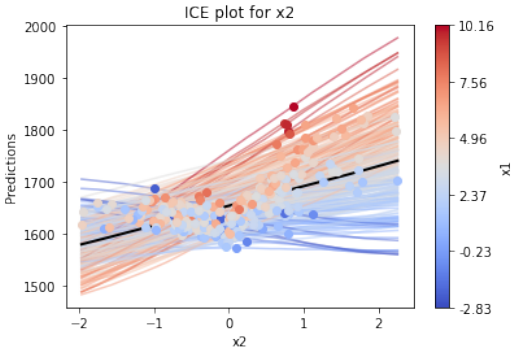

Figure 73: This is the ICE plot for variable $x2$ of the full model.

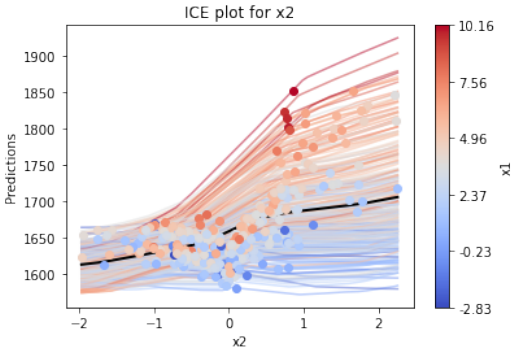

Figure 74: This is the ICE plot for variable $x2$ of the confounding model.

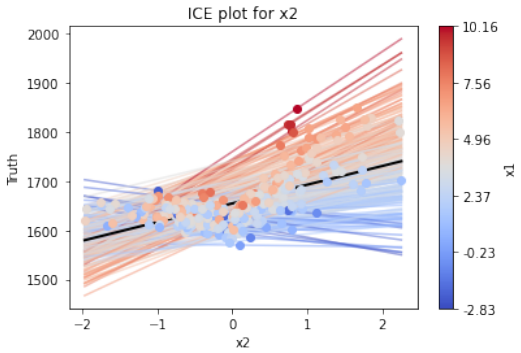

Figure 75: This is the ICE plot of the ground truth function for variable $x2$.

Figure 76

Now we look at $X3$ which is $\sqrt{|X_1 X_2|}$ plus some noise. In y the variable gets scaled and shifted and then squared. The ICE plots look all similar (77,78), but the plot of the confounding model has a more cone-like shape for bigger values of $x3$. The smaller $x3$ is the narrower the bundle of lines gets. This may be an artifact of not having $x1$ available.

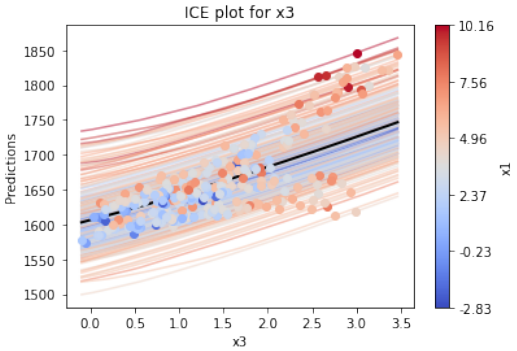

Figure 77: This is the ICE plot for variable $x3$ of the full model.

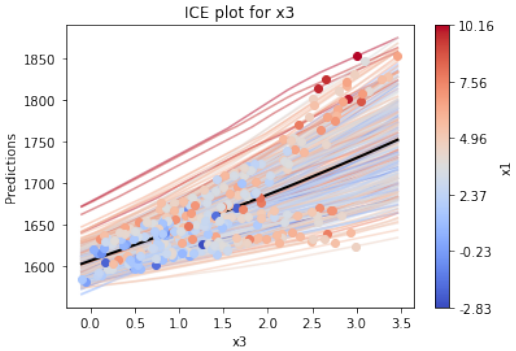

Figure 78: This is the ICE plot for variable $x3$ of the confounding model.

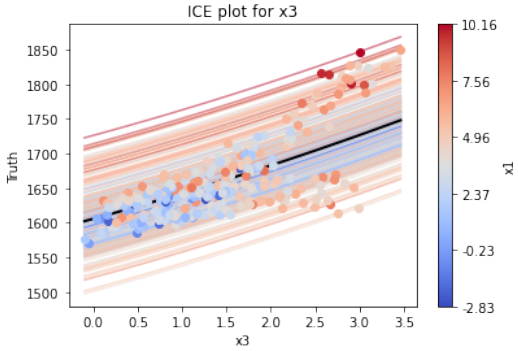

Figure 79: This is the ICE plot for variable $x2$ of the ground truth model.

Since $x4$ is inside the *sin* in the equation for y it has not much impact on the magnitude of y. The ground truth ICE plot 98 reveals the shallow periodicity, but both models approximated this with just straight lines.

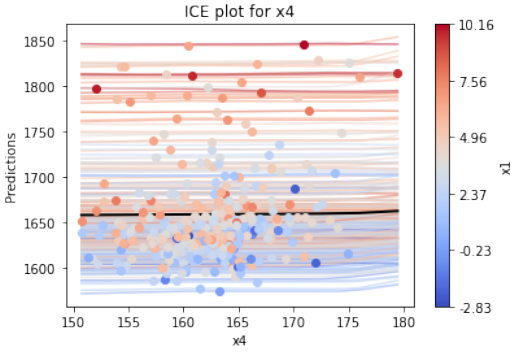

Figure 80: This is the ICE plot for variable $x4$ of the full model.

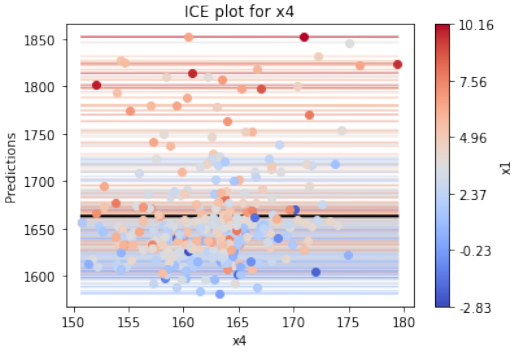

Figure 81: This is the ICE plot for variable $x4$ of the confounding model.

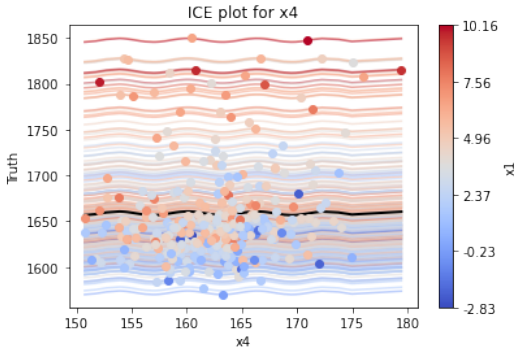

Figure 82: This is the ICE plot for variable $x3$ of the true model.

The last variable in this data set is $x5$ which should resemble a binary variable. With probability 0.5, we either add 30 or not. This can be clearly seen by all ICE plots. If the variable is 0 we get a lower prediction as if $x5$ is 1. Note that the lone between the values of 0 and 1 are interpolated by the graph and are not function evaluations.

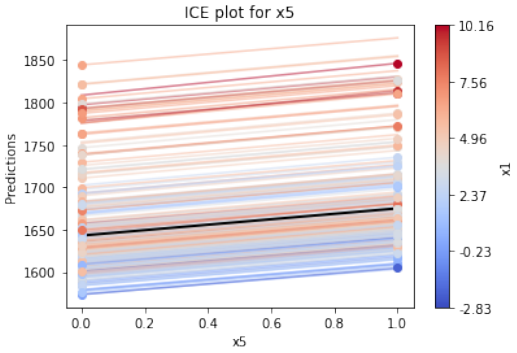

Figure 83: This is the ICE plot for variable $x5$ of the full model.

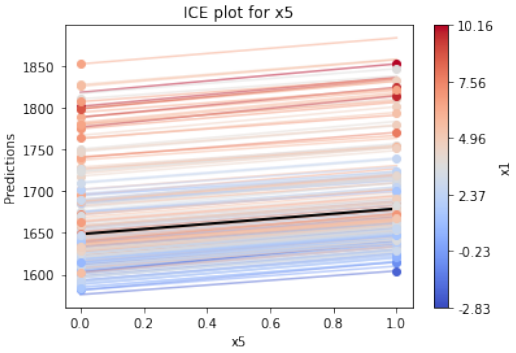

Figure 84: This is the ICE plot for variable $x5$ of the confounding model.

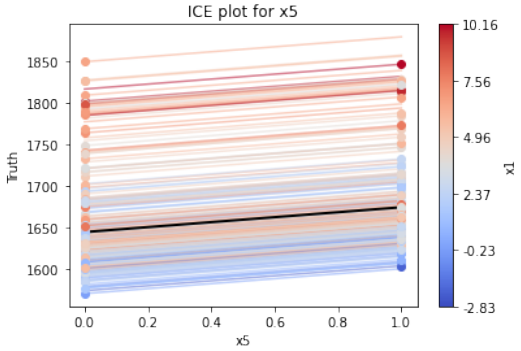

Figure 85: This is the ICE plot for variable $x5$ of the true model.

3.3.2 Feature-importance/SHAP values

In this analysis, we'll again focus on the instances leading to the highest and lowest values of y. As before, we'll use the mean as our baseline, but with one important adjustment for $x5$. Since $x5$ is a binary variable taking only values 0 or 1, the neural network can behave in an arbitrary manner for intermediary values without incurring any penalty. Averaging $x5$ yields a value of 0.5. However, evaluating the mean instance results in a prediction even lower than the lowest instance in the test set.

This situation presents a problem when calculating the SHAP values against this baseline as it attributes an undeservedly significant influence to $x5$. If we plot a single ICE line for the baseline while altering $x5$, we notice that the model has learned a parabolic trajectory, instead of a straightforward linear one. This observation is depicted in Figure 86.

While this doesn't qualify as an error per se, since the model precisely predicts the relevant values at 0 and 1, it does provide an inaccurate impression of the baseline. Consequently, to solve this problem, we have adjusted the baseline value for $x5$ to 1. It is important to note that setting it to 0 would have been equally acceptable.

The baseline has the following values for each feature :

Predictions from our models of the baseline indicate 1690.68 for the full model and 2766.26 for the confounding model. Importantly, please note that the input for the confounding model does not incorporate season variables.

Full Model

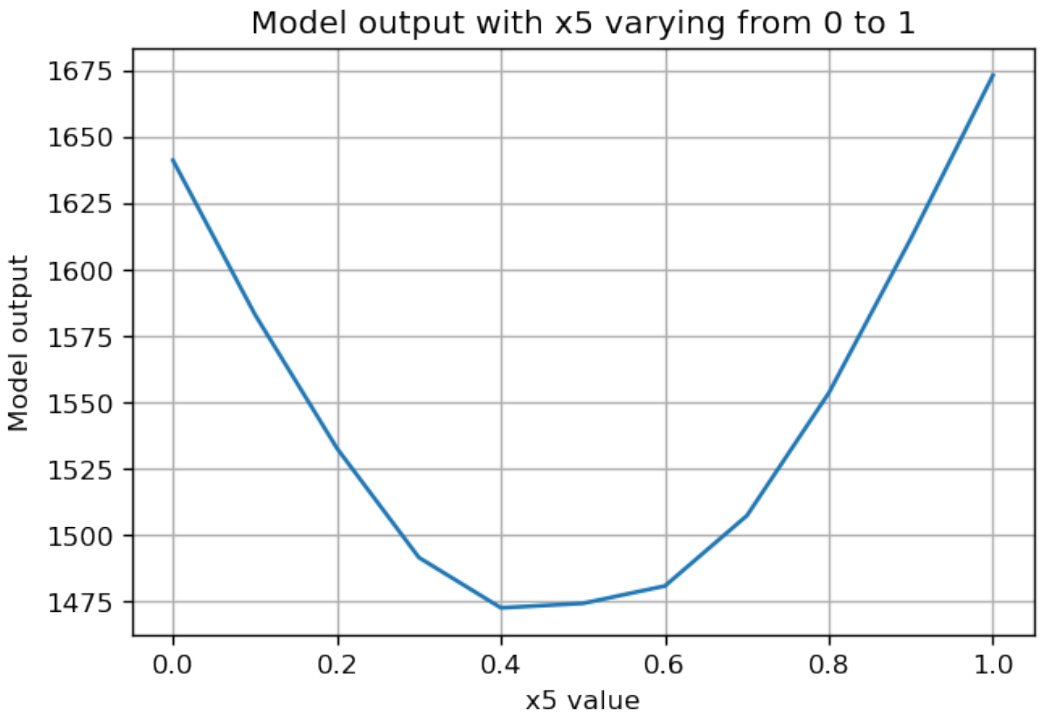

Figure 86

Feature	Value
$x0$	1.1127847
$x1$	3.7967844
$x2$	0.067418396
$x3$	1.3843178
$x4$	162.34824
$x5$	1

Table 16: Baseline summary

From the test data set, the maximum number of bikes rented in a single day reached 8555, while the minimum dropped to 506.

To better visualize the impact of various features on these extreme days, we turn to SHAP analysis. Figure 90 presents the SHAP waterfall plot for the day with the highest number of bike rentals. This graph allows us to see how each feature contributes towards pushing the model's prediction away from the baseline.

References

SHAP Waterfall Plot for the Day with Maximum Bike Rentals (full model)

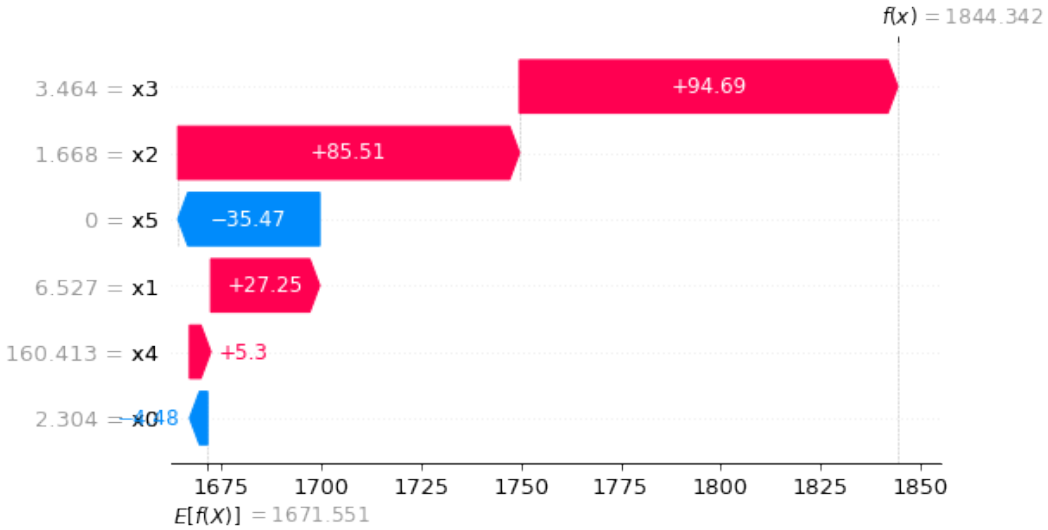

Figure 87

SHAP Waterfall Plot for the Day with Maximum Bike Rentals (confounder model)

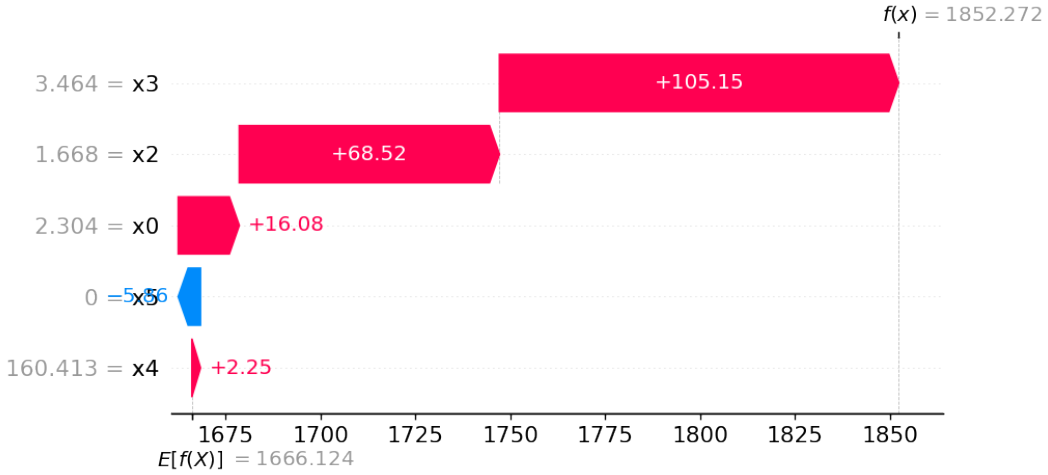

Figure 88

SHAP Waterfall Plot for the Day with Maximum Bike Rentals (true model)

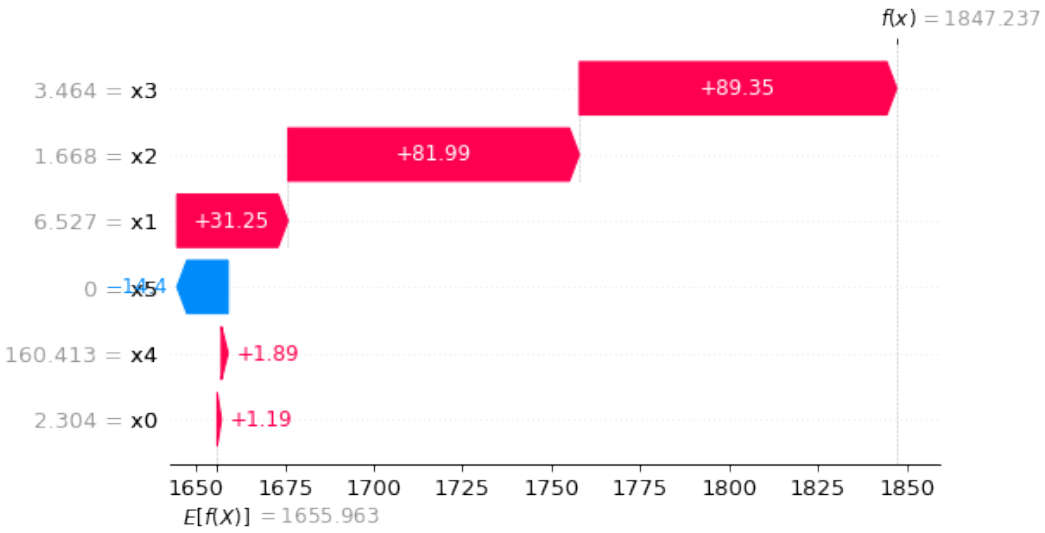

Figure 89

SHAP Waterfall Plot for the Day with Maximum Bike Rentals (full model)

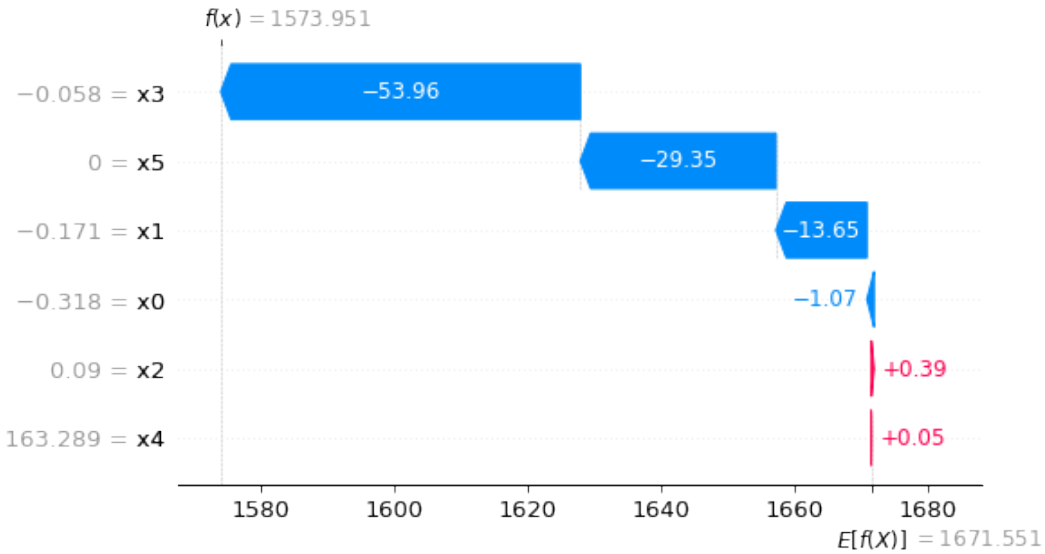

Figure 90

SHAP Waterfall Plot for the Day with Maximum Bike Rentals (confounder model)

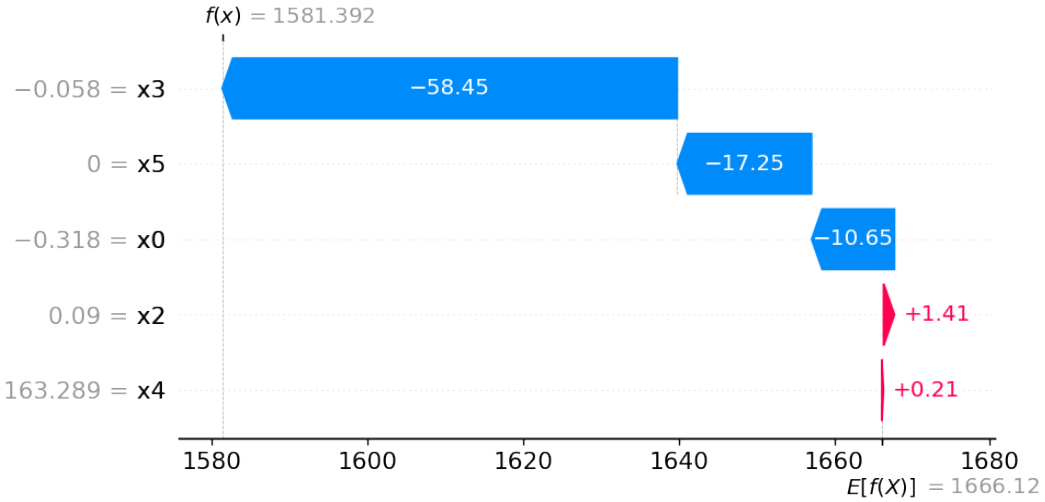

Figure 91

SHAP Waterfall Plot for the Day with Maximum Bike Rentals (true model)

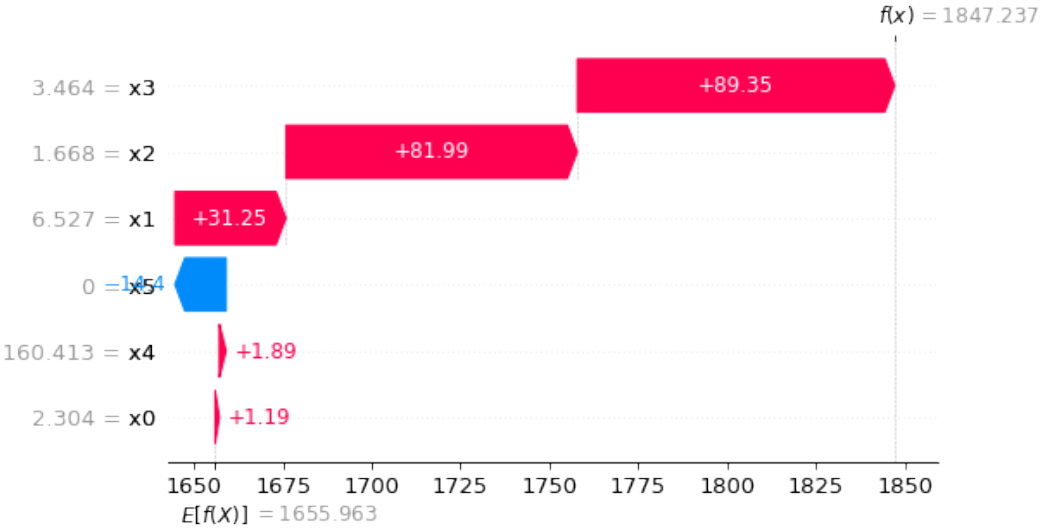

Figure 92

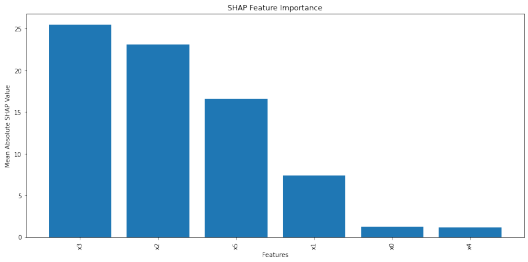

Figure 93

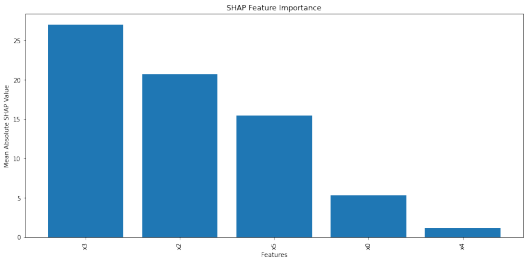

Figure 94

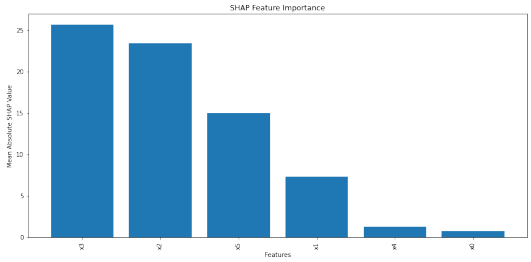

Figure 95

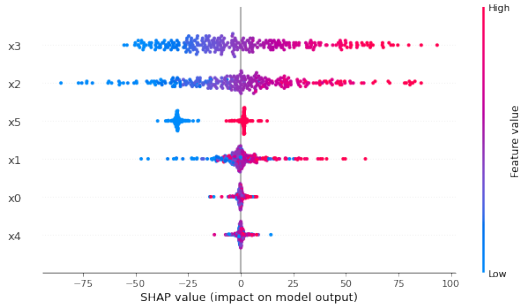

Figure 96

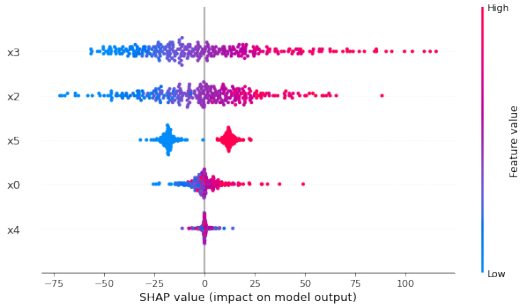

Figure 97

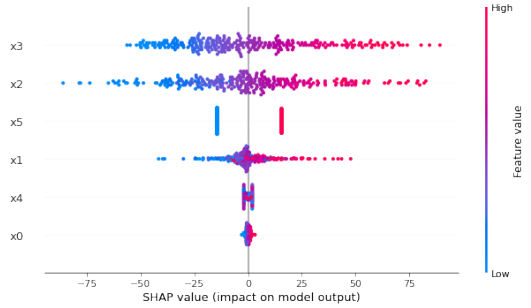

Figure 98