Report: Partial Dependence Plot (PDP) Analysis

Teresa Codoñer Esparza Pilar Mas Sarrión
Vicente Martínez Sánchez

May 16, 2025

1 Introduction

The objective of this exercise is to analyze the influence of various features on the predicted outcomes using Partial Dependence Plots (PDPs). PDPs help visualize the marginal effect of one or two features on the predicted outcome of a machine learning model. We used datasets related to bike rentals and house prices to understand these relationships. The models were trained using Random Forest regressors, and the analysis was carried out using Python. **Note on Terminology:** A *Partial Dependence Plot (PDP)* shows the average predicted outcome as a function of one or more input variables, while holding other variables constant. It helps interpret how a model uses certain features. A *Random Forest* is an ensemble learning method that builds multiple decision trees and averages their outputs for robust prediction and reduced overfitting.

2 Methodology

2.1 Tools and Technologies

- **Programming Language**: Python
- **Libraries**: `pandas`, `scikit-learn`, `matplotlib`, `seaborn`
- **Version Control**: Git and GitHub

2.2 Process

1. **Data Loading and Preparation**: Loaded the datasets and selected relevant features for analysis.
2. **Sampling**: To reduce computational load, random samples of 1000 rows were extracted from the full datasets for Exercises 2 and 3.
3. **Model Training**: Trained a Random Forest model to predict bike rentals and house prices using the selected features.

4. **PDP Generation**: Generated both unidimensional and bidimensional Partial Dependence Plots to visualize the relationships learned by the model.

5. **Version Control**: Git was used for tracking changes and GitHub for storing and sharing the code and report.

6. **Data Sampling:** For Exercises 2 and 3, we randomly sampled 1000 rows from the datasets to reduce computation time. A fixed seed was used to ensure reproducibility.

7. **Feature Selection:** Features were selected based on domain knowledge and relevance to the target variable. For bike rentals, environmental and time-based features were chosen. For house prices, standard real estate predictors were included.

2.3 Version Control and Collaboration

Git was used for version control. Each team member committed regularly with descriptive messages (e.g., `Added 2D PDP with KDE`). The project was hosted on GitHub, allowing for collaboration and automatic backup. The repository contains the notebook, datasets, plots, and the final report. Link: `https://github.com/TPV-EDM/EDM-XAI3.git`

3 Results

3.1 Exercise 1: Influence of Days Since 2011, Temperature, Humidity, and Wind Speed on Predicted Bike Counts

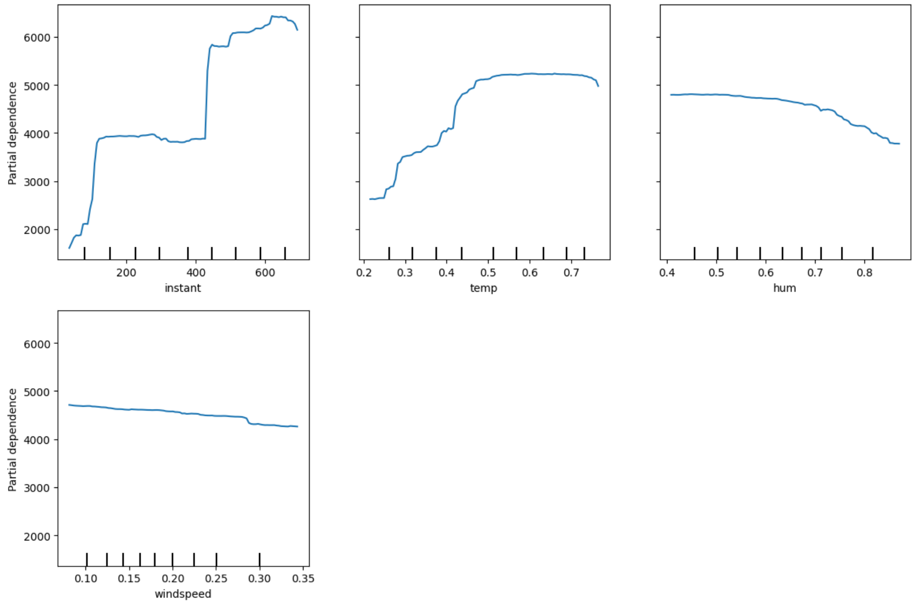

Figure 1: Unidimensional Partial Dependence Plots for Bike Rentals

Analysis:

- **Days Since 2011 (`instant`)**: There is a clear increasing trend in predicted bike rentals over time. This may reflect a rise in service adoption, growing popularity of biking as a transportation mode, or long-term seasonal trends.
- **Temperature (`temp`)**: There is a strong positive relationship between temperature and bike rentals. As the temperature increases, the predicted number of bike rentals rises significantly up to a certain plateau, suggesting that users prefer biking in warmer weather.
- **Humidity (`hum`)**: Higher humidity levels tend to slightly reduce the predicted number of bike rentals. This inverse relationship indicates discomfort or reduced willingness to use bikes during humid conditions.

- **Wind Speed (`windspeed`)**: As wind speed increases, the number of predicted bike rentals slightly decreases. Windy conditions might discourage people from biking due to physical effort or discomfort.

  Answer: Among the four variables, `temp` (temperature) had the strongest positive effect on bike rental predictions. `hum` (humidity) and `windspeed` had weaker, negative effects. The variable `instant` (days since 2011) showed a consistent increase over time, reflecting a long-term rise in rentals.

3.2 Exercise 2: 2D Partial Dependence Plot for Temperature and Humidity

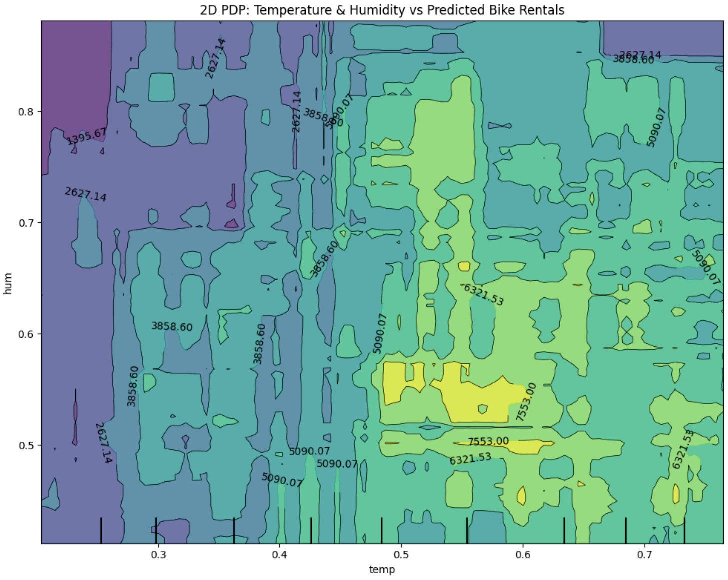

Figure 2: 2D Partial Dependence Plot for Temperature and Humidity

Interpretation:

- The 2D PDP visualizes the joint influence of `temp` and `hum` on the predicted number of bike rentals. A contour heatmap shows how the model responds to combinations of these variables.
- We observe that higher predicted rental counts are associated with moderate to high temperatures (around 0.6) and moderate humidity (0.4 to 0.6). In contrast, high humidity levels tend to reduce the model's prediction, even if the temperature is favorable.
- Additionally, we overlaid a kernel density estimation (KDE) plot which shows the concentration of data points in the feature space. Most of the

data is concentrated in regions with mid-range temperature and humidity, suggesting the model's learning is strongest in those regions.

- This visualization helps to understand interactions that wouldn't be captured by analyzing one feature at a time.

  Answer: The 2D PDP clearly shows that the highest predicted rentals occur under warm and moderately humid conditions. High humidity reduces predicted rentals even if temperatures are favorable. This confirms that temperature and humidity interact: good biking conditions require both warmth and comfort (i.e., not too humid).

3.3 Exercise 3: Influence of Bedrooms, Bathrooms, Living Area, and Floors on Predicted House Prices

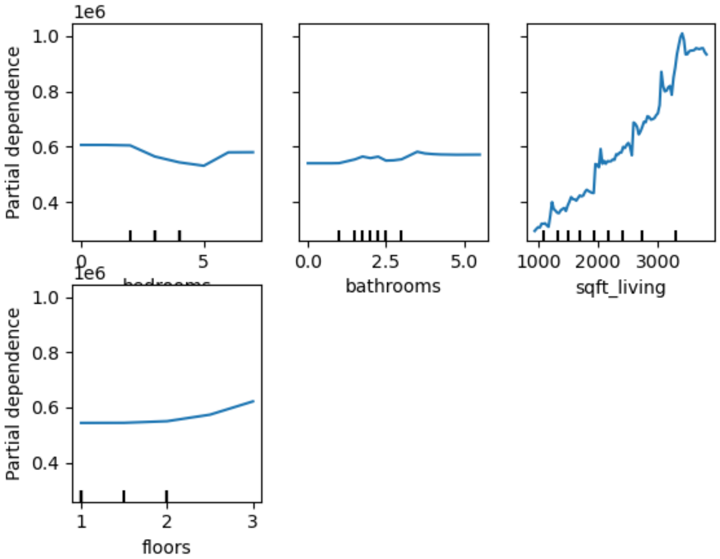

Figure 3: Unidimensional Partial Dependence Plots for House Price Prediction

Analysis:

- **Bedrooms**: The number of bedrooms has a minimal effect on predicted price. Prices remain relatively stable across values, with a slight dip for houses with many bedrooms, possibly reflecting less desirable, outdated properties with many rooms.
- **Bathrooms**: Bathrooms show a small positive effect on predicted prices, especially between 1 and 3. Beyond that, the increase levels off.

- **Living Area (`sqft_living`)**: This feature has the most noticeable impact on price. Larger living areas correspond to significantly higher predicted prices, showing a steep positive trend, as expected in real estate valuation.

- **Floors**: Houses with more floors show a slight increase in predicted price. However, the effect is less pronounced compared to living area, indicating that floor count alone is not a major price driver.

  Answer: The most influential variable for house prices is `sqft_living` (living area). `Bathrooms` and `floors` have moderate effects, while `bedrooms` show minimal and inconsistent impact. These observations align with real estate valuation principles.

4 Conclusions

4.1 Key Findings

- **Bike Rentals**: Temperature was the most influential feature, showing a strong positive relationship with rental count. Humidity and wind speed negatively influenced the outcome, and the use of PDPs allowed us to quantify these effects clearly. The 2D PDP for temperature and humidity revealed complex interactions and reinforced these findings.

- **House Prices**: The size of the living area was by far the most influential variable in predicting house prices, followed by the number of bathrooms. Bedrooms and floors had smaller impacts. This matches domain expectations and validates the model's ability to learn realistic relationships.

4.2 Limitations

- The models were trained on random subsets of the datasets to reduce computation time. This might slightly affect the generalizability of the insights.

- We used a single type of model (Random Forest Regressor). Exploring additional models could improve robustness.

- PDPs assume feature independence, which may not always hold in real-world data.

4.3 Future Improvements

- Evaluate other model-agnostic explanation methods, such as SHAP or ALE, to compare with PDPs.

- Use grid search or hyperparameter tuning to optimize model performance.

- Apply PDP analysis to more features and larger datasets for deeper insights.

5 Appendices

5.1 Code

The code used for generating the results can be found in the Jupyter Notebook linked in the GitHub repository: `https://github.com/TPV-EDM/EDM-XAI3.git`

5.2 References

- Scikit-learn documentation: `https://scikit-learn.org`
- Pandas documentation: `https://pandas.pydata.org`
- Matplotlib documentation: `https://matplotlib.org`
- Seaborn documentation: `https://seaborn.pydata.org`