Predicting Demand for Bike-Sharing in Washington D.C.

Fadeel Sher Khan [*1] **Lakshya Jagadish** [*2] **Asvin Kumar** [*2] **Powell Lowe** [*2]

Abstract

Faced with a diverse range of objectives such as optimal bike placement, financial constraints and other external factors such as topology and climate, bike-sharing companies are more interested in studying user demand and coming up with trustworthy predictions to up their revenues. With more bike companies coming to the forefront, the companies need to maintain relevance in the market by maintaining a competitive edge. With the rise of cost-effective, policy-oriented transportation initiatives, such as the promotion of walking and bicycling, the market is booming. In such a situation, we study the bike-sharing patterns of registered and casual users and predict their usage patterns using ML models, while comparing them using certain quantitative measures and reasoning why certain models perform better than the rest. We also conduct a thorough Exploratory Data Analysis to understand the type of data we are dealing with. We show some exciting results with the various models with or without using PCA, interpret the models using SHAP for increased understandability and finally present our results with possible future directions for this work.

1. Introduction

1.1. Background

Capital Bikeshare is a bicycle-sharing system which serves Washington, D.C. and surrounding counties. It has 700+ stations and more than 6,000 bicycles (Bikeshare, a). Until 2013, it was the largest bike sharing service in the United States (Martinez, 2010). The company has seen a steady increase in ridership, as seen in Figure 1, reaching a record ridership of 428,000 in the month of May 2023 (Pascale, 2023). Riders typically buy annual memberships or pay per pass (single trip or daily) (Bikeshare, b). Bike sharing is a key aspect of the metropolitan culture of Washington D.C. In 2021, officials in D.C. announced a $19 million bike share expansion plan over six years to add 80 stations and 2,500 electric bikes (Pérez-Moreno, 2023). According to the Washington Post, the city also added 1.5 miles of protected bike lanes between Florida and Pennsylvania avenues, adding to more than 30 miles of such lanes (Pérez-Moreno, 2023). D.C. Mayor Muriel E. Bowser (D) has said the bike lane expansion is part of the city's goal of reviving downtown (Pérez-Moreno, 2023). Thus, there is a significant interest in understanding ridership trends and variations. Doing so can further help in city planning and development, in marketing campaigns on bikes, and on overall planning for revenue generation and societal prosperity. At the same time, it can allow significant research to be done into mobility, environment and health.

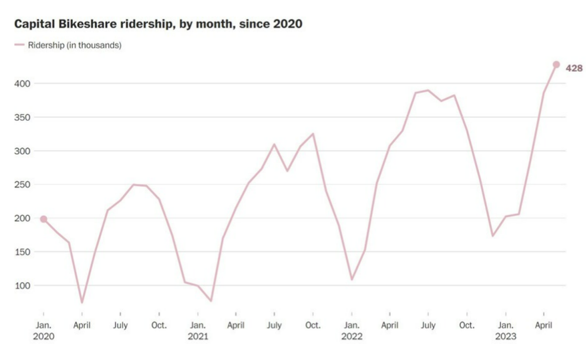


Figure 1. Trends in Capital Bikeshare Ridership, adapted from (Pérez-Moreno, 2023)

1.2. Dataset Information

Capital Bikeshare donated a segment of their rental bike sharing dataset to the University of California Irvine Machine Learning Repository in 2013 (Fanaee-T, 2013). The dataset contains information about rental bikes from 2011

[*]Equal contribution [1]Department of Biomedical Engineering, University of Texas, Austin, United States [2]Department of Electrical Engineering, University of Texas, Austin, United States. Correspondence to: Fadeel Sher Khan <fadeelkhan@utexas.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

and 2012. There are 17389 instances with 13 features. Each instance represents an hour. The feature set is shown in Table 1. For this study, the dataset was divided into "hour" and "day" where the "hour" dataset included all 17389 instances while the "day" dataset is a subset of "hour" and aggregated by the "day" feature of each instance. The "day" dataset is composed of 731 instances.

Table 1. Feature set of the data; noth "hour" and "day" have the following fields, except hour which is not available in "day"

FEATURE SET	DESCRIPTION/ENCODING
date	DATE FORMAT
year	0: 2011, 1:2012
hour	0 TO 23
holiday	BOOLEAN
working day	BOOLEAN
day of week	0 TO 6
season	DUMMY ENCODING
weather	SCALE: 1 (BEST) TO 4 (WORST)
temperature	NUMERIC IN CELSIUS
"feels like" temperature	NUMERIC IN CELSIUS
humidity	NUMERIC
windspeed	NUMERIC

There are three potential target variables in the dataset (Table 2). All annual members were considered "registered" bike users and all riders that used the bike on either the daily or single pass were considered "casual" bike users.

Table 2. Target variables identified

TARGET	DESCRIPTION
registered	RIDERS WITH ANNUAL MEMBERSHIP
casual	RIDERS WITH SINGLE OR DAILY PASS
cnt	SUM OF *registered* AND *casual*

1.3. Previous Work

This dataset has largely been explored through a Kaggle competition that was launched in 2014 and included 32,000+ entries (Kaggle). The vast majority of these entries included non-deep learning methods. Many of these entries received high accuracy scores, with R^2 values reaching approx. 1.0 and RMSE values of approx. 0. However, to our knowledge, none of these entries attempt to conduct model interpretation or attempt to explain or draw conclusions from these trends. At the same time, none of these methods attempt principal component analysis (PCA) on the given dataset.

This dataset has also been previously utilized for more deep models. Petneházi et al. utilized this dataset to create recurrent neural networks to predict registered and casual users, achieving R^2 values between 0.78-0.96 and RMSE values of 103-31.3 counts for various versions of their models (Petneházi, 2018). However, they note that the limited data is a significant barrier to the successful implementation of deep neural networks (shown by less successful RMSE and R^2 values as compared to traditional machine learning methods used in (Kaggle)).

Outside of (Kaggle) and (Petneházi, 2018), we did not find any literature that used novel methods for our desired target variables. However, Britton et al. utilized this dataset to visualize statistical interactions between features to further model interpretability, further lending credence to an approach that incorporates both predictive analysis and associated model interpretation (Britton, 2019). Additionally, Subbaswamy et al. utilizes this dataset in their paper understanding data shift but do not conduct any predictive analysis (Subbaswamy et al., 2018). This study attempts to do the following contributions to related work:

- Conduct predictive modeling of registered, casual and total bike riders
- Conduct modeling after data has been inverse transformed using PCA
- Conduct model interpretation and reasoning based on state-of-the-art approaches

2. Exploratory Data Analysis

Exploratory data analysis was performed to better understand the dataset. This included null entry counting, data set plotting (complete and subset), correlation matrices, and seasonal decomposition.

2.1. Null Values

731 and 17379 non-null values are found in each column of the daily and hourly datasets, respectively. This indicates a complete dataset without missing values.

2.2. Dataset Plotting

Both raw datasets are plotted in Figure 2. Each demonstrates a clear yearly trend with peaks during the warmer months and that majority of riders are registered. The third plot magnifies the hourly data for a two week interval and demonstrates extreme peaks during commuting hours on weekdays. Figure 3 explores these observations further and highlights the differences between average weekdays and weekends (Srinivasan). The averaged weekday curves confirm that commuting hours consistently have a ridership peak on weekdays while weekends are generally a smoother, more well rounded curving peaking in the early afternoon. In the second plot, it becomes clear that the peaks are largely caused by registered users. While a helpful insight for this project, this phenomena is a useful business conclusion on its own and highlights the successful conversion of consist commuting riders from casual to registered users.

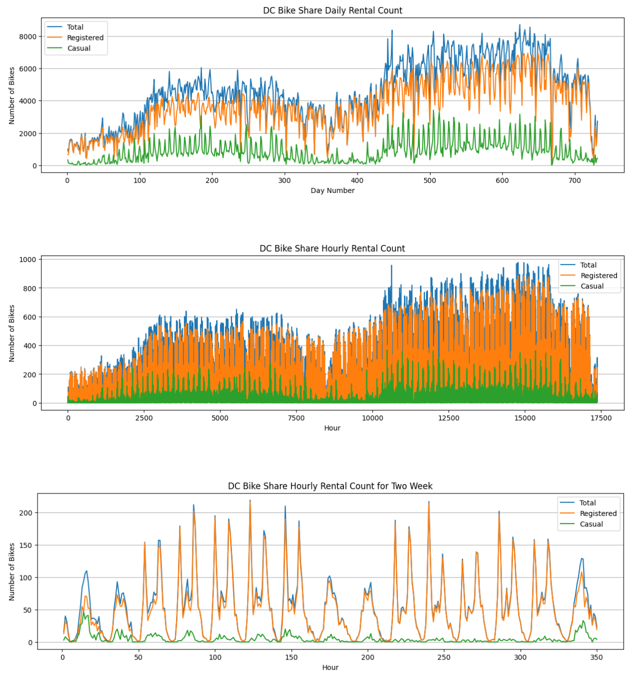


Figure 2. Raw plots of total, casual, and registered users for the daily and hourly dataset. The hourly dataset is also plotted over a two-week interval for increased seasonal trend visibility.

2.3. Seasonal Decomposition

We also made observations concerning seasonal decomposition for the given dataset. The trend data roughly indicates a yearly trend with possible shorter term trends. Seasonality is difficult to interpret at this resolution but demonstrates 104 spikes indicating weekly seasonality. The residuals have similar variance throughout the data with a slight increase in outliers in the second year. Figure 4 shows only the daily dataset decomposition, however the hourly dataset behaved in a similar manner.

2.4. Weather Feature Correlation Matrix

The correlation matrix in Figure 5 indicated that temperature was the most correlated with total count, registered, and casual users.

3. Modeling

3.1. Pre-Processing

A series of pre-processing steps were conducted to prepare the data for predictive modeling. The continuous weather features (Table 1) were normalized to the range [0, 1]. The dataset was divided up to predict 1) total bike riders, 2) casual riders, and 3) registered riders. A test-train split

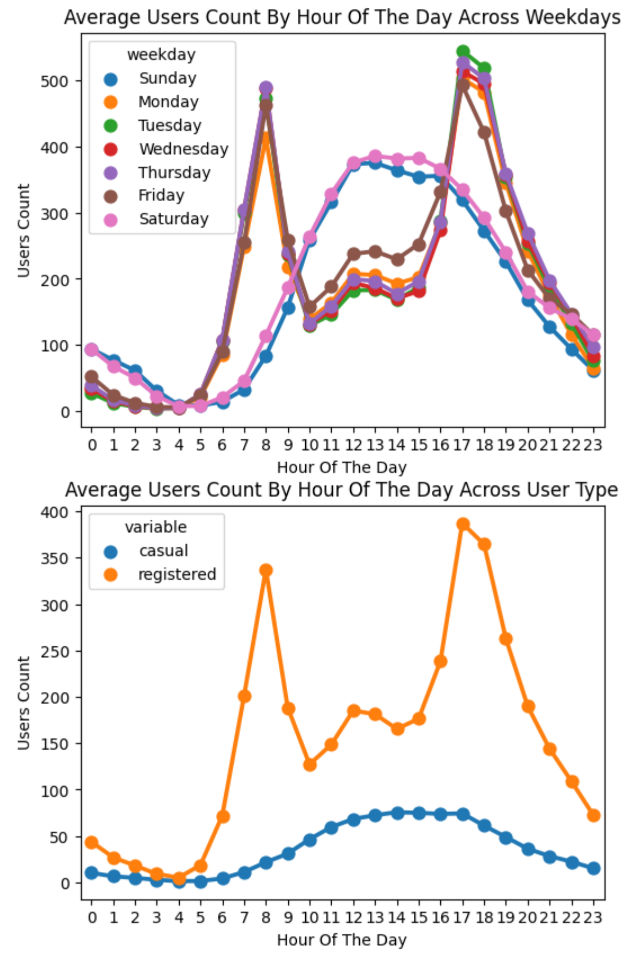


Figure 3. Average user across 24 hours distinguished by day of the week and rider type

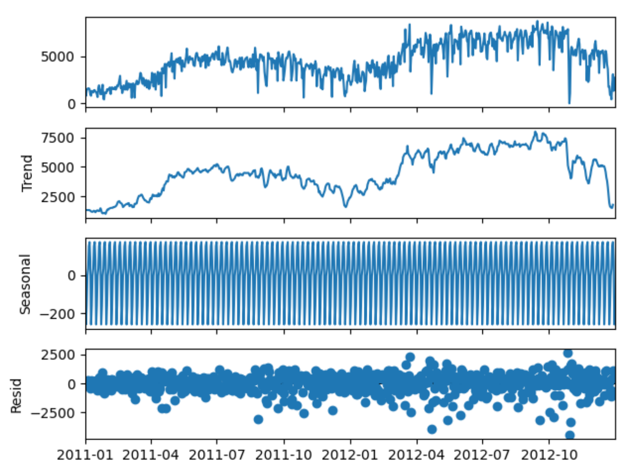


Figure 4. Seasonal decomposition of the dataset features using $statsmodels.tsa.seasonal.seasonal\_decompose()$

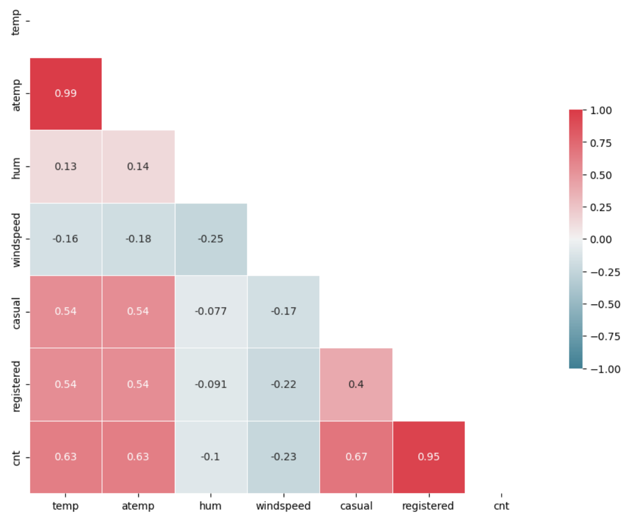


Figure 5. Temperature is the more correlated weather feature at 0.63. Humidity and windspeed show minimal correlation.

of 20-80 was conducted. The training dataset was further evaluated via a 3-fold cross validation. To conduct hyper-parameter tuning, a grid search was conducted over the maximum range of all model training parameters. Finally, principal component analysis (PCA) was done to understand the variance of the data captured by the feature set. Through the cumulative variance plot (Figure 6), it was determined that 97% of variance was captured by the top 3 components. Thus, the data was then inversely transformed based on the top 3 components and predictive modeling was performed with- and without-PCA for the top performing models.

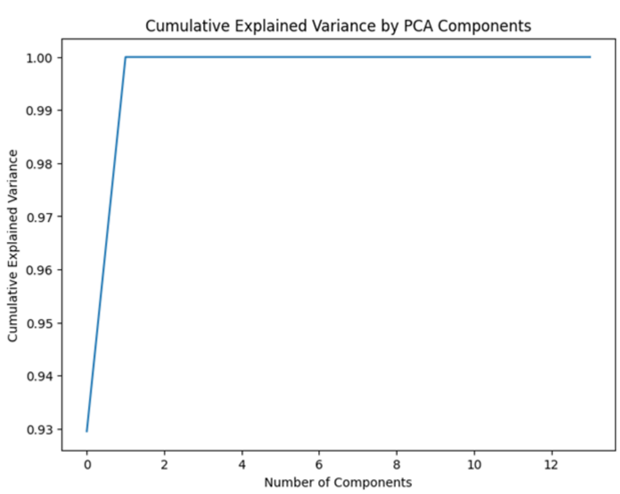


Figure 6. Cumulative variance plot for PCA

The final set of features used for the LSTM predictions are elaborated in the images shown below. One can observe the differences between the feature patterns of the dataset to which PCA has been applied vs the dataset without PCA. An 80:20 split was used for the training and testing of the LSTM model.

We trained and tested on multiple models, which have been detailed in the subsequent sections.

3.2. Linear Regression

Linear regression was used as a baseline model. Here, the target variable is estimated as a linear combination of the input features. (Weisberg, 2005) We used the input feature data as-is and the model was trained with K-fold cross validation with five folds chosen appropriately for time series data.

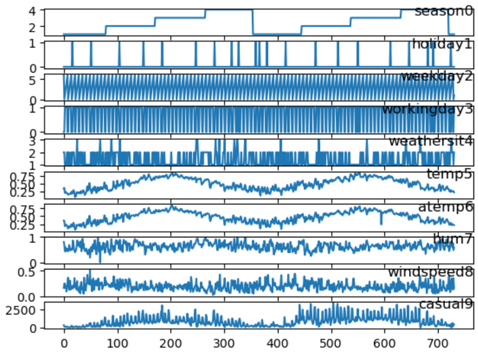
3.3. TBATS

Since the dataset is primarily a time series, a TBATS model was initially used to apply several classical time series modeling approaches to the data as an initial baseline. TBATS stands for Trigonometric seasonality, Box-Cox transformation, ARMA errors, Trend, and Seasonal components. It was created for modeling time series with multiple seasonalities such as this dataset. The TBATS method attempts to fit the data with different combinations of these time series modeling methods and choose the final model based on Akaike information criterion (AIC) (Alysha M. De Livera, 2011). However, regardless of which seasonalities were input into the model, TBATS continued to perform poorly. Because of this, TBATS was dropped from further exploration in favor of the more successful machine learning models described below.

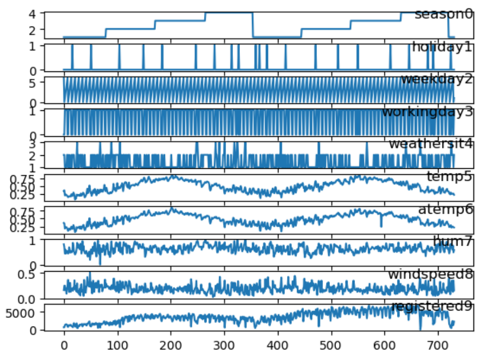
3.4. XGBoost

XGBoost is a highly efficient gradient boosting algorithm for decision trees that utilizes an ensemble of weak learners to generate a more accurate model. (Chen & Guestrin, 2016). To pick the best model, we performed a grid search with the following choices. We considered max depths of 3, 4 and 5, learning rates of 0.03, 0.1 and 0.3, minimum leaf node weights of 1, 2 and 3, data sampling ratios of 0.8 and 1.0 and finally percentage of features considered when constructing trees with options of 80% and 100%.

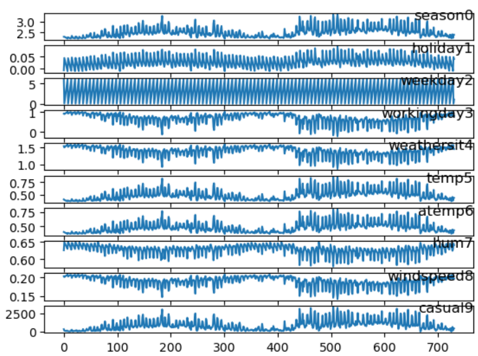
Additionally, we applied PCA to the training data and restricted it to the top 2 components and inverted the data back to its original shape. We performed a similar grid search on this restricted data and built XGBoost models for the same.



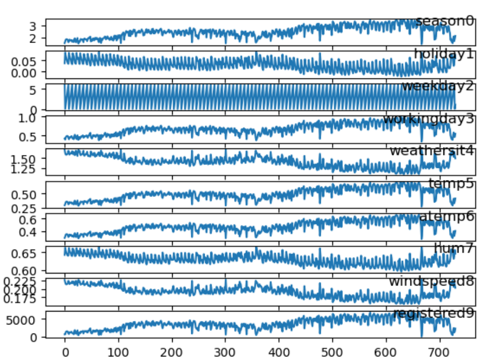
(a) Daily Casual riders features



(b) Daily Casual riders features



(c) Daily Casual riders features with PCA



(d) Daily Casual riders features with PCA

Figure 7. LTSM data features summarization with and without PCA

3.5. Random Forest Regression

Random Forest regression uses an ensemble of decision trees built with bootstrapped samples and a random subset of features for each tree. For regression, the numerical outputs of all trees are averaged to generate the ensemble's collective decision (Breiman, 2001). The ensemble was trained over a grid search including max depth (1,2,3,4), number of estimators (100,300,500), minimum number of samples per split (2,6,10), and maximum number of features as a percentage (0.2,0.4,0.6,0.8,1).

3.6. SVM

A Support Vector Regressor (SVR) tries to find a hyperplane that fits the data points in a higher dimensional space. This is done by using a non-linear kernel to map the input data to the higher dimensional space. We trained an SVR model using 5-fold time-series cross-validation and radial basis functions as the kernels.

3.7. LSTM (Long Short Term Memory)

Long Short-Term Memory networks, or LSTMs, are a type of recurrent neural network (RNN) architecture designed to overcome the limitations of traditional RNNs in capturing long-term dependencies in sequential data. The primary challenge that LSTMs address is the vanishing gradient problem, which occurs when training deep neural networks with long sequences (Staudemeyer & Morris, 2019).
This comes to particular use in our use case as we want to model a long-term time series problem effectively.
We performed a hyperparameter optimization of the LSTM model over the number of epochs trained (10, 50), batch size (10,32) and the number of LSTM units (16,32,64). We found the best-performing model was trained for 10 epochs with a batch size of 10, containing 32 LSTM units. It has a total of 5537 trainable parameters, with 5504 LSTM parameters and 33 parameters for the final dense layer to predict the registered of casual riders.
As in section 4, it is clear that the LSTM model performs the best in the hourly dataset rather than the daily dataset as it has more datapoints to model its parameters better. LSTMs are also able to retain the time-series information in the model, unlike the Linear Regression or XGBoost models. With more data, one can expect the LSTM model to perform well, even when there are a lot of variations in the data.

4. Results

Results for each model are compared on root mean squared error and R^2 score in Tables 3, 4 and 5. XGBoost and the Random Forest Regressor consistently out performed the other models with near perfect R^2 scores and comparatively low RMSE values.

Table 3. RMSE and R^2 for hourly registered dataset

MODEL	RMSE	R^2
LINEAR REG.	831.3	0.83
RF REG.	10.64	0.99
RF REG W/ PCA	10.14	1.00
XGBOOST	36.0	0.95
XGBOOST W/ PCA	1.90	1.00
LSTM	117.05	0.61
LSTM W/ PCA	112.75	0.64

Table 4. RMSE and R^2 for daily registered dataset

MODEL	RMSE	R^2
LINEAR REG.	925.06	0.76
RF REG.	80.11	1.00
RF REG W/ PCA	81.02	1.00
XGBOOST	523	0.90
XGBOOST W/ PCA	42.46	1.00
LSTM	915.02	0.66
LSTM W/ PCA	968.07	0.62

5. Key Limitations

The main limitation of this work is the size of the dataset. Modeling yearly trends with only two years of data is challenging, especially when trying to execute a train/test split. This is expected to be the main contributing factor to the issues with the classical time series method in the TBATS model as well as the the less accurate results of the LSTM. For both models, increasing the input data size would likely increase their accuracies.

6. Future Work

With resources to handle large-scale data constantly improving, continuously validating and updating the model as new data becomes available is important for the model to be robust to changes. The current dataset is dated at 2011, and is trained on data worth two years. Other basic improvements like dummy coding input features like ”weekday” or ”month” would improve the performance on linear models further. Introducing real-time prediction capabilities will be useful as the model will learn to adapt to sudden changes in demand. This can be done by exploring the use of RL for optimizing bike-sharing operations in real-time. Integrating other data sources through social media and location information can provide insights into user behaviour, sentiments and preferences. Using ensemble models to combine the inferences from different models could be interesting to see how the model can perform in comparison to the XGBoost model. For model interpretation, exploring other frameworks like LIME and ELI5 could lead to exciting results.

Table 5. RMSE and R^2 for hourly casual dataset

MODEL	RMSE	R^2
LINEAR REG.	36.0	0.88
RF REG.	4.89	0.99
RF REG W/ PCA	4.91	0.99
XGBOOST	14.5	0.91
XGBOOST W/ PCA	1.08	1.00
LSTM	20.13	0.87
LSTM W/ PCA	20.14	0.87

Table 6. RMSE and R^2 for daily casual dataset

MODEL	RMSE	R^2
LINEAR REG.	342	0.71
RF REG.	144.57	0.97
RF REG W/ PCA	116.33	0.97
XGBOOST	251	0.84
XGBOOST W/ PCA	38.92	1.00
LSTM	450.14	0.59
LSTM W/ PCA	484.0	0.53

Software

The models are all created using the $sklearn$ libraries and the $Tensorflow$ libraries available. All code for models described in this report can be found at https://github.com/asvin-kumar/bikedemandprediction.

References

Alysha M. De Livera, Rob J. Hyndman, R. D. S. Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106:1513–1527, 2011.

Bikeshare, C. All about capital bikeshare. https://capitalbikeshare.com/how-it-works, a.

Bikeshare, C. Pricing. https://capitalbikeshare.com/pricing, b.

Breiman, L. Random forests. *Machine Learning*, 45:5–32, 2001.

Britton, M. VINE: Visualizing statistical interactions in black box models. April 2019.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16. ACM, August 2016. doi: 10.1145/2939672.2939785. URL http://dx.doi.org/10.1145/2939672.2939785.

Fanaee-T, H. Bike Sharing Dataset. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5W894.

Kaggle. Bike sharing demand. https://www.kaggle.com/competitions/bike-sharing-demand/overview.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4765–4774. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf`.

Martinez, M. Washington, d.c., launches the nation's largest bike share program. *Grist*, 2010.

Pascale, J. Capital bikeshare hits highest monthly use ever, showing gains out of the pandemic. *DCist*, 2023.

Petneházi, G. Recurrent neural networks for time series forecasting. *ArXiv*, abs/1901.00069, 2018. URL `https://api.semanticscholar.org/CorpusID:57373758`.

Pérez-Moreno, H. Growth in e-bikes, d.c. tourism help to fuel bikeshare ridership record. *The Washington Post*, 2023.

Srinivasan, V. Eda ensemble model (top 10 percentile). https://www.kaggle.com/code/viveksrinivasan/eda-ensemble-model-top-10-percentile.

Staudemeyer, R. C. and Morris, E. R. Understanding lstm– a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*, 2019.

Subbaswamy, A., Schulam, P., and Saria, S. Preventing failures due to dataset shift: Learning predictive models that transport. December 2018.

Weisberg, S. *Applied linear regression*, volume 528. John Wiley & Sons, 2005.

			1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Linear Regression	Casual	Day	workingday	temp	yr	atemp	weathersit	season	weekday	windspeed	hum	mnth	holiday	
Linear Regression	Casual	Hour	workingday	hum	temp	atemp	hr	yr	season	weekday	weathersit	holiday	windspeed	mnth
Linear Regression	Registered	Day	yr	season	workingday	atemp	weathersit	temp	windspeed	mnth	weekday	hum	holiday	
Linear Regression	Registered	Hour	hr	yr	hum	atemp	season	workingday	temp	windspeed	weathersit	weekday	holiday	mnth
Support Vector Regressor	Casual	Day	workingday	temp	yr	atemp	weathersit	season	weekday	hum	windspeed	mnth	holiday	
Support Vector Regressor	Casual	Hour	workingday	hum	temp	atemp	hr	yr	weekday	season	weathersit	holiday	windspeed	mnth
Support Vector Regressor	Registered	Day	yr	season	atemp	workingday	weathersit	temp	mnth	windspeed	weekday	hum	holiday	
Support Vector Regressor	Registered	Hour	hr	yr	hum	season	atemp	workingday	temp	windspeed	weathersit	weekday	holiday	mnth
XGBoost	Casual	Day	workingday	temp	yr	atemp	weekday	hum	weathersit	windspeed	mnth	season	holiday	
XGBoost	Casual	Hour	hr	temp	weekday	workingday	yr	atemp	mnth	hum	season	weathersit	windspeed	holiday
XGBoost	Registered	Day	yr	temp	workingday	season	atemp	hum	mnth	weathersit	windspeed	weekday	holiday	
XGBoost	Registered	Hour	hr	yr	workingday	temp	season	hum	mnth	weathersit	weekday	atemp	windspeed	holiday
XGBoost (PCA)	Casual	Day	weekday	workingday	temp	holiday	weathersit	windspeed	season	atemp	yr	hum	mnth	
XGBoost (PCA)	Casual	Hour	weekday	workingday	holiday	weathersit	temp	yr	windspeed	hum	hr	season	mnth	atemp
XGBoost (PCA)	Registered	Day	yr	season	holiday	workingday	weathersit	mnth	weekday	windspeed	hum	atemp	temp	
XGBoost (PCA)	Registered	Hour	mnth	holiday	yr	season	workingday	windspeed	hr	weekday	weathersit	hum	atemp	temp
Random Forest Regressor	Casual	Day	workingday	temp	atemp	yr	weekday	hum	weathersit	windspeed	mnth	season	holiday	
Random Forest Regressor	Casual	Hour	hr	workingday	temp	yr	atemp	hum	weekday	mnth	season	weathersit	windspeed	holiday
Random Forest Regressor	Registered	Day	yr	temp	workinday	atemp	mnth	season	hum	weathersit	windspeed	weekday	holiday	
Random Forest Regressor	Registered	Hour	hr	yr	workinday	temp	atemp	hum	mnth	season	weathersit	weekday	windspeed	holiday

Table 7. List of features in decreasing priority order for each model from left to right

A. Model Interpretation

The models developed to predict the time series data were interpreted using the publicly available SHAP python toolbox (Lundberg & Lee, 2017). SHAP uses game theoretic approaches to assign credit to the input features of data points towards predicting the output. The SHAP value for each data point represents the impact each input feature has on the target. For a specific data point, adding a ”base rate” along with the SHAP values for all the input features sums up to the target value. In this framework, features with larger magnitudes of SHAP values have higher impacts on the outputs of trained models while features with SHAP values closer to 0 have lower impact.

Some sample SHAP summary plots are shown in Figures 8, 9, 10, 11. The x-axis shows the SHAP value and the y-axis shows the features in their decreasing order of importance from top to bottom. In each figure, the summary plot on the left is for daily user data while the summary on the right is for hourly user data. Each data point is represented as a dot in the summary plot and are colored based on their min and max values according to the colorbar shown. This gives us insights into how the low values and high values of a certain feature tend to affect the model's output.

Table 7 summarizes these results from all the models we interpreted using SHAP. Here are our takeaways:

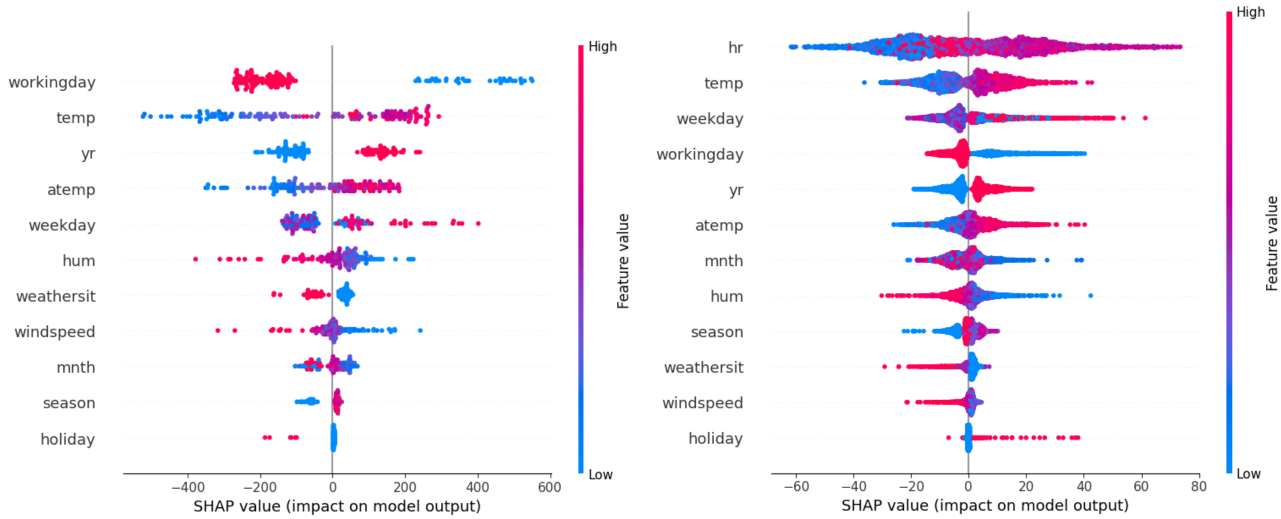
1. From figures 8 and 9, we notice that both the tree based models give high priorities to similar features. We see that daily casual rider count depends on factors like workingday, temperature, and humidity. Workingday and weekday are highly correlated. The year feature is important because ridership increased overall from 2011 to 2012. For hourly data, the most important factor is naturally the hour of the day followed by the other factors we saw for daily data.

2. Looking at figure 9, we see that the year feature is still high on the list. Once again features like temp and workingday are important but it's interesting to note that the yr feature is higher on the list than all these other factors. We also see features like month and season rising up the list. This is probably because registered users make a commitment to using bikes by purchasing annual memberships and follow through with regular bike usage. We also notice that the holiday feature contributes the least and that's probably because of how sparse the data is and how rarely it helps with predicting the target.

3. Looking at the summary plot for the linear regression model in figure 10, we see how the binary variable “working day” is now at the top of the list and how the numeric features like weekday and month are lower. Weekday values, for example, are assigned from 1-7. These values make sense to us as categories but are not as useful to the linear model. Dummy coding these features would've been more useful to the model.

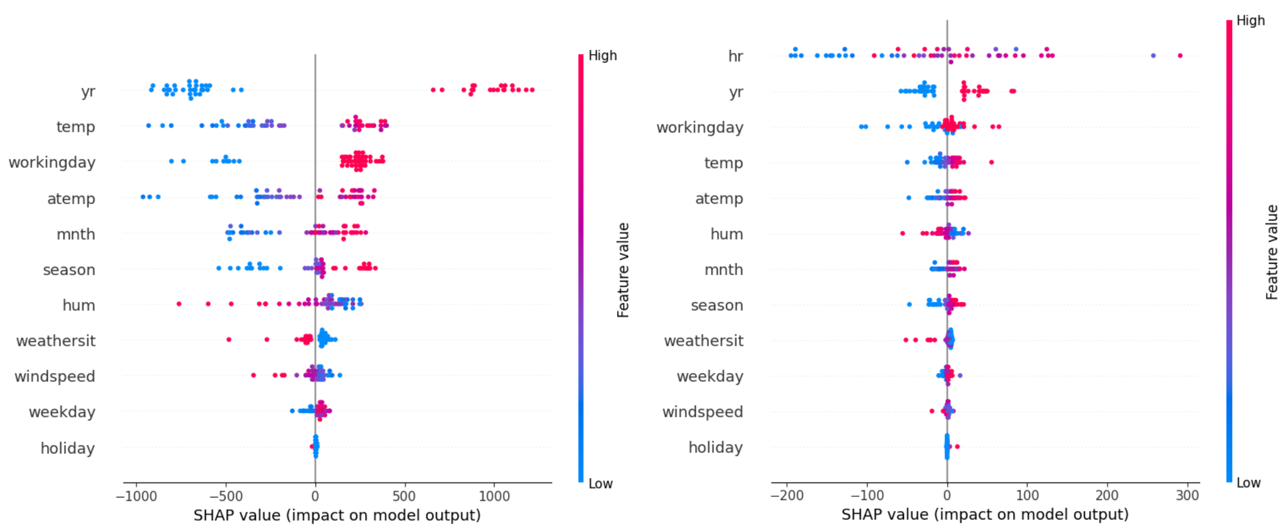
4. Finally, figure 11 shows the SHAP plot for XGBoost but with PCA applied to the input features to restrict information to the top two PCA components. We observe that all the information for predicting the model output comes from just about 2-3 input features.

Overall, the takeaways from the models developed align with intuition. Having more complex input features could enable better interpretation and conclusions.



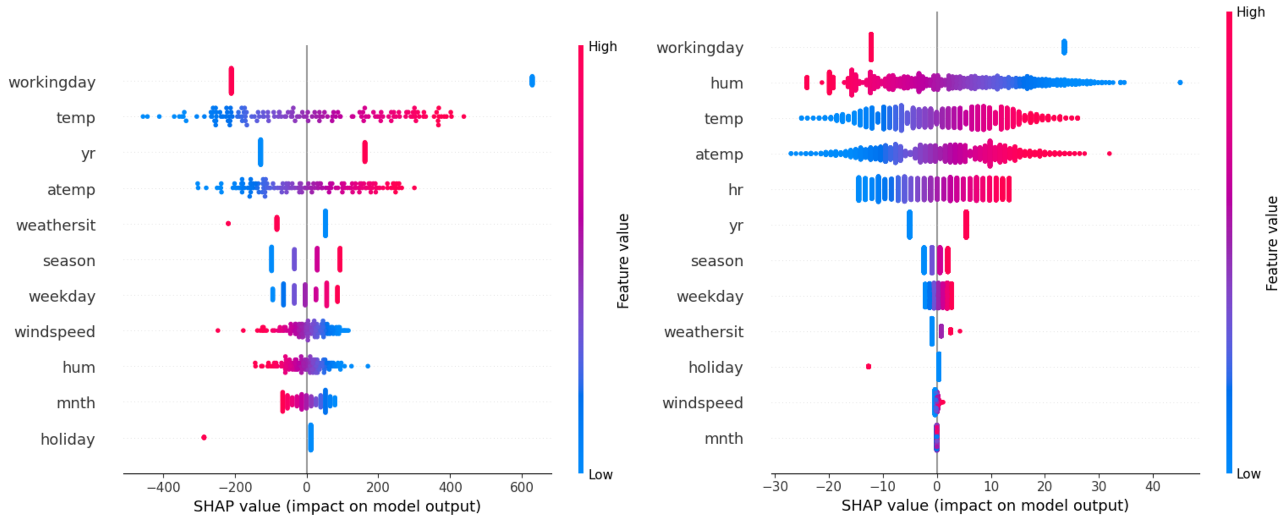
(a) casual, daily (b) casual, hourly

Figure 8. SHAP summary plots for the following configurations (rider type, time scale) of the XGBoost model



(a) registered, daily (b) registered, hourly

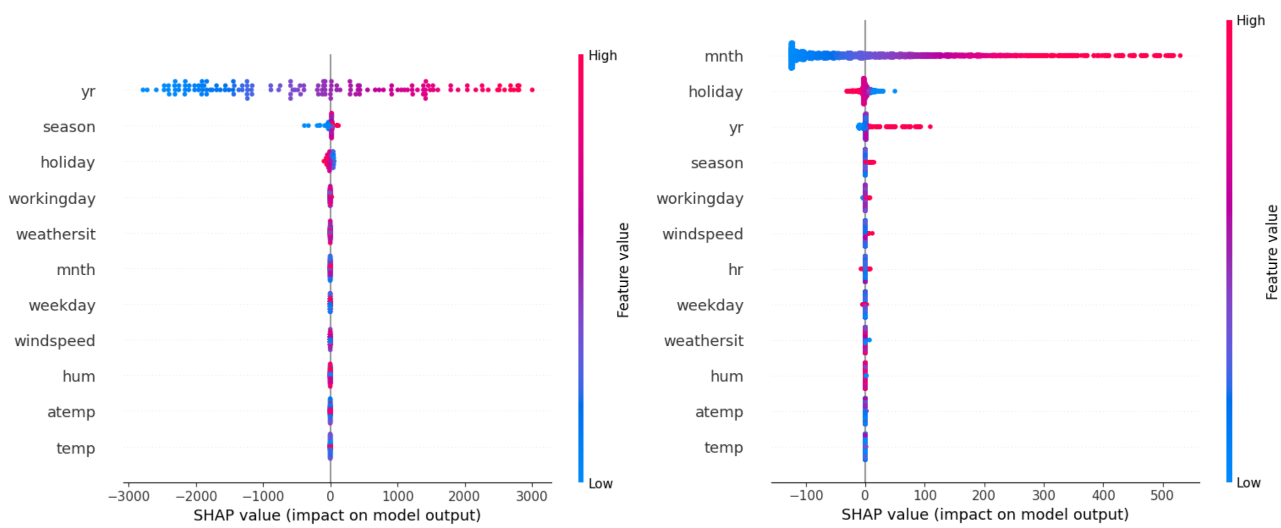
Figure 9. SHAP summary plots for the following configurations (rider type, time scale) of the Random Forest model



(a) casual, daily

(b) casual, hourly

Figure 10. SHAP summary plots for the following configurations (rider type, time scale) of the Linear Regression model



(a) registered, daily

(b) registered, daily

Figure 11. SHAP summary plots for the following configurations (rider type, time scale) of the XGBoost (PCA) model