2023

Home Exercise 2

TWAN GERJO VELDHUIS, H22TWAVE@DU.SE

Task 1: Predictive Modelling for Bike Rental Data

Introduction

Bike rental systems have gained significant popularity in urban environments as a sustainable and efficient means of transportation. In this project, we explore the application of regression algorithms on the supplied bike rental dataset to predict the number of bikes rented based on the features. The dataset comprises information from approximately 11,000 individuals and includes various factors such as weather conditions, season, and day, which have a possible influence on the bike rental patterns.

Methodology

The objective of this project is to develop regression algorithms to accurately predict the number of bikes rented within each time slot. The methodology will be explained in the following chapter.

The Dataset

The dataset that will be used is a bike rental dataset that contains a lot of information about the bike rental over a period of 2 years (2011 and 2012), where there are 11,000 hourly observations made about the bike rental itself and external factors that could have an influence on the rental numbers.

Table 1 - dataset description

#	Variables	Variables Description
1.	ID	Time slot's unique identifier (not related to the time order)
2.	Datetime	Time slot
3.	Season	Time slot's seasonal category (1 = winter, 2 =spring, 3 = summer, 4 = autumn) to identify the different seasons
4.	Holiday	Time slot's identifier if the day is a holiday
5.	Workingday	Time slot's identifier if the day was a working day (neither a holiday nor a weekend)
6.	Weather	Time slot's identifier of weather quality ranging from 1 to 4 suggesting best to worst weather
7.	Temp	Time slot's temperature in degrees Celsius
8.	Atemp	Time slot's temperature sensation in degrees Celsius
9.	Humidity	Time slot's relative humidity
10.	Windspeed	Time slot's current wind speed (km/h)
11.	Count	Time slot's identifier for total number of rentals

These are all the different variables before the pre-processing, then there was done some outlier detection. In regard to the outliers there were no outliers detected in the data, so in the end this was no problem. Then feature engineering was applied, where first of all the datetime variable was split into five different variables and the datetime variable was afterwards deleted.

Table 2 - datetime extracted variables

#	Variable	Variable Description
1.	Weekday	Time slot's day of the week from 1 to 7 (1 = Monday, 2 = Tuesday, etc..)
2.	Hour	Time slot's hour of the day from 0 (00:00) to 23 (23:00)
3.	Day	Time slot's day of the month
4.	Month	Time slot's month of the year
5.	Year	Time slot's identifier of the year

After the creation of these columns the "temp", "atemp", and "windspeed" variables were normalized with a mean of 0 and a standard deviation of 1. This was done to improve the predictive performance in the upcoming data mining models, especially the SVM model which is highly influenced if features have different scales.

The dataset is also checked on correlations, these are as shown in Figure 1. Here is shown that there is not a lot of correlation/linearity in regard to the "count" variable, which has only a correlation of around the 0.4 with the "hour", "temp", "atemp", and "windspeed" variables. This indicates that the regression model will probably have a hard time with predicting the Furthermore the "temp", "atemp", and "windspeed" variables have a high correlation with eachother, this will now not yet be deleted. But have quite the possibility to be deleted during the data mining steps.

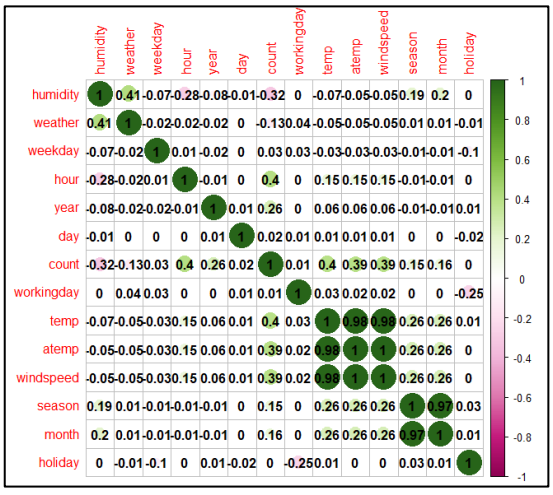

Figure 1 - correlation matrix

In the end the dataset 14 variables which are: "season", "holiday", "workingday", "weather", "temp", "atemp", "humidity", "windspeed", "weekday", "hour", "day", "month", "year", and "count". Which all are featured engineered, cleaned and scaled to now be used in the models.

This description was done for the "train.csv"-file, but can also be used for the "test.csv"-file, where the same process was done to pre-process the data for the prediction. The only difference is that in the "test.csv"-file there was no "count" variable, because this variable needs to be tested by our model.

Data mining method(s)

In this project we had to use at least three different models, two of these were already decided. One of them was the Support Vector Machine (SVM) model, and the second one was a decision tree-based model. The third model will be a linear regression model, which is a supervised learning algorithm that works really wel for predicting continuous numerical values.

In general, the data is first pre-processed (which is explained in the previous chapter). Then the train dataset – from the "train.csv"-file – will be used to test the model by splitting the data 30/70. This split is chosen, because this is the general standard for splitting the data into train and test data.

After the splitting of the model backward feature selection is executed to be able to select the most important features and reduce dimensions for the models. Feature selection is chosen, because it helps in reducing noise and computational performance while improving the model's interpretability. This version of feature selection is chosen, because in general it is the best performing feature selection method. And at last, all the methods will be tested using all features and the features that were selected by the backward selection model and it will be trained using 10-fold cross validation to get a estimation of the model performance and get the best results.

Linear Regression

Linear regression is a widely used data mining model that aims to create a linear relationship between a dependent variable (target) and one or more independent variables (features). As mentioned earlier, the algorithm is supervised and works really well for predicting continuous numerical values. In the context of the bike rental dataset, the linear regression model was chosen for several reasons. First of all the linear regression algorithm functions as a good baseline, against where more complex models can be compared to. This will help with evaluation of the performance of other models. Secondly, is the simplicity and interpretability of linear regression, which allow us to easily understand and explain the relationship between the target variable and the features. And lastly, linear regression allows for assessment of any violations of its assumptions and make any necessary adjustments or explore alternative models if the assumptions are violated. By using this model, we can build a predictive model that estimates the number of bikes rented based on various attributes such as weather conditions, seasonality, and day of the week.

Support Vector Machine (SVM)

SVM (Support Vector Machine) for regression, also known as Support Vector Regression (SVR), is a supervised learning algorithm used for solving regression problems. Unlike other regression models, SVR aims to find a hyperplane that best fits the data points while minimizing the error within a specified tolerance range. To model non-linear relationships, SVR uses a kernel trick, which maps the data to a higher-dimensional space. This mapping allows for capturing complex patterns that may not be linearly separable in the original feature space. Next to this advantage, this model is also chosen its ability to handle non-linearity, robustness to outliers, generalization ability, and flexibility. It is really good at capturing complex relationships in the bike rental dataset, handles outliers effectively, generalizes well on unseen data, and offers adaptability through various kernel functions. The model will furthermore be tuned on the cost (penalty for misclassifying data points) and gamma (influence of a single data point on the model) arguments.

Decision trees

Decision trees are a data mining model used in this project due to their interpretability, versatility, and ability to handle both categorical and numerical features. Decision trees are non-parametric models that partition the dataset based on the most effective features, creating a tree-like structure. The decision tree algorithm was selected for this analysis because it can capture complex relationships between the predictor variables (such as weather conditions, season, and day) and the target variable (number of bikes rented). Decision trees provide a clear visualization of the decision-making process, allowing for easy interpretation and understanding of the factors influencing bike rental patterns. Furthermore, decision trees are well-suited for handling both continuous and discrete target variables, making them an appropriate choice for predicting the number of bikes rented within each time slot. Additionally, decision trees are robust against outliers and missing values, as they can handle both cases effectively without requiring extensive data pre-processing.

To tune the decision trees the choice was made to first make a decision on how to grow the trees using either bagging, boosting or random forest. Bagging is an technique that combines multiple models by training them on different subsets of the data, reducing variance and improving stability and will only be tuned on the number of trees. Boosting sequentially builds weak models, adjusting weights to focus on misclassified instances, and combines their predictions to create a strong model. This technique will be tuned on the number of trees and interaction depth (maximum depth of one tree). Random Forest combines bagging with random feature selection, constructing multiple decision trees on bootstrapped samples with randomly selected features, resulting in an accurate and robust ensemble model. Which won't be tuned on any arguments.

Results

The selected features from the backward selection that were used in the end were "season", "atemp", "humidity", "weekday", "hour", "month", and "year".

In regard to the models we executed all the models as earlier said with and without backward selection, but all with 10-fold cross validation. Appendix A provides an overview of the performance metrics for various regression models applied to bike rental prediction. The models include linear regression without backward selection (BS), linear regression with BS, Support Vector Machine (SVM) without BS, SVM with BS, Bagged Decision Tree without BS, Bagged Decision Tree with BS, Boosted Decision Tree without BS, Boosted Decision Tree with BS, Random Forest without BS, and Random Forest with BS.

The mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), R-squared (R^2), and adjusted R-squared are used as evaluation metrics to assess the accuracy and goodness-of-fit of the models.

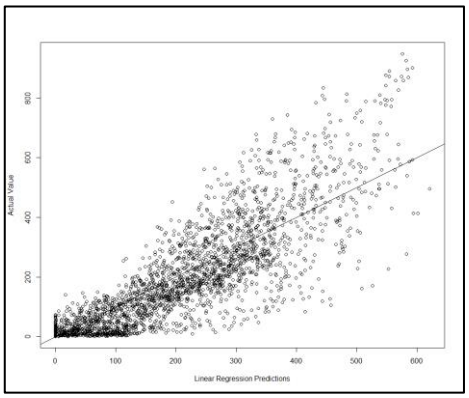

Figure 2 - best scoring linear regression model

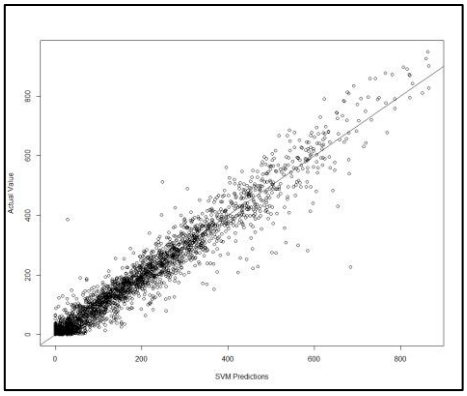

Figure 3 - best scoring svm model results

In terms of MSE and RMSE, lower values indicate better predictive performance. The linear models, both with and without backward selection, exhibit relatively higher MSE and RMSE compared to the other models. This suggests that the linear models may not capture the nonlinear relationships between the predictors and the target variable effectively, which we could also conclude from the correlation matrix. On the other hand, the SVM models, bagged decision tree models, boosted decision tree models, and random forest models demonstrate lower MSE and RMSE, indicating superior prediction accuracy.

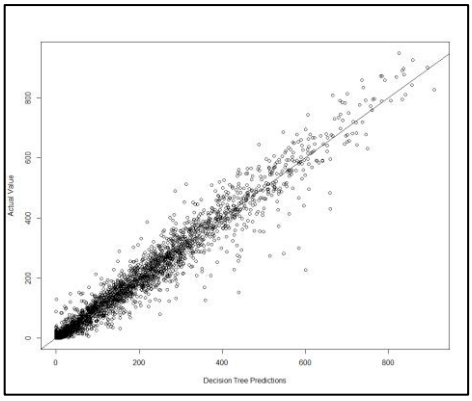

Figure 4 - best scoring decision tree model

The MAE metric measures the average absolute difference between the predicted and actual values. Similar to MSE and RMSE, lower MAE values indicate better model performance. Among the models, the boosted decision tree without BS yields the lowest MAE values, suggesting its ability to make predictions with lower overall error.

Moving on to the R-squared and adjusted R-squared values, these metrics provide an indication of how well the models explain the variance in the target variable. Higher values of R-squared and adjusted R-squared signify a better fit. All the models, except the linear regression model, show a high R-squared and adjusted R-squared value, indicating a strong ability to explain the variability in bike rental predictions.

Limitations

Regarding these results, it must be taken into consideration that the models that were executed were not fully tuned, because the computer crashed when running SVM in a tuning scenario. This is why the parameters are not perfectly optimized, but done to the best of the researchers ability.

Next to this, it can be that the models had different arguments that would perform better – even in cross validation – but have a lot of bias and relatively high parameters. Even though the assumably high bias, I would still choose these models. Because the models have been validated using cross validation.

Conclusion

In conclusion, our analysis of various regression models for predicting bike rentals revealed that the Support Vector Machine (SVM) models, bagged decision tree models, boosted decision tree models, and random forest models outperformed the linear regression models. These models demonstrated superior accuracy in predicting the number of bikes rented, as indicated by lower mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE) values. Additionally, the boosted decision tree models showed the highest R-squared and adjusted R-squared values, indicating their strong ability to explain the variability in bike rental predictions. In summary, the predictive modelling results presented in this study provide valuable insights into accurately forecasting bike rental demand. The superior performance of SVM, bagged decision trees, boosted decision trees, and random forests suggests their potential for enhancing operational efficiency and customer satisfaction in bike sharing systems. This research lays the foundation for further advancements in predictive analytics for bike rental systems, paving the way for data-driven decision-making and optimization of urban transportation networks.

Task 2: Clustering of Bike Rental Data

Introduction

This second task (and at the same time project) is again about bike rental cant popularity in urban environments as a sustainable and efficient means of transportation. In this project, we delve into clustering analysis to partition the dataset into homogeneous groups. By combining the two datasets, we try to extract meaningful insights and identify clusters that exhibit similar or different characteristics in terms of bike rental patterns. We investigate whether the resulting clusters separate the dataset based on the target variable, getting it more clear if there are relationships between different rental patterns and their attributes.

Methodology

The objective of this project is to identify patterns in the data by clustering it into different clusters and identifying if the dependent variable can be separated using clustering. How this is done, will be explained in the following chapter.

The Dataset

For the description about the dataset there will be referenced to the dataset description in the previous task. This is due to both the datasets being exactly the same in both projects.

Data mining method(s)

In this project K-means clustering was used for the unsupervised clustering, which was chosen because of it's simplicity and simplicity compared to other unsupervised clustering methods.

In general, the data is first pre-processed (the method for this is referenced in the previous chapter). The data is in total then combined without the dependent variable (count) and because k-means only supports numerical values and not categorical variables, the data checked on numeric only and then it goes to PCA.

After this is done Principal Component Analysis is applied to reduce de dimensionality of the dataset from 13 variables to 3 variables, which accounts for ~96% of the variance in the data. This is done so we can see if it will be possibility to quickly identify if the data is at all separatable. Next to this the PCA also helps with reducing the dimensionality of the data, so to improve the clustering of the k-means model by giving the cluster less dimensions to work with.

K-means clustering

K-means clustering is a widely used data mining model for unsupervised learning tasks, including clustering analysis. It is a clustering algorithm that aims to group similar data points together into homogeneous clusters based on their feature similarities. Which it does by minimizing the within-cluster sum of squares by iteratively assigning data points to the nearest centroid and updating the centroids until convergence. This model was chosen over other models, because of its simple yet effective clustering method that is computationally efficient and scalable, making it suitable for handling large datasets with numerous dimensions. Secondly, K-means clustering allows us to uncover latent structures and patterns within the dataset without the need for prior knowledge or labelled data. As an unsupervised learning technique, K-means does not rely on a target variable but instead identifies clusters based on the inherent similarities in the feature space. This enables us to achieve the aim of this project and to answer if it is possible to separate the dependent variable using clustering.

The main parts for tuning this model will lay in the preparation of the data itself, starting of with normalization, one-hot encoding, and other techniques. The only tuning that can be done within the method is the number of clusters that will be used for the separation of the points. This will be tuned using the elbow method (within-cluster sum of squares (inertia) against the number of clusters) and the average silhouette method (how well each data point fits within its assigned cluster in K-means clustering).

Results

The model has been tested executed using 3 different clusters which is based on the, previously mentioned, elbow method and the silhouette method.

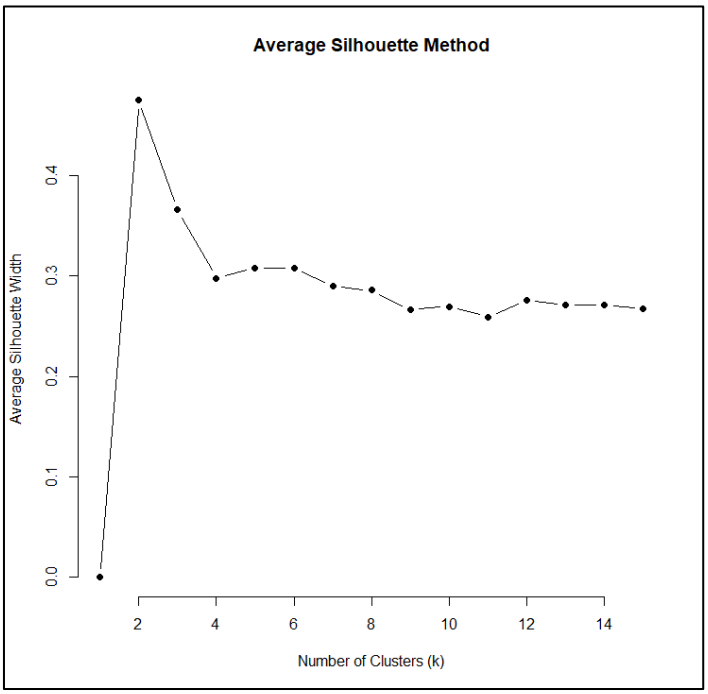

Figure 6 - average silhouette graph

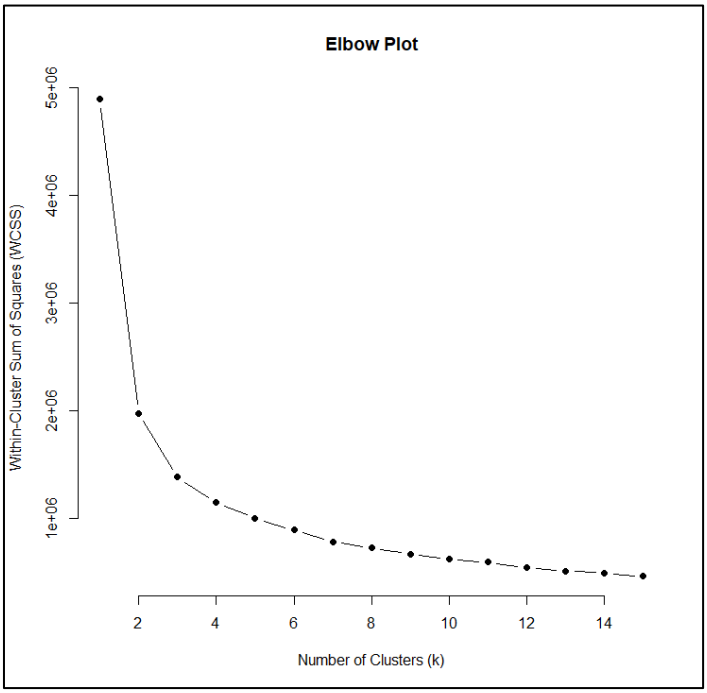

Figure 5 - elbow plot

The k-means clustering model produced a total within-cluster sum of squares (WSS) value of 1,396,021. The WSS reflects the sum of the squared distances between each data point and its assigned cluster centroid. A lower WSS indicates tighter and more cohesive clusters, suggesting a better separation of data points within each cluster.

The analysis of the actual count values for the train data within each cluster provides insights into the rental patterns and demand levels associated with each cluster.

Cluster 1: The summary statistics for the count values within Cluster 1 reveal that the minimum count is 1, indicating instances where only a single bike was rented. The first quartile (25th percentile) is 44, while the median (50th percentile) is 152. The mean count value for Cluster 1 is approximately 194.4. The third quartile (75th percentile) is 285, indicating that a majority of the rental instances in this cluster fall

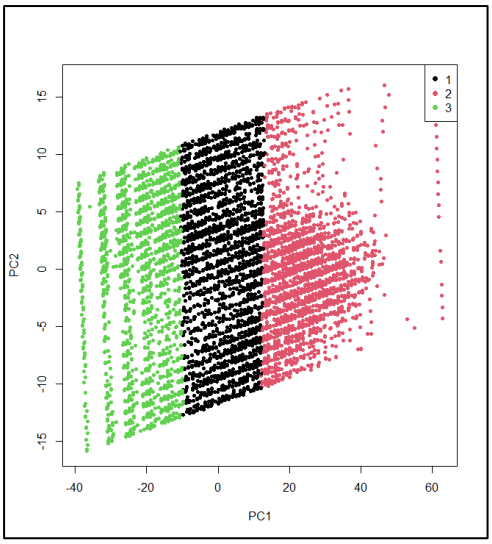

Figure 7 - PC1 and PC2 plot with cluster-based colouring

below this value. The maximum count value is 901, reflecting instances of high demand for bike rentals within Cluster 1.

Cluster 2: Within Cluster 2, the summary statistics reveal that the minimum count value is 1, similar to Cluster 1. The first quartile is 123, while the median count value is 230. The mean count for Cluster 2 is approximately 267.1, indicating a slightly lower average demand compared to Cluster 1. The third quartile is 375, suggesting that a significant proportion of the rental instances in this cluster fall below this value. The maximum count value is 970, indicating instances of relatively higher demand for bike rentals within this cluster.

Cluster 3: The summary statistics for Cluster 3 show a minimum count value of 1, which is consistent with the other clusters. The first quartile is 18, and the median count value is 74. The mean count value for Cluster 3 is approximately 127, indicating a lower average demand compared to the other clusters. The third quartile is 181, suggesting that a majority of the rental instances within this cluster fall below this value. The maximum count value is 839, reflecting instances of moderate demand for bike rentals within Cluster 3.

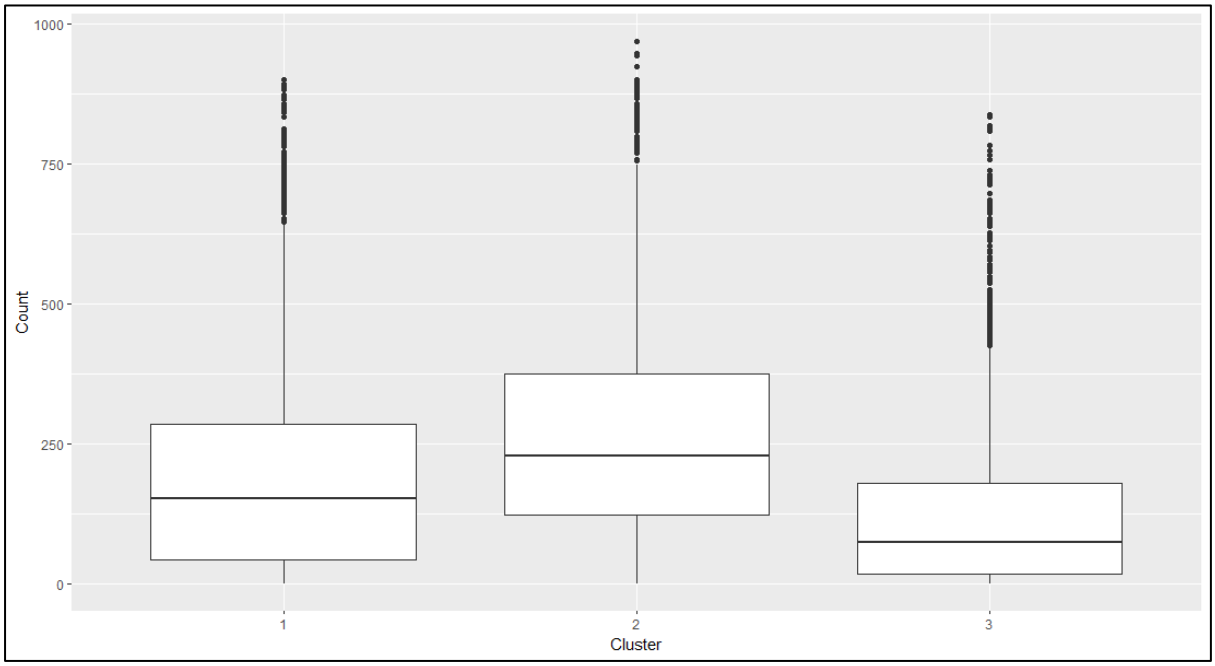

Figure 8 - boxplots of actual cluster counts

Finally, using the ANOVA approach we can support our results if the clustering has a significant impact on the count variable. This is done by using the p-value that comes out of this technique, which is in this case a p-value that is less than 0.001. This indicates that the clusters have a substantial impact on the variable. Also the F-value of 446.6 suggests that the variability between clusters is significantly higher than the variability within clusters.

Limitations

Even though these results all sound really promising, there are still some things to consider when interpretating the following results. First of all, the k-means algorithm assumes clusters are homogeneous, but real-world data may have variations and overlapping characteristics within clusters. And next to this the findings and clusters are specific to the bike rental dataset used and may not directly apply to other systems. Validation on different datasets and contexts is needed for generalizability.

Conclusion

The analysis of the clustering results reveals valuable insights into the relationship between the clusters and the target variable, which in this case is the bike rental count. By examining the summary statistics of each cluster, we can observe distinct characteristics ("humidity" and "hour") that differentiate them based on rental counts.

Cluster 1 represents a group with relatively lower rental counts overall. The minimum rental count in this cluster is 1, with a median of 152 and a maximum of 901. The mean rental count for Cluster 1 is 194.4, and the third quartile falls at 285.0. These statistics suggest that Cluster 1 encompasses a subset of individuals or time periods with comparatively lower bike rental demand.

In contrast, Cluster 2 exhibits a higher range of rental counts compared to Cluster 1. With a minimum rental count of 1, a median of 230, and a maximum of 970, this cluster represents a group that experiences a wider variation in rental demand. The mean rental count for Cluster 2 is 267.1, with the third quartile at 375.0. These statistics indicate a relatively higher level of bike rental activity within this cluster.

Cluster 3, similar to Cluster 1, showcases lower rental counts overall. The minimum rental count is 1, the median is 74, and the maximum is 839. The mean rental count for Cluster 3 is 127, with the third quartile at 181. This cluster captures a segment with relatively lower bike rental demand, similar or even lower to Cluster 1.

To assess the significance of the relationship between the clusters and the target variable, an analysis of variance (ANOVA) was performed. The results from the ANOVA table indicate a highly significant relationship between the clusters and the bike rental count. With an F-value of 446.6 and a p-value of less than 0.001 ($p < 2e\text{-}16$), we can confidently conclude that the clusters separate based on the target variable.

In summary, the clustering analysis successfully identified distinct clusters based on the bike rental count. Each cluster represents a different segment with varying levels of rental activity. The statistical analysis confirms the significance of the relationship between the clusters and the target variable. These findings provide valuable insights into the rental patterns and demand levels, enabling further analysis and the development of targeted strategies to optimize bike rental services.

Appendix A: task 1 results table

In this table BS will be used as a short term for Backward Selection.

Table 3 - prediction model evaluation results

Model	Mean Squared Error (MSE)	Root Mean Squared Error (RMSE)	Mean Absolute Error (MAE)	R-squared (R^2)	Adjusted R-squared
Linear Model without BS	9118.327	95.490	68.424	0.717	0.716
Linear Model with BS	9485.103	97.391	69.633	0.705	0.704
SVM without BS	2009.561	44.828	29.964	0.938	0.937
SVM with BS	2689.717	51.862	31.764	0.916	0.916
Bagged Decision Tree without BS	2376.535	48.750	29.381	0.926	0.925
Bagged Decision Tree with BS	2372.05	48.703	29.285	0.926	0.926
Boosted Decision Tree without BS	1632.539	40.405	25.784	0.949	0.949
Boosted Decision Tree with BS	2354.772	48.525	29.018	0.927	0.926
Random Forest without BS	2350.343	48.480	31.971	0.927	0.926
Random Forest with BS	2432.493	49.320	30.143	0.924	0.924