Predicting Wind Turbine Performance

INST737 Milestone 3

Nour Ali Ahmed, Zahra Halimi, Kurubel Belay

Introduction

The significance of incorporating renewable energy sources into our energy portfolio is widely recognized. One avenue that nations are exploring to bolster their renewable energy capacity is harnessing wind energy. In this study, our aim is to evaluate the key design parameters of wind turbines and forecast their nominal capacity based on these parameters. To accomplish this, we employ a combination of Support Vector Machines (SVMs), neural networks, and clustering techniques to construct our predictive model. Furthermore, we utilize cross-validation methods to gauge the performance accuracy of these models. In addition, we conduct feature selection to discern any enhancements in performance attributable to this process. Lastly, we delve into potential ethical considerations associated with our research findings.

Question 1. SVMs

Support Vector Machines (SVMs) are robust machine learning algorithms renowned for their versatility in handling both classification and regression tasks. These models excel in identifying optimal hyperplanes to separate classes in classification problems or predict continuous outcomes in regression scenarios. In this section, our analysis commences with data preprocessing, involving the removal of missing values, resulting in a dataset comprising approximately 68943 observations. Subsequently, we standardize the dataset to ensure uniformity and enhance model performance. With our dependent variable being continuous, we opt for the **regression SVM** method. Moreover, we decided to use **one vs one** approach to increase the accuracy of the develped models.

To conduct our analysis rigorously, we allocate 75% of the data for training and reserve the remaining 25% for testing, adhering to standard practices in machine learning experimentation. Additionally, we explore multiple SVM approaches including linear, Gaussian, and polynomial SVMs. By employing this diversified methodology, we aim to discern the most suitable model for our dataset and research objectives. In the following sections, the results of each approach are shown.

Linear SVM

In this section, we embark on utilizing Linear Regression Support Vector Machine (SVM) methodology to forecast the continuous variable, turbine capacity (t_cap), leveraging the predictive potential of design features t_rd, t_hh, and t_ttlh. Our approach is anchored in the selection of the epsilon-support vector regression (ε-SVR) variant of SVM, tailored to the continuous nature of the dependent variable.

By adopting the ε-SVR framework, we extend the capabilities of traditional SVMs to regression tasks, aiming to minimize margin violations while accommodating a predetermined tolerance (ε) for deviation from the regression line. As shown in **Figure 1,** number of support vector for this model is 34228 and the training error is around 0.24.

```
Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 1

Linear (vanilla) kernel function.

Number of Support Vectors : 34228

Objective Function Value : -12272.51
Training error : 0.244127
```

Figure 1: Linear SVM

In the subsequent phase, the model's performance was evaluated by computing the root mean square error (RMSE) and correlation between predicted and actual turbine capacities. As shown in **Table 1**, the RMSE is approximately 0.5, while the correlation stands at approximately 0.88.

Table 1: Linear SVM Performance

RMSE	0.49099
Correlation	0.8773763

Gaussian Radial Basis Kernel

In this step, we also developed the Radial Basis Function (RBF) kernel to investigate whether altering the dimension of analysis could enhance the model's performance. Here, we designate the kernel parameter as "rbfdot." As illustrated in **Figure 2**, the number of support vectors for this model is 24162, with a corresponding training error of approximately 0.084. In comparison

to the previous model, it appears that altering the dimension could potentially reduce the training error.

```
Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 1

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  3.08367400456928

Number of Support Vectors : 24162

Objective Function Value : -5507.924
Training error : 0.084646
```

Figure 2: Radial Basis Kernel SVM

As a result, the root mean square error (RMSE) and correlation between the predicted turbine capacity and the actual values were computed. As indicated in **Table 2**, the RMSE is 0.3, while the correlation is 0.95. Comparing to the previous model, the correlation has notably increased from 0.88 to 0.95.

Table 2: Radial Basis SVM Performance

RMSE	0.2953918
Correlation	0.9547117

Polynomial Function

In this step, the model was additionally tested with a polynomial function to assess whether it could improve model performance. As depicted in Figure 3, the number of support vectors is 34204, and the training error is 0.24, which exceeds that of the previous model.

```
Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 1

Polynomial kernel function.
 Hyperparameters : degree =  1  scale =  1  offset =  1

Number of Support Vectors : 34204

Objective Function Value : -12273.14
Training error : 0.244121
```

Figure 3: Radial Basis Kernel SVM

As a result, the root mean square error (RMSE) and correlation between the predicted turbine capacity and the actual values were computed. As indicated in Table3, the RMSE is 0.5, while the correlation is 0.88. Comparing to the previous model, the correlation has notably decrease and is closer to first model.

Table 3: Polynomial Function SVM Performance

RMSE	0.4909854
Correlation	0.877383

In conclusion, it appears that employing the Gaussian Radial Basis Kernel yields superior results for this dataset. With a correlation of approximately 0.95 and a lower RMSE compared to the first and third models, it demonstrates enhanced predictive performance.

Question 2. Neural Networks

Continuing with the process of testing the effectiveness of different models to predict turbine capacity using parameters such as hub height, rotor diameter, total turbine height, and rotor swept area, we employed neural networks. In the data cleaning process, we removed any null variables involved and developed a normalization function to scale numeric variables to a common range between 0 and 1. This prevented any individual feature from dominating the model due to differences in scale.

Given the large dataset size of over 6,000 rows post-cleanup, which led to extended processing times, we opted to randomly sample 20% of that dataset to expedite computations. As the selected rows were randomly sampled, the distribution remained relatively similar compared to the original dataset. Subsequently, the preprocessed dataset was partitioned into training and testing sets, with 70% of the data allocated for training and the remaining 30% for testing.

The neural network model architecture was then flexibly configured with varying numbers of hidden layers, enabling experimentation with different complexities. Specifically, the script iterated over three configurations: 1 hidden layer, 5 hidden layers, and 10 hidden layers.

The neural network model was developed for each setup using the "neuralnet" function and the training data as input. Notably, the "neuralnet" function parameters included the stepma maximum number of iterations, the number of hidden layers, and other hyperparameters required for model optimization. After confirming that the weights were calculated, the script checked to see if the model had successfully converged during training. The architecture of the trained neural network was shown, and using the testing data, predictions were made on the presence or absence of convergence.

After that, performance metrics were calculated for each model configuration and recorded into a file called "neuralnet_results.txt." These metrics included mean squared error (MSE), root mean square error (RMSE), mean absolute error (MAE), and the correlation coefficient between predicted and actual values. These metrics offered insightful information about the model's precision and accuracy in identifying the underlying patterns in the data.

The analysis of different configurations of neural network models for predicting turbine capacity reveals several significant insights. Firstly, when employing a single hidden layer, the model failed to converge effectively, indicating potential limitations in capturing the complex relationships within the data. This could be due to several reasons. The biggest one is that neural networks with only one hidden layer may struggle to capture the intricate patterns and relationships present in complex datasets such as turbine capacity prediction. With fewer hidden layers, the model may lack the necessary depth to extract and represent the underlying features effectively.

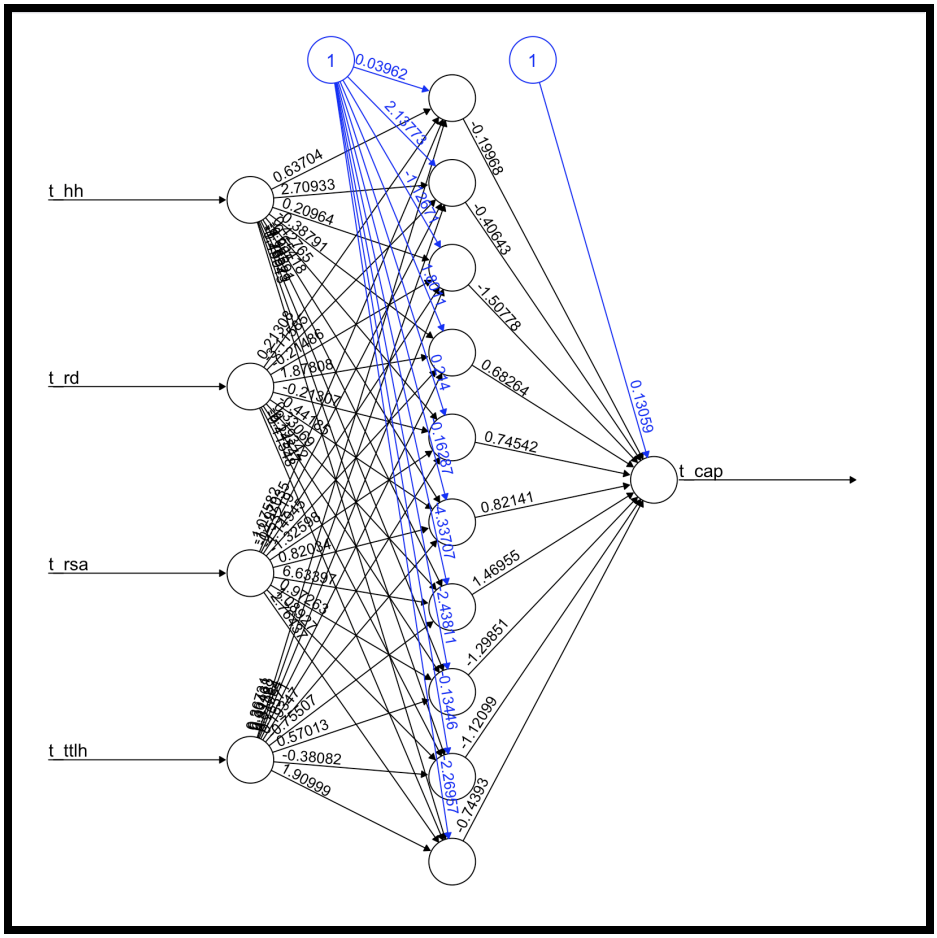


Figure: 10 Hidden Layer Nerual Network Model

However, models with 5 and 10 hidden layers demonstrated successful convergence, leading to robust predictions. Despite the increase in model complexity with additional hidden layers, there was no substantial improvement observed in predictive performance between the 5-hidden-layer and 10-hidden-layer models. Both models exhibited similar levels of accuracy, as reflected in their comparable mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE) values.

Moreover, the high correlation coefficients between predicted and actual turbine capacities suggest a strong linear relationship in both cases. Thus, while increasing model complexity may offer marginal improvements, a neural network with 5 hidden layers appears to strike a balance between complexity and accuracy, making it a practical choice for predicting turbine capacity.

	MSE	MAE	RMSE	Correlation
5 Hidden Layers	0.002319151	0.03418952	0.04815756	0.9330128
10 Hidden Layers	0.00231363	0.03433689	0.0481002	0.9331748

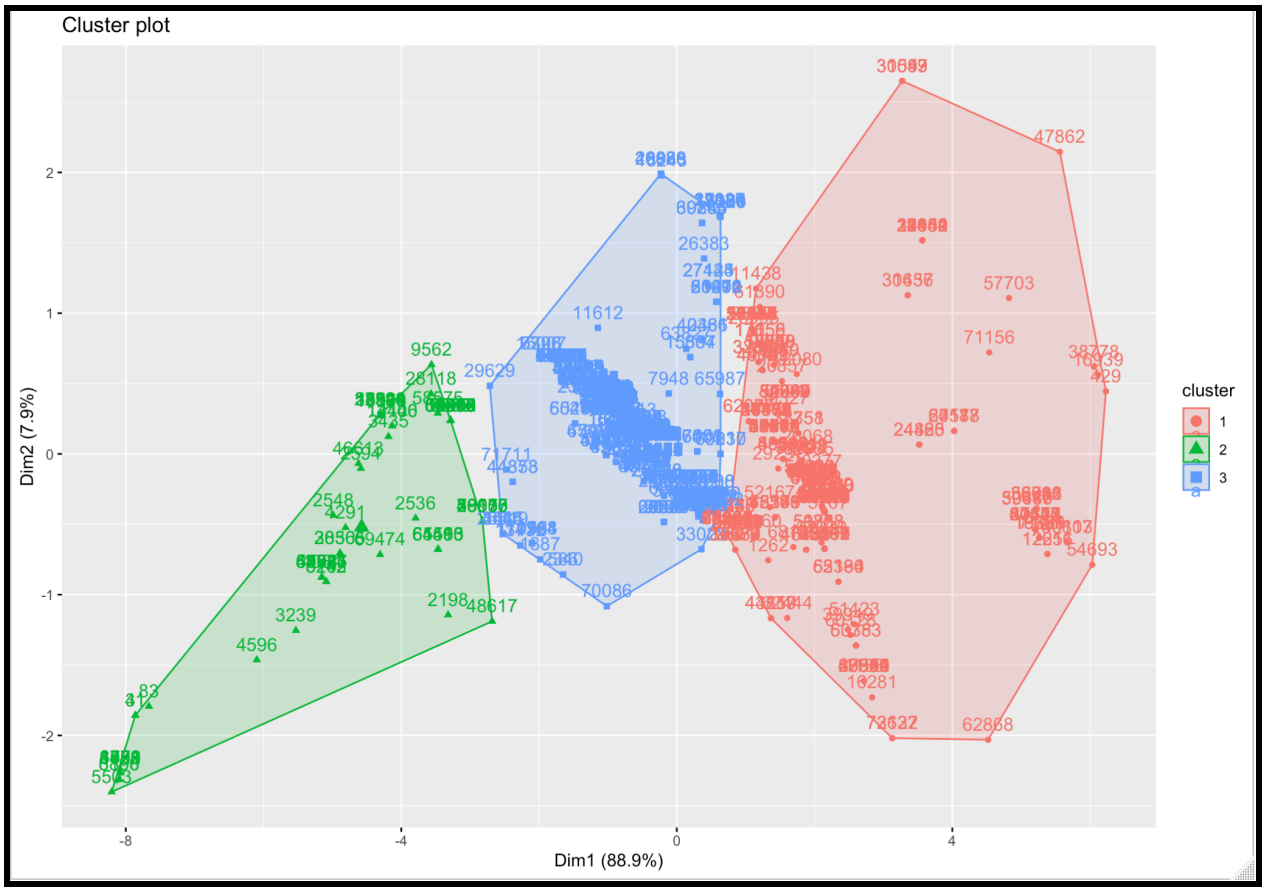
These findings emphasize how crucial it is to properly take model complexity and performance parameters into account when creating neural network topologies for practical uses. It emphasizes how important it is to choose models carefully, keeping in mind both prediction accuracy and computational economy.

Question 3. Clustering

For this assignment, we leveraged the following three clustering techniques to better visualize and understand the relationships of the data across our dataset: K-means/partitional clustering, hierarchical clustering, and density based clustering. Given the size our original dataset (data on nearly 69,000 wind turbines across the United States), we created a randomized subset of 1,000 wind turbines in order to more easily manipulate the data. For each clustering technique, we include the ideal number of clusters, the number of elements per cluster, a visual plot of the clusters as well as an analysis of the patterns in the clusters.

K-Means Clustering

Best number of clusters	3
Elements per cluster	Cluster 1 = 348 Cluster 2 = 74 Cluster 3 = 578



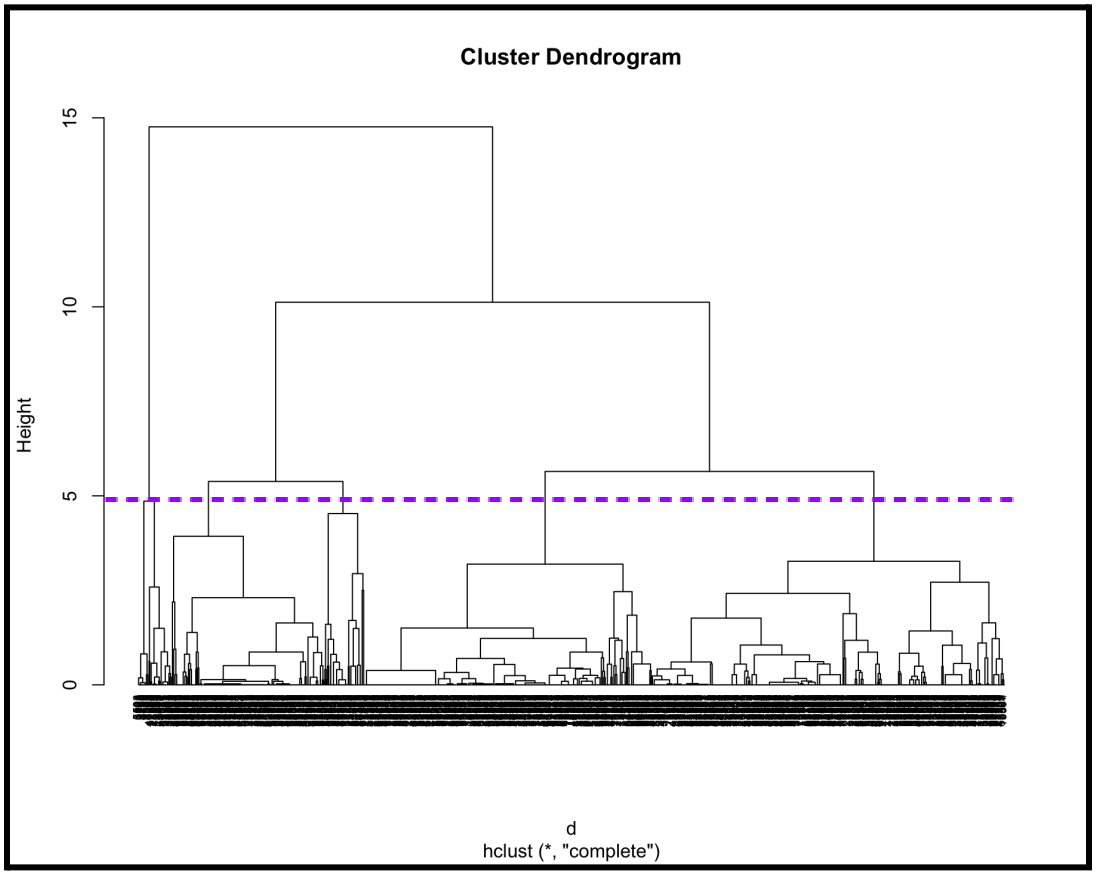
By looking at the summary statistics for the clusters, we look at rotor diameter (t_rd) and rotor swept area (t_rsa) as highly predictive variables from previous analysis. The stats show that

- For Cluster 1, the t_rd values are roughly 1.01sd **above** the mean; the t_rsa values are roughly 1.09sd **above** the mean
- For Cluster 2, the t_rd values are roughly 2.15sd **below** the mean; the t_rsa values are roughly 1.74sd **below** the mean
- For Cluster 3, the t_rd values are roughly 0.34sd **below** the mean; the t_rsa values are roughly 0.43sd **below** the mean

Given that these parameters are measures of size, it can be interpreted that the k-means clustering has clustered the wind turbines into larger (cluster 1), smaller (cluster 2), and average (cluster 3) sized groups.

Hierarchical Clustering

Best number of clusters	5
Elements per cluster	Cluster 1 = 407 Cluster 2 = 39 Cluster 3 = 176 Cluster 4 = 47 Cluster 5 = 331



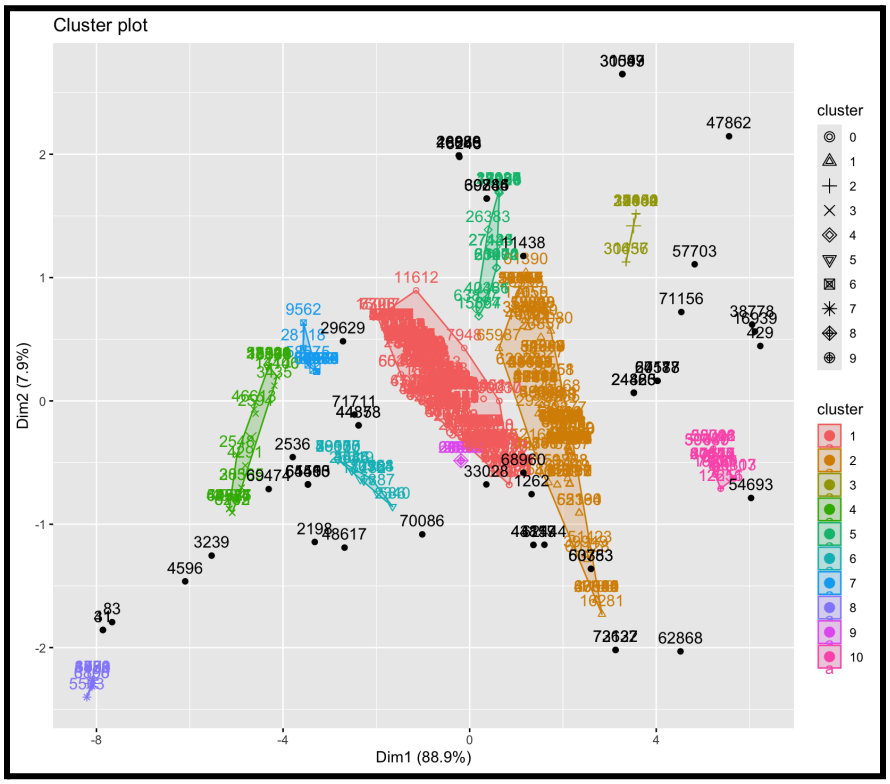
Looking at summary statistics for the dendrogram, it is clear that clusters 1 and 5 have the largest number of turbines, 407 and 331, respectively. The statistics for these 2 clusters show that when compared to the 3 other clusters, their mean values for turbine rotor diameter (t_rd) are small (<+/-1), indicating that the majority of the turbines in these 2 clusters are close to the average size across the dataset. Considering this key variable, the clustering suggests that cluster 2 has the smallest turbines, and cluster 4 has the largest.

Similar to Cluster 4, cluster 3 can also be considered a cluster of larger sized turbines although slightly smaller with t_rd values are roughly 1.097 standard deviations **above** the mean. Additionally, these 2 clusters have the largest turbine capacity values (t_cap) with cluster 3

having a value of 1.083 standard deviations above the mean and cluster 4 having a value of 2.256 standard deviations above the mean. This further suggests that the clustering of turbines is primarily driven by rotor diameter size.

Density-Based Clustering

Best number of clusters	10 with 55 noise points	
Elements per cluster	Cluster 1	548
	Cluster 2	256
	Cluster 3	8
	Cluster 4	24
	Cluster 5	23
	Cluster 6	19
	Cluster 7	23
	Cluster 8	9
	Cluster 9	12
	Cluster 10	23



By generating a k-nearest neighbor distance plot, we determined that the optimal epsilon value was 0.5 and the minimum number of points per cluster should be 8. This resulted in 10 clusters and 55 noise points. Cluster 1 and cluster 2 have the highest concentration of data points with 548 and 256 respectively. When analyzing the summary statistics for those clusters, you see that the average standard deviations for the variables in cluster 1 are all pretty small, indicating that they are very close to the mean and very densely clustered. This corroborates earlier findings in the k-means testing suggesting that the majority of the turbines are average size.

When looking at the statistics for cluster 2, you see that all of the values for turbine rotor diameters (t_rd) are positive, indicating that all of the turbines in this cluster have rotor diameters that are relatively larger compared to the rest of the dataset.

The other clusters are comparatively pretty small with cluster sizes ranging from 8 -24, which indicates that outside of the dimensions of clusters 1 and 2, the rest of the turbines in the dataset are pretty spread out.

Question 4. Comparative Analysis

In addressing Question 4, we embark on a comparative analysis focusing on a classification task derived from one of our research question. Since our dependent variable, t_cap, is continuous, we first segment it into classes by identifying quantiles and assigning labels to each row accordingly. Table 4 is the percentage table for each class.

Table 4: Percentage table for each class

Class 1	Class 2	Class 3	Class 4
%0.2804926	%0.2782008	%0.2268831	%0.2144235

Subsequently, we construct models using Linear Support Vector Machine (SVM), Random Forest, and Neural Network algorithms.To conduct this comparative analysis, we leverage the CARET package in R, employing three different cross-validation techniques: k-fold cross-validation, repeated k-fold cross-validation, and bootstrapping. Each model's performance is evaluated across these techniques, and the results are meticulously examined and discussed below.

K-Fold Cross Validation

For this analysis, we evaluated the linear SVM models from Question 1 and the neural network model from Q2, alongside another neural network from Milestone 2. Employing the k-fold cross-validation method with 3 folds, we compared their performance. Results demonstrate that among these models, random forest exhibits higher mean accuracy and kappa values. Notably, while SVM and neural network models display similar accuracy levels, a significant disparity exists in their kappa values, indicating inferior performance of neural networks compared to SVM. Refer to Table 5 for detailed results, while Figures 4 and 5 depict box plots and dot plots of accuracy and kappa values.

Table 5: K-Fold Cross validation

Accuracy						
Model	Min	1st Qu.	Median	Mean	3rd Qu.	Max
RF	0.8974850	0.8983095	0.8991341	0.8986119	0.8991754	0.8992167
SVM	0.7967887	0.7978112	0.7988338	0.7982826	0.7990296	0.7992254
NN	0.7482378	0.7482378	0.7530353	0.7646898	0.7729158	0.7879988
Kappa						
RF	0.8626336	0.8637499	0.8648661	0.8641509	0.8649095	0.8649529
SVM	0.7274116	0.7288277	0.7302438	0.7294577	0.7304808	0.7307178
NN	0.6612438	0.6677596	0.6742753	0.6834968	0.6946232	0.7149712

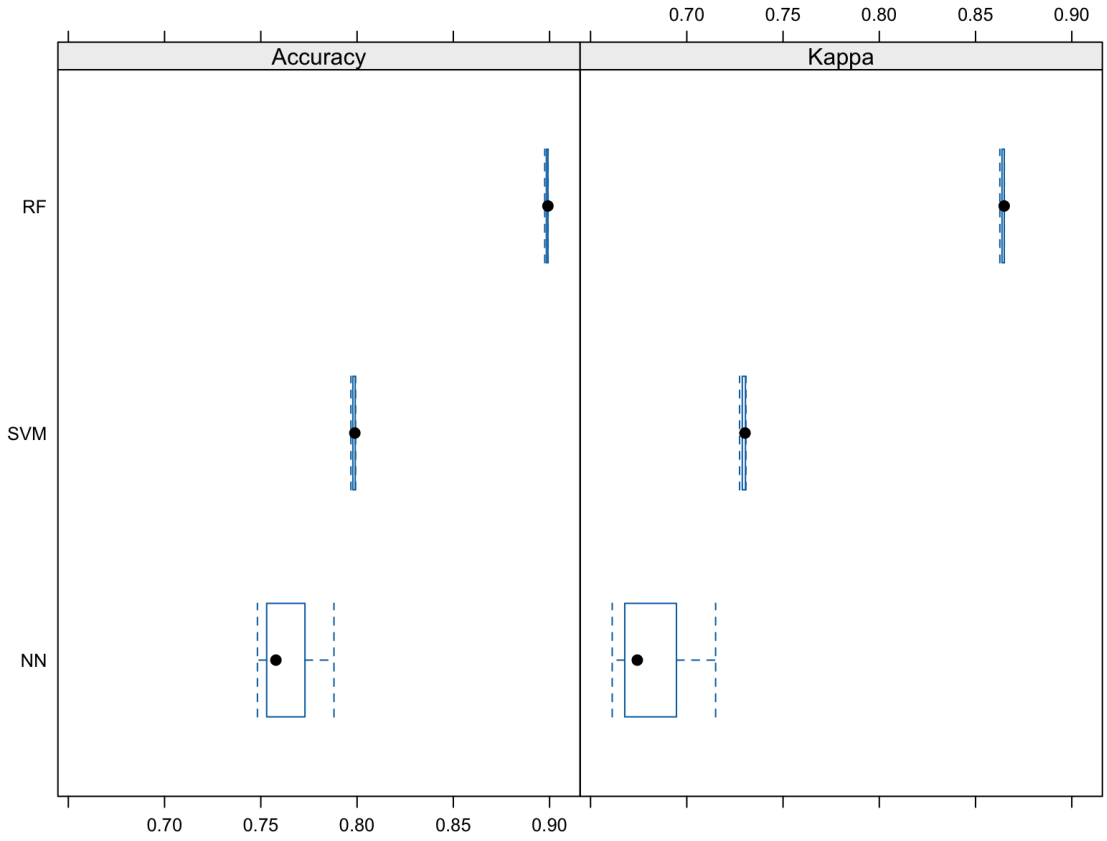


Figure 4: Box Plot of K-Fold Cross validation

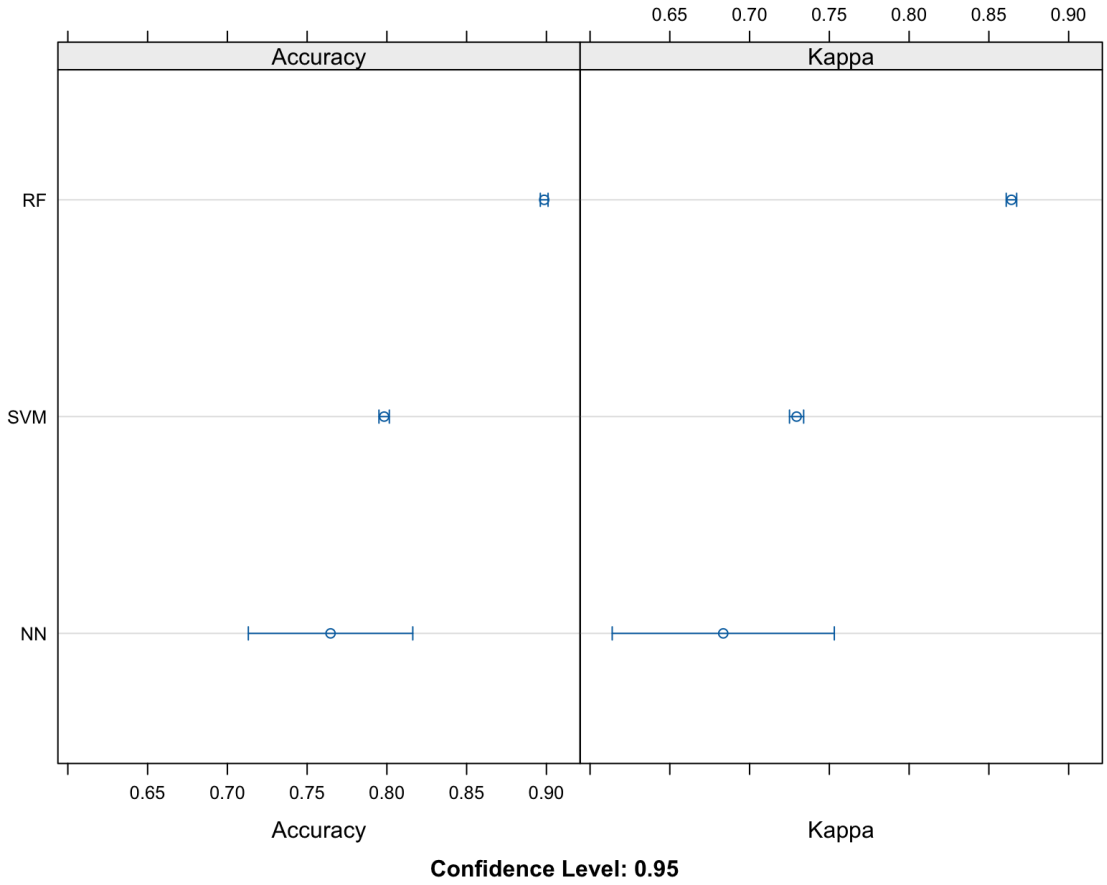


Figure 5: Dot Plot of K-Fold Cross validation

Repeated K-Fold Cross Validation

In this section, we utilize repeated k-fold cross-validation, a rigorous evaluation technique involving the subdivision of the dataset into k folds. With each fold acting as both training and testing data, this process is iterated multiple times, as specified by the number of repeats, ensuring comprehensive model assessment. Here, we employ 3 folds with 3 repeats, allowing for thorough examination of model performance and enhancing the reliability and robustness of our findings.

As depicted in Table 6, the random forest model demonstrates superior accuracy and kappa values compared to SVM and NN. While SVM and NN exhibit similar accuracy rates, a significant discrepancy is observed in their kappa values. Figures 6 and 7 depict box plots and dot plots of accuracy and kappa values.

Table 6: Repeated K-Fold Cross validation

Accuracy						
Model	Min	1st Qu.	Median	Mean	3rd Qu.	Max
RF	0.8938642	0.8968233	0.8987860	0.8983314	0.8996127	0.9017449
SVM	0.7964405	0.7982594	0.7993125	0.7990369	0.8000522	0.8009225
NN	0.7170829	0.7403072	0.7458881	0.7478741	0.7598346	0.7766416
Kappa						
RF	0.8577855	0.8617821	0.8643714	0.8637823	0.8654782	0.8683812
SVM	0.7269952	0.7294648	0.7308398	0.7304769	0.7318032	0.7329756
NN	0.6204685	0.6513368	0.6570059	0.6609114	0.6765706	0.6997583

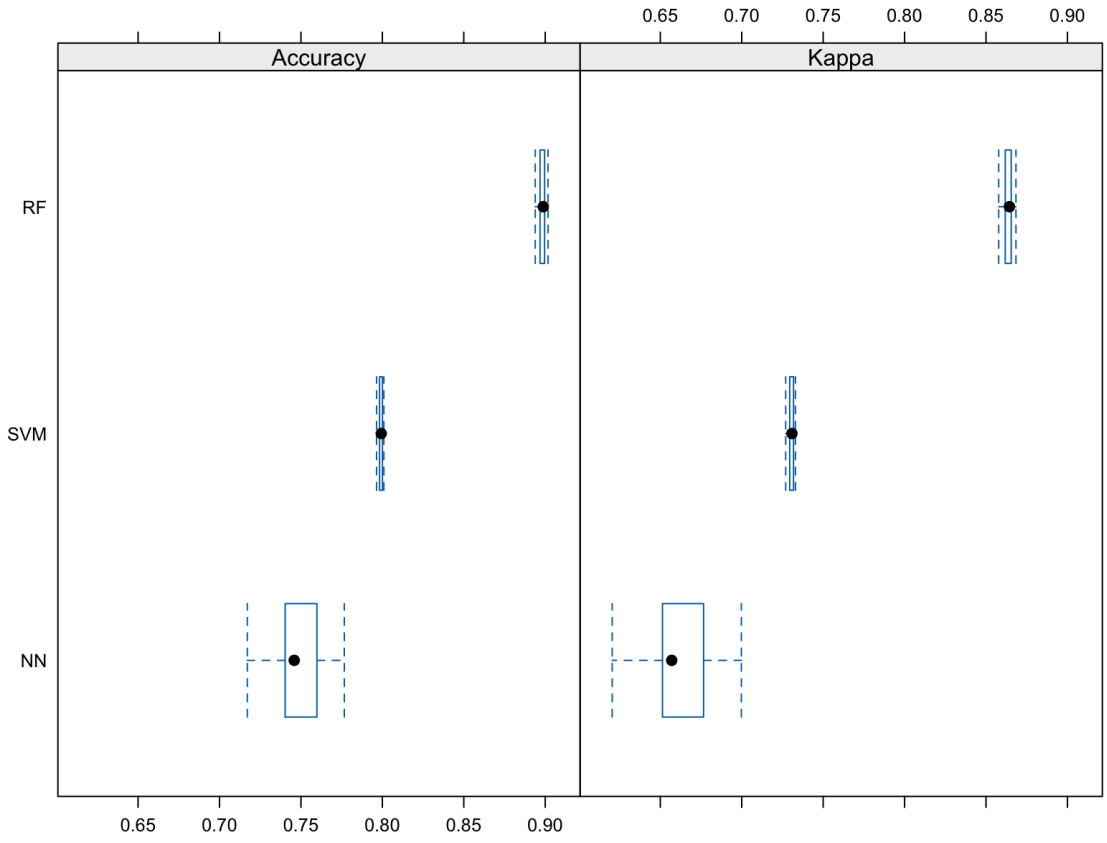


Figure 6: Box Plot of Repeated K-Fold Cross validation

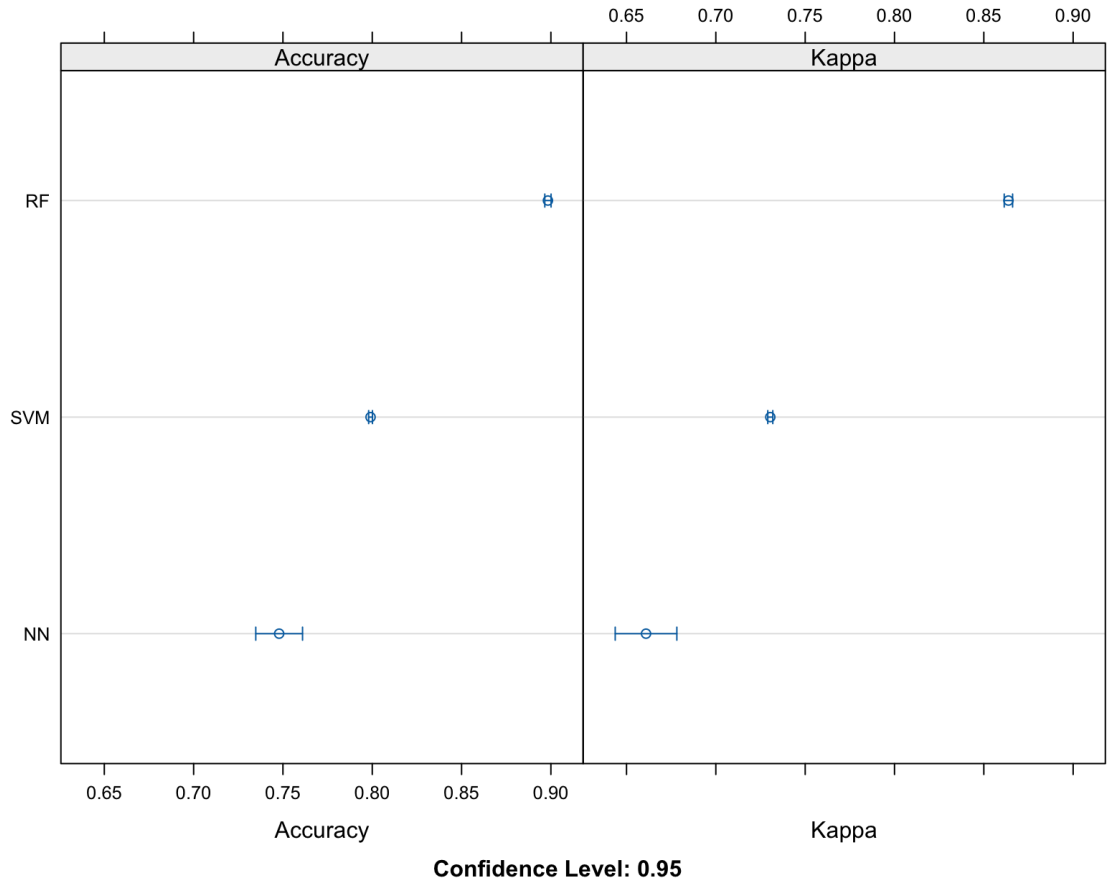


Figure 7: Dot Plot of Repeated K-Fold Cross validation

Bootstrapping

Bootstrapping reveals higher accuracy and kappa values across models compared to k-fold and repeated k-fold cross-validation methods, indicating improved model performance assessment through bootstrapping. Among the models, Random Forest (RF) maintains the highest accuracy and kappa, followed by SVM, while NN exhibits relatively lower metrics. Table 7 shows the result.Figures 8 and 9 depict box plots and dot plots of accuracy and kappa values.

Table 7: Bootstrapping

Accuracy						
Model	Min	1st Qu.	Median	Mean	3rd Qu.	Max
RF	0.8987122	0.8991944	0.8996766	0.9003221	0.9011270	0.9025774
SVM	0.8015957	0.8022285	0.8028612	0.8033204	0.8041827	0.8055042
NN	0.7176638	0.7544732	0.7912827	0.7747257	0.8032567	0.8152307
Kappa						
RF	0.8642817	0.8649075	0.8655334	0.8664201	0.8674893	0.8694453
SVM	0.7338845	0.7347555	0.7356264	0.7362184	0.7373853	0.7391442
NN	0.6201344	0.6706593	0.7211842	0.6977636	0.7365782	0.7519723

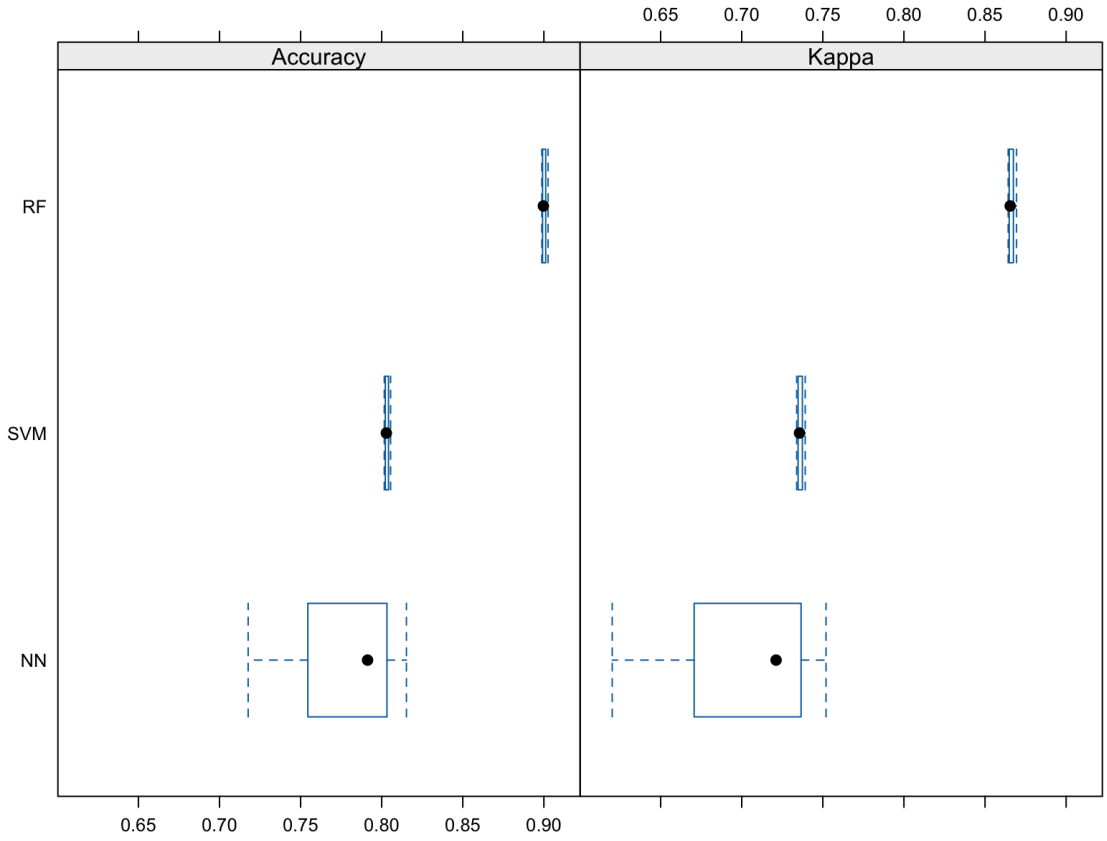


Figure 8: Box Plot of Bootstrapping

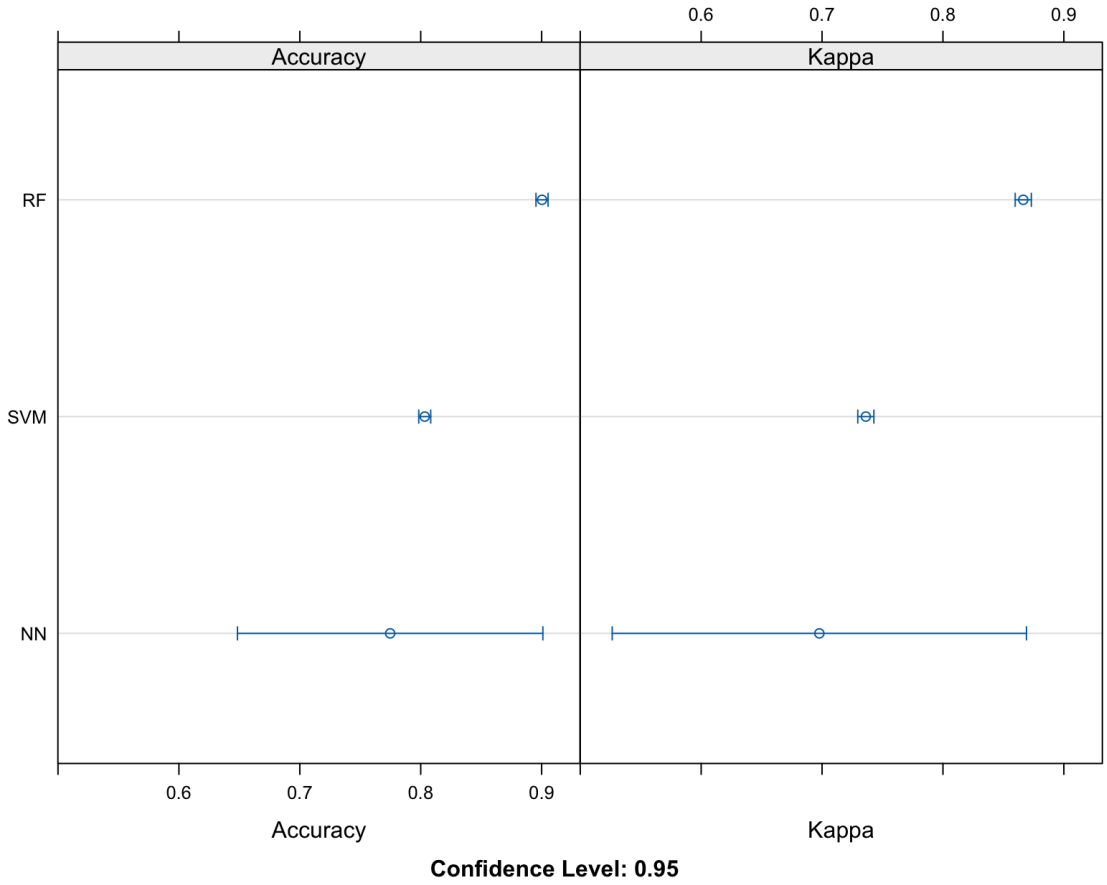


Figure 9: Dot Plot of Bootstrapping

Question 5. Feature Selection

In order to improve the predictive performance of our models, we applied three different feature selection techniques (filtering, lasso regularization, and wrapper method) on previously generated models from previous milestones. Differences between the original and modified models are described for each below:

1. Linear Regression - Filtering Technique

Original R^2 error	0.8048
Filtered R^2 error	0.8044

Upon analyzing the correlation of independent variables in the linear regression model, it was determined that the hub height variable (t_hh) had the least impact on the dependent variable, turbine capacity. Using the filter technique, we re-ran the model excluding this variable and found that the R^2 value decreased slightly from 0.8048 in the original model to 0.8044 in the filtered model. Given the minimal change, it can be assumed that the hub height is not an important factor when considering the variance of turbine capacity.

2. Logistic Regression - Lasso Regularization

Original AIC	84.087
Regularized AIC	0.103126

Using the log regression model that was built for milestone 2, we applied a lasso regularization to improve the prediction accuracy of our model. To evaluate this, we looked at the Akaike Information Criterion (AIC) of both models, which is a measure of the fit of a statistical model. The original model had an AIC of 84.087 and once we applied the lasso regularization, the model had an AIC of 0.103126. This dramatic decrease suggests that the regularization method significantly improved the fit and accuracy of the log regression model.

3. Naive Bayes - Wrapper Method

Original accuracy rate	99.8%
Wrapper accuracy rate	98.58%

Using the Naive Bayes model that was built for milestone 2, we applied the forward selection wrapper method to improve the accuracy of the model. In the original model, the accuracy rate was pretty high at 99.8% accurate true negative and true positive predictions. Upon applying the wrapper and rerunning the model, the accuracy rate slightly decreased to 98.58%. This decrease in predictive performance suggests that the wrapper might have introduced some level of complexity or noise into the model causing overfitting.

Question 6. Ethical Issues

Across the three milestone assignments, our analyses have leveraged the United States Wind Turbine Database (USWTDB), a large publicly available repository holding significant data on both onshore and offshore wind turbines in the United States. This repository is the joint effort of several organizations including the United States Geological Society, American Wind Energy Association, Lawrence Berkeley National Laboratory, and the American Clean Power Association. In addition to the turbine parameters and dimensions data that we have focused on in our assignments, the dataset also includes a host of other information including average retail price, net generation, federal identification codes, and manufacturing information. Given the actors involved in developing this repository, there are two main areas of potential ethical concern to consider:

1. Conflicts of Interest - In this partnership of organizations, there are industry associations like the American Wind Energy Association that have contributed to the collection of data. These organizations may have vested interests in promoting wind energy development and could potentially influence the presentation or interpretation of data to favor their agendas. Potentially compromised data could result in inaccurate models and data interpretations. This is particularly relevant in today's political climate where environmental data can be heavily politicized and impact significant policy decisions. As we consider data science questions that evaluate predictive performance based on this data, it becomes very important to consider how data are sourced and the potential biases in those sources.

2. Environmental Impact - Although the dataset does not include any information that is directly connected to individuals, it does include longitudinal and latitudinal data of wind turbines across the United States. Ethical considerations become relevant when you consider that wind turbines have been reported to increase noise pollution and marginalized communities, including low-income neighborhoods and communities of color, often face a disproportionate burden of environmental pollution due to wind farms. This can lead to environmental injustice, where vulnerable populations are unfairly exposed to the negative impacts of wind turbines. With this in mind, it becomes vital to ensure that any analyses conducted using this dataset take into account the experiences of marginalized communities in regards to how findings may be interpreted and used to inform policy decisions.
3. Data privacy - Although the USWTDB offers a plethora of data that is vital for wind energy research and policy development, it is important to recognize that there may be privacy issues. The file might include wind turbine location data, which could obliquely disclose private information about neighboring towns, infrastructure, or even individual properties. Despite the dataset's macro-level data aggregation goal, there is a chance that private information may be inadvertently identified or made public. To reduce the

possibility of data misuse or privacy rights violations, data handling methods must place a high priority on anonymization and compliance with privacy laws. In order to safeguard the interests of all parties concerned, this makes sure that ethical standards of data privacy are respected while utilizing the dataset's insights.

Contributions

The contributions of each team member were evenly distributed among their respective skill sets and knowledge.

● Code:

○ READ_ME file: Nour

○ R file: All students contributed equally to the code portion of this project.

● Presentation: All students contributed equally to the preparation and recording of the presentation.

● Report:

- Introduction: Zahra
- SVMs: Zahra
- Neural Networks: Nour
- Clustering: Kuru
- Comparative Analysis: Zahra
- Feature Selection: Kuru
- Ethical Issues: Kuru &Nour