Turbine Wind: Analysis

1. Problem statement

Wind energy is a crucial source of renewable energy that has seen rapid growth in recent years. However, the performance, efficiency, and reliability of wind turbines are often affected by variable wind conditions. To address this issue and optimize the operation of wind turbines, it is essential to carries out big data analysis and power prediction. In this way, insights of optimal operating conditions can be found to increase the power productivity of this equipment.

2. Solution approach

The development of data analysis and power prediction model based on wind features was performed using the following steps:

a) Real-data collection.
b) Data cleaning and formatting.
c) Data analysis
d) Machine learning development.

The complete project code can be accessed here.

3. Stakeholders and Benefits

Big data analysis and power prediction of wind turbines offer various benefits to stakeholders, contributing to efficient wind energy production and a sustainable future. Key benefits for each stakeholder group include:

a) *Wind farm owners/operators*: Improved decision-making for maintenance, grid management, and energy production. Increased efficiency and reduced operational costs.
b) *Energy consumers*: Stable and reliable energy supply. Lower energy prices due to improved efficiency.
c) *Government/regulatory agencies*: Achievement of renewable energy targets. Progress towards greenhouse gas emission reduction goals.
d) *Turbine manufacturers/technology providers*: Enhanced wind turbine technologies driven by data analysis. Competitive advantage in the market.
e) *Researchers/academic institutions*: Opportunities for research, innovation, and development of new models/algorithms. Collaboration with industry stakeholders for practical applications.

4. Data

The dataset was obtained from Kaggle Repository (Turkey Wind Turbine Data). In Wind Turbines, Scada Systems measure and save data. The dataset presents the following attributes (50530 instances):

a) *Date/Time* (for 10 minutes intervals).
b) *LV ActivePower (kW)*: The power generated by the turbine for that moment.
c) *Wind Speed (m/s)*: The wind speed at the hub height of the turbine (the wind speed that turbine use for electricity generation).

d) *TheoreticalPowerCurve (kWh)*: The theoretical power values that the turbine generates with that wind speed which is given by the turbine manufacturer
e) *Wind Direction (°)*: The wind direction at the hub height of the turbine (wind turbines turn to this direction automatically).

5. Tools

- Spark, Data visualization, Data cleaning, Machine-Learning, Python

6. Results

The data cleaning and processing procedure was performed using Pyspark library. PySpark is the Python library for Apache Spark, an open-source, distributed computing system designed for big data processing and analytics. PySpark allows data scientists and developers to process large volumes of data in parallel across a cluster of computers. Some key features of PySpark include:

a. *Resilient Distributed Datasets (RDDs)*: RDDs are a fundamental data structure in Spark, allowing fault-tolerant parallel processing of data. They can be created from data stored in Hadoop Distributed File System (HDFS).
b. *DataFrames and Datasets*: PySpark provides higher-level abstractions called DataFrames and Datasets that offer more optimized and convenient ways to work with structured data.
c. *MLlib*: PySpark includes a machine learning library called MLlib, which provides various algorithms and tools for building scalable machine learning models.
d. *GraphX*: PySpark also includes a graph processing library called GraphX, allowing users to perform graph computations and analytics.
e. *Streaming*: PySpark supports real-time data stream processing through its built-in streaming capabilities, allowing users to process and analyze live data.

6.1. Data Distribution

The time series was decomposed into months, hours and days. The data summary was performed using Pandas Autoprofiling. The six variables are numeric, with no missing or duplicated values. Fig 1 shows the data distribution of the multiple features.

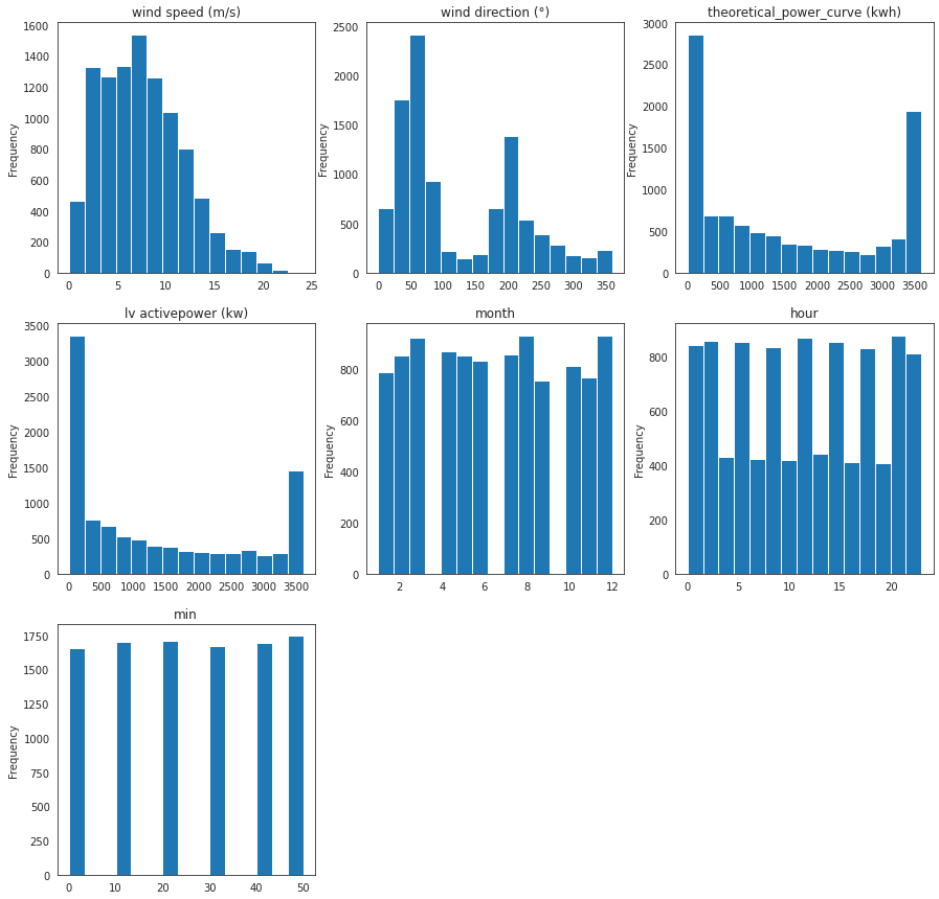

Fig 1. Data distribution using histograms.

As shown in Fig 1, the following findings can be observed:

a. *Wind speed*: the data distribution is skewed right, also known as a positively skewed distribution. The mean (average) is greater than the median and mode in a positively skewed distribution. This is because the few large values on the right side of the distribution significantly impact the mean, pulling it to the right. Zero wind velocity values are also presented in the dataset. On the other side, wind velocities higher than 20 m/s are scarce.
b. *Wind direction*: it shows a bimodal distribution (data distribution with two distinct peaks or modes), indicating two preferential wind directions (70 and 210°). The 70° peak is narrower than another one, suggesting a more concentrated distribution around this value. The distance between the two peaks is relatively large, indicating that wind direction groups are not overlapping.
c. *Theoretical power curve*: This feature shows excessive outliers and missing values. A null theoretical power curve for modelling purposes has few incidences. Excluding the outliers, the data distribution is approximated to a uniform distribution.
d. *Lv active power*: It shows a similar data distribution to the theoretical power curve, indicating that no energy is generated under a specific range of wind conditions (speed and direction). In this way, theoretical and active power curves are strongly correlated.
e. *Month/min*: These features exhibit uniform distribution.
f. *Hour*: It shows the stratified uniform distribution. Most hour registers are at the beginning or the end of the day. During the day, the amount of hour registers collected oscillates between two strata.

6.2. Feature average by month

The wind characteristic of a place varies with month, season, and weather conditions, which directly affect power generation. Fig 2 shows the power generation and wind conditions average by month.

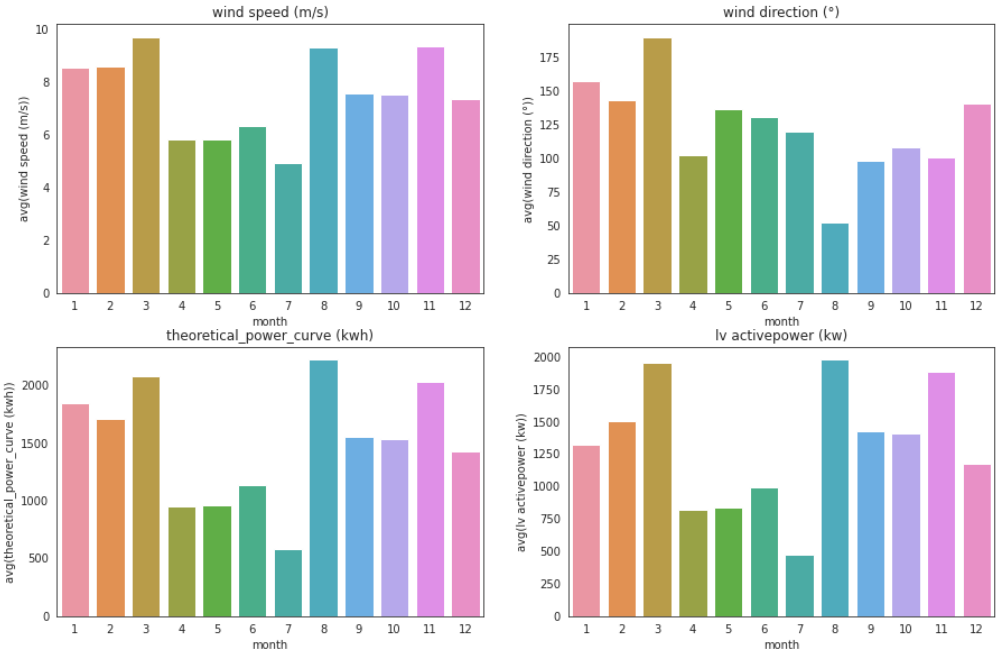

Fig 2. Average features by month.

March, August and November are the top three months with significant active power production. Theorical and active power has a very similar distribution. Conversely, the warm months in Turkey (from April to July) present lower wind speed averages. In August, the average wind speed is high. Furthermore, the months with the highest wind speed are the same with the most increased power production. This observation is not observed for wind direction, which changes over the year and the top months with power generation (March: 180°, August: 50°, and November: 100°). This behavior indicates that wind speed correlates to power generation more than wind direction.

6.3. Feature average by hour

The wind conditions, as well as the power generation, varies over the day. For this reason, analyzing the features on an hourly is relevant. Fig 3 shows the average feature values by the hour.

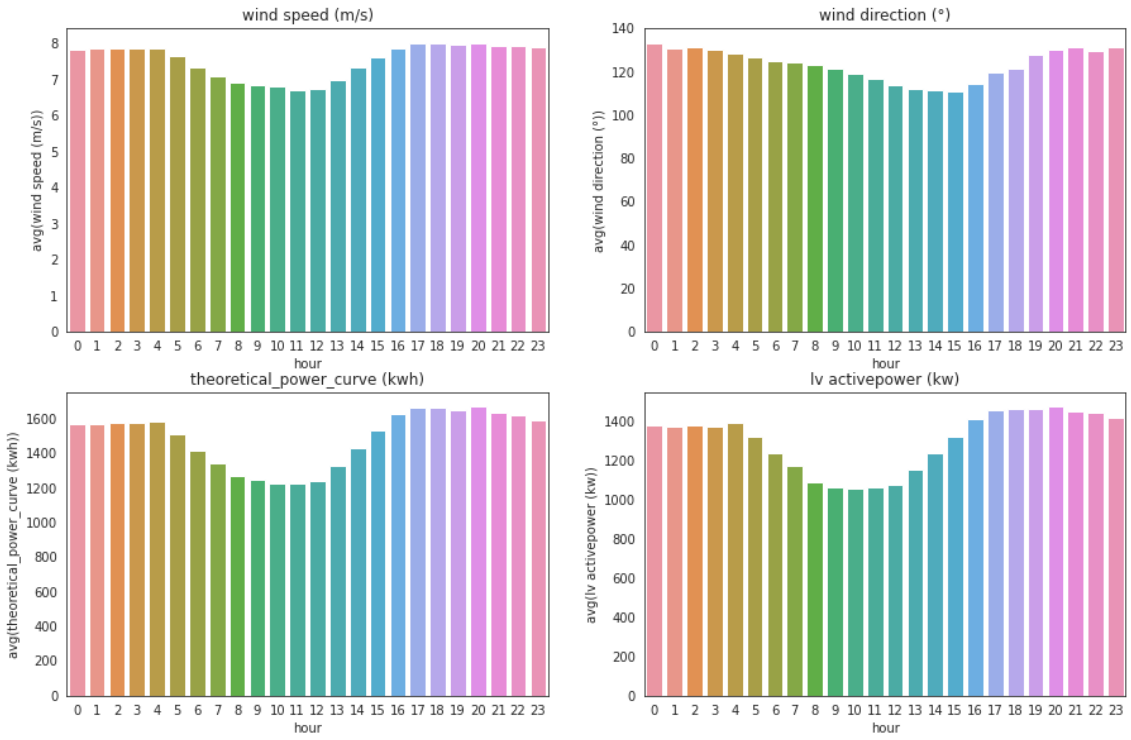

Fig 3. Feature average values by hour.

As stated above, the wind speed is higher during nighttime (from 6 p.m. to 4 a.m.), around 8 m/s. Between 9 a.m. and 12 p.m., the average wind speed decreases to around 6.5 m/s. In concordance, the highest power generation is observed during the last part of the day (about 1400 kw), when the wind velocity is more elevated. On the other side, power generation is minimum in the morning and the noon.

6.4. Correlation heatmap

A correlation heatmap is a graphical representation of the relationships between variables in a dataset. It is often used to visualize the strength and direction of these relationships, which can help identify patterns and potential associations among the variables. One common method for calculating correlation is the Pearson correlation coefficient.

The Pearson correlation coefficient (also known as Pearson's r) is a measure of the linear relationship between two continuous variables. It ranges from -1 to 1, where:

a. A value of -1 indicates a perfect negative linear relationship (when one variable increases, the other decreases in a linear fashion).
b. A value of 0 indicates no linear relationship between the variables.
c. A value of 1 indicates a perfect positive linear relationship (when one variable increases, the other also increases in a linear fashion).

It is important to note that correlation does not imply causation – a strong correlation between two variables does not necessarily mean that one causes the other. Fig 4 shows the correlation heatmap by Pearson coefficient.

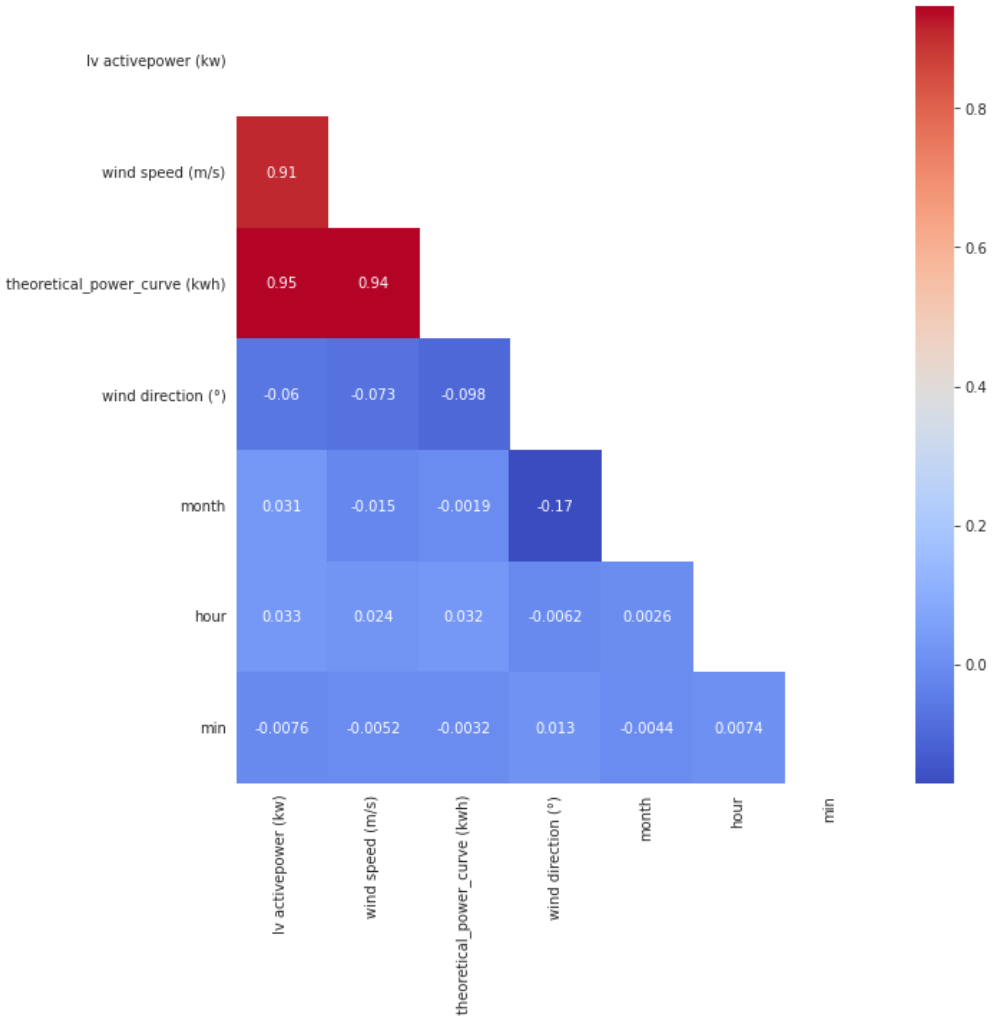

Fig 4. Correlation heatmap by Pearson coefficient.

As shown in Fig 4, theoretical and active power generated is highly correlated, as expected. On the other hand, active power generation-wind speed has a strong correlation (R = 0.91) with wind speed, as inferred previously. The power generation is increased with the wind speed, as expected—furthermore, negligible Pearson correlation between power generation and wind direction and time variables.

No multicollinearity is observed. Multicollinearity is when two or more independent variables in a regression model are highly correlated. This high correlation can cause problems in estimating the individual effects of each variable and lead to unstable model coefficients. A common rule of thumb is that a correlation coefficient greater than 0.8 (or less than -0.8) indicates a high correlation between variables. The highest correlation was between wind direction and month (-0.17).

6.5. Power Production vs wind features

A first overview of power production versus wind features was obtained using a pair plot (grid of scatter plots where each variable in a dataset is plotted against every other variable). The correlation heatmap shows the scatterplot of power generation against wind speed in Fig 5.

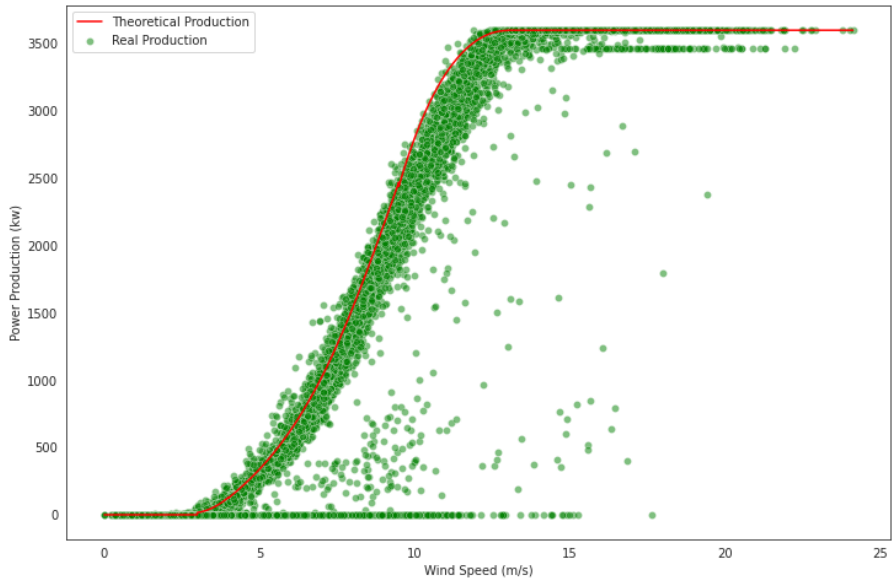

Fig 5. Power production (real and theoretical) versus wind speed.

From the previous figure, the following observations can be made:

a. The theoretical power production satisfactory reproduces the real power production behavior.
b. The power production shows an asymptotic behavior (3500 kW for higher wind speed value than 13 m/s, approximately.
c. Zero values of real power production occur in a broad range of wind speeds. This observation could be attributed to the fact that the data collection continued when the wind turbine was stopped, probably during routine maintenance, being an anomaly for modelling purposes.

The zero values distribution in power generation was analyzed, and the results plotted in Fig 6.

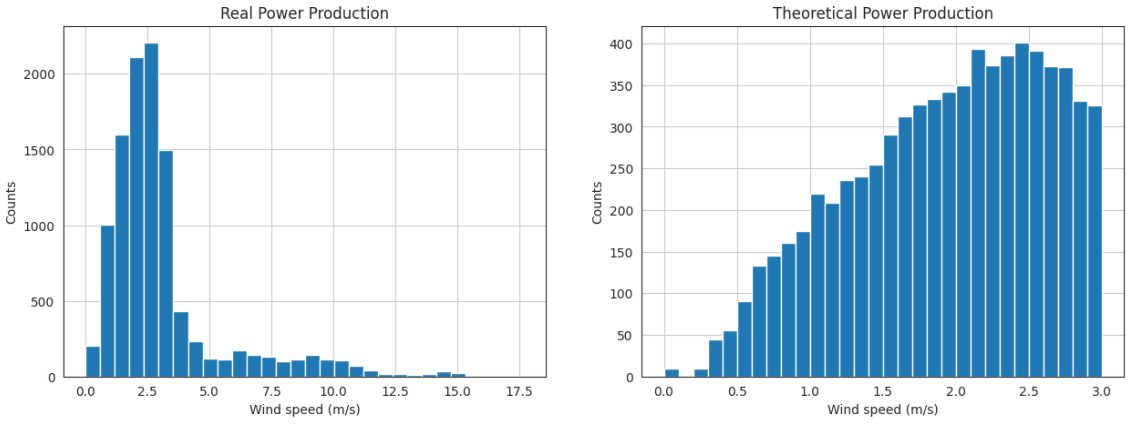

Fig 6. Zero values distribution in power generation versus wind speed.

Most zero values for real power production occur at a wind speed from 0.5 to 4 m/s (long-tail). On the other hand, for zero theoretical power production, the maximum value observed was 3.0 m/s, which represents the minimum speed required to generate theoretical power production. The majority of zero values in real production corresponds to velocity lower than the cited minimum, pointing out that can be attributed that the wind velocity was not spin the turbine blades for power production.

Fig 7 exhibits zero values of power production (real and theoretical) against months. August is the month with lower numbers of zero values. Meanwhile, December and April are the months with hither zero values (around 1550). Similar observations were made for theoretical power generation. In this way, most null values can be attributed to wind speed lower than 3 m/s. Fig 8 plots the amount of wind speed lower than 3 m/s for each month. Fig 7 and 8 similarity supports the previous finding.

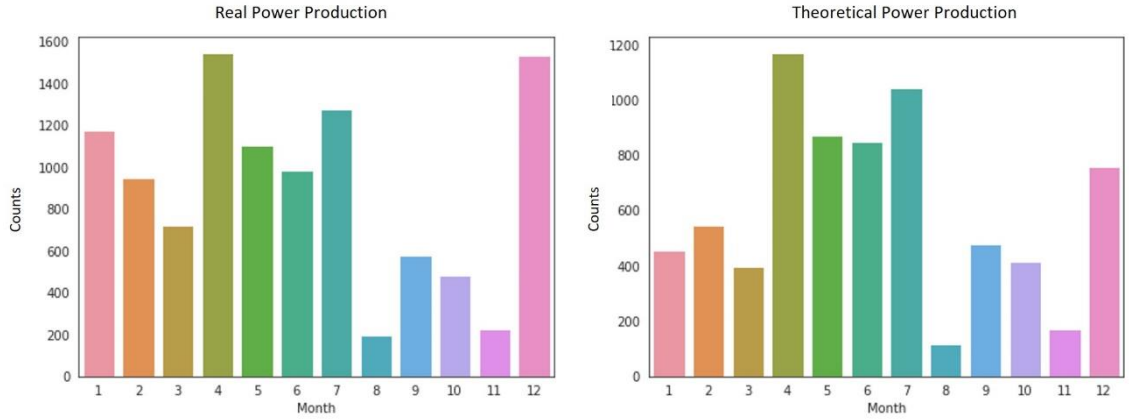

Fig 7. Zero values of power production versus month.

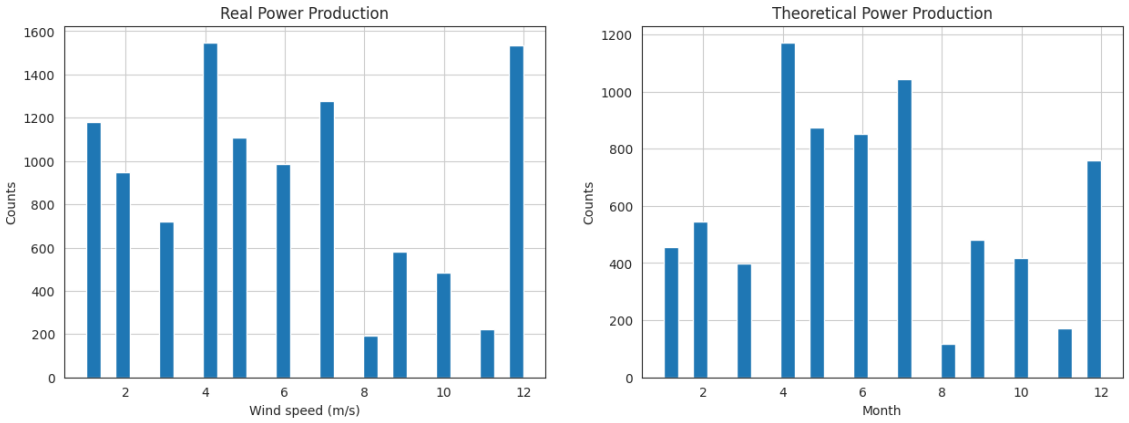

Fig 8. Wind speed lower than 3 m/s versus month.

6.6. Outlier handling

Zero values for power production were removed, considering the available data. Outlier visualization with a violin plot is used in data analysis to graphically represent the distribution of data points, including potential outliers. A violin plot is created by combining aspects of a box plot and a kernel density plot, displaying the data's spread, central tendency, and shape. Fig 9 shows the violin plot for the four main features for each month.

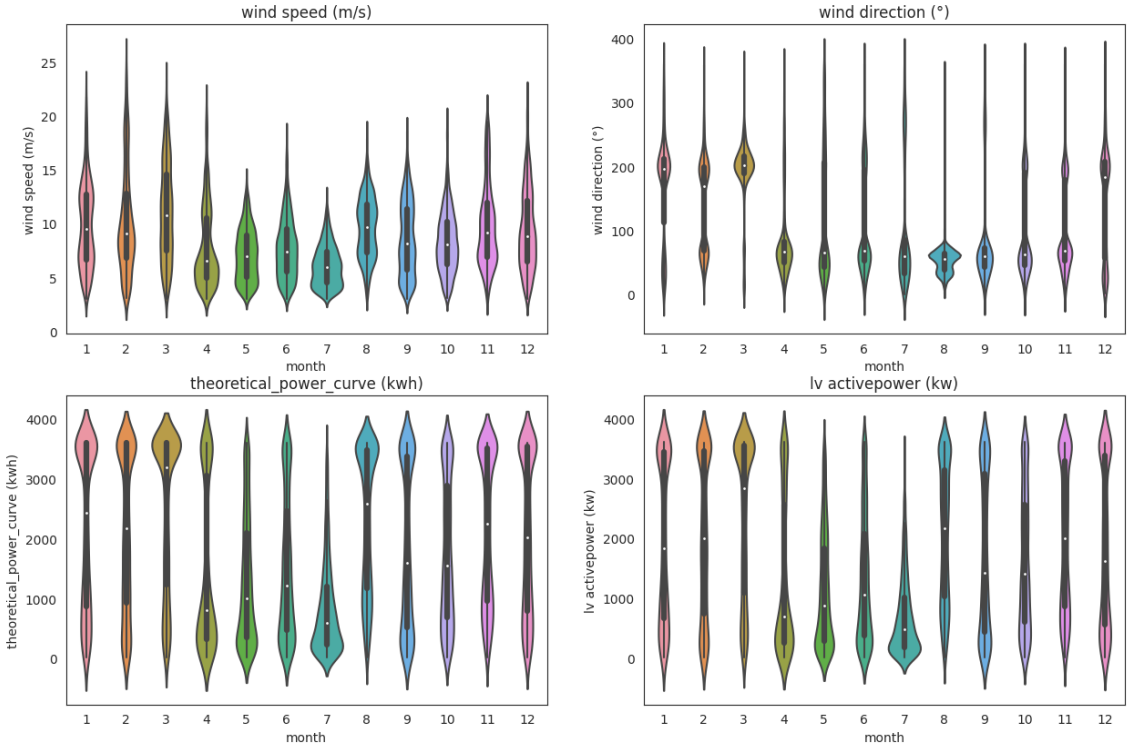

Fig 9. Violin plot for outlier visualization.

As shown in Fig 9, all features present significant outliers over the year, especially wind speed and direction. Power production exhibited the highest number of outliers in July. The outliers can infer anomalies during the model training. For this reason, outliers were removed based on quantile criteria (wind speed: 3- 18.8 m/s). Fig 10 presents the real power generation based on wind conditions in polar diagrams. Most records (after outlier remotion) give a predominant wind direction between 170-225°, and 15-90°, showing the highest wind speed values.

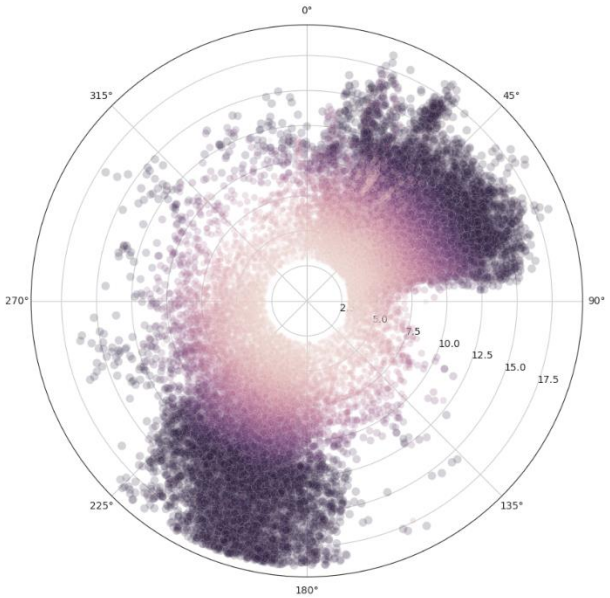

Fig 10. Polar representation of real power generation versus wind conditions.

6.7. Data Modeling

6.7.1. Data Processing

The data processing was performed using Apache Spark MLlib, which It provides various tools and techniques for preparing and processing data for machine learning tasks. One such technique is the vectorization of features, which is crucial for transforming raw data into a suitable format for machine learning algorithms. The vectorization of features is an essential step in the machine learning pipeline:

a. *Dense and Sparse Vectors*: Dense vectors store all the values in an array, whereas Sparse vectors store only the non-zero values along with their indices, making them more memory-efficient for datasets with a large number of features and lots of zeros.
b. *Feature Transformers*: Some common transformers include Tokenizer (for text data), OneHotEncoder (for categorical data), and StandardScaler (for numerical data). These transformers can be applied sequentially in a pipeline to prepare the data for machine learning models.
c. *Feature Selection*: Spark MLlib also supports feature selection techniques like Chi-Squared selector and Variance Threshold selector.
d. *Pipeline*: Spark MLlib's Pipeline API allows you to chain multiple feature transformers and a machine learning model into a single, unified workflow.

The target label is real power production, with the following features: month, hour, and wind speed and direction. The dataset was split into train dataset (80%) and test dataset (20%).

6.7.2. Models

The following models were tested: Gradient-Boosted Trees (GBTs), Generalized Linear Regression, Decision Tree Regressor, Random Forest Regressor, Linear Regression, FM Regressor, and Isotonic Regression.

a. *Gradient-Boosted Trees (GBTs)*: Gradient-Boosted Trees is an ensemble learning technique. It works by iteratively training trees to correct the errors made by previous trees. The final model is the weighted sum of these trees. GBTs are particularly effective for handling non-linear relationships and can be used for both regression and classification tasks. They are known for their high performance and robustness but can be prone to overfitting if not properly tuned.
b. *Generalized Linear Regression*: It is a generalization of linear regression that allows for response variables that have error distribution models other than a normal distribution. It combines a linear predictor with a link function to model the relationship between the input features and the response variable.
c. *Decision Tree Regressor*: A Decision Tree Regressor is a tree-based model used for regression tasks. It recursively splits the input space into different regions based on the values of the input features. Each split is determined by choosing the feature and split point that minimize the overall error in predicting the target variable.
d. *Random Forest Regressor*: Random Forest Regressor is an ensemble learning method that combines multiple decision trees to create a more robust model. The trees in the forest are trained independently, and their predictions are aggregated (typically by averaging) to form the final prediction. Random Forests introduce randomness in the tree construction process to reduce correlation between the trees, which helps improve

performance and reduce overfitting. They can handle non-linear relationships and are generally more accurate than individual decision trees.

e. *Linear Regression*: It is a simple and widely used statistical method for modeling the relationship between a response variable and one or more input features. It assumes a linear relationship between the input features and the target variable and estimates the coefficients of the linear equation by minimizing the sum of squared errors between the predicted and actual values.
f. *FM Regressor*: Factorization Machines (FM) Regressor is a versatile and efficient model that can handle sparse data and high-dimensional feature spaces. It works by factorizing the feature interactions, allowing it to capture the interactions between the input features even when data is sparse. It can handle both linear and non-linear relationships and is particularly useful when dealing with categorical data with high cardinality.
g. *Isotonic Regression*: Isotonic Regression is a non-parametric technique used to fit a monotonic function to the data. It aims to find the best non-decreasing (or non-increasing) function that minimizes the sum of squared errors between the predicted and actual values. Isotonic Regression is useful when there is a natural ordering in the input data, and the relationship between the input and output variables is expected to be monotonic. It can handle non-linear relationships but is limited to monotonic functions.

The performance metrics to test the models were correlation coefficient (R^2), MAE (Mean Absolute Error), and RMSE (Root Mean Squared Error). Table 1 shows the metric for each tested model.

Table 1. Metrics of the tested models.

Model	**R^2**	**MAE**	**RMSE**
GBTs	0.9768	95.94	193.6
Generalized Linear Regression	0.8916	282.8	409.7
Decision Tree Regressor	0.9654	119.0	231.3
Random Forest Regressor	0.9421	212.1	299.4
Linear Regression	0.8916	282.8	409.7
FM Regressor	0.6035	617.7	783.6
Isotonic Regression	0.01804	1097	1233

According to the metrics obtained, the best performance was reached using GBTs, with R2 = 0.9768 (the model can represent 97.68% of data behavior). Finally, Fig 11 plots real and theoretical power generation and predicted values using the GBTs model for the entire dataset.

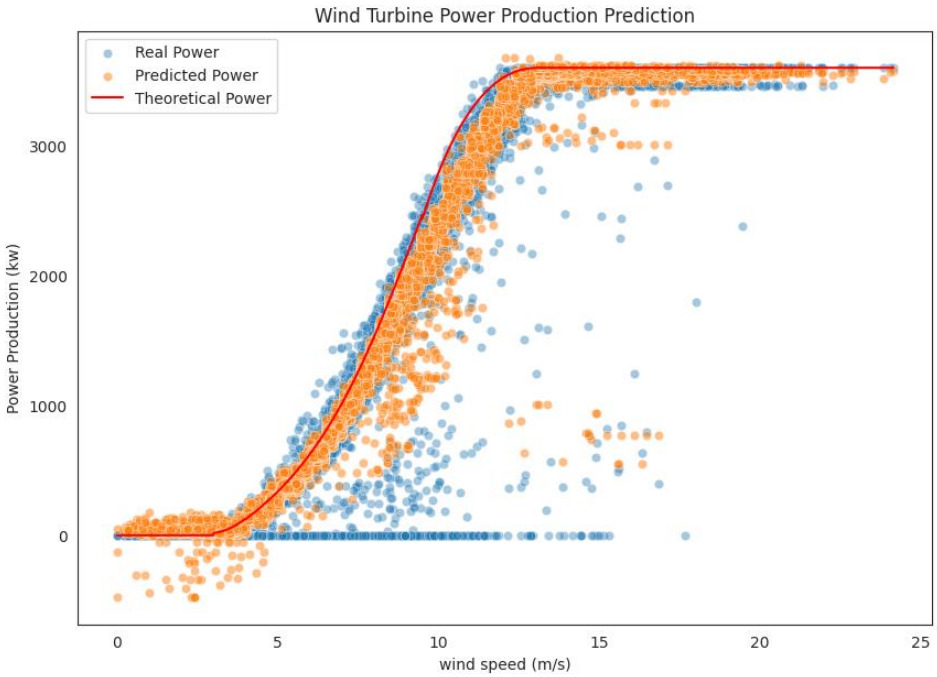

Fig 11. Real, theoretical and predicted power generation versus wins speed.

From a general perspective, theoretical power generation represents a simplification of real power generation. Conversely, the GBTs model satisfactorily reproduced the real power data for wind speeds higher than 5 m/s.

As shown in Fig 11, the GBTs model could produce outbound predictions (real power production > 0). GBT even accurately reproduce real power for wind speed higher than 12.5 m/s. According to the findings, the appropriate solution for real power production prediction consists of using the ensemble method: theoretical power for wind speed lower than 5 m/s and GBT model for higher.

7. Conclusions

a. Apache Spark and MLlib were suitable raw data analysis, processing and modelling tools.
b. Raw data contains many outliers, primarily due to zero power production (wind speed < 3 m/s).
c. March, August and November are the top three months with significant average power production. Conversely, the warm months in Turkey (from April to July) present lower wind speed averages.
d. The average wind speed decrease during the morning and noon.
e. Power generation is strongly correlated with wind speed and not with wind direction.
f. The power production is asymptotic to 3500 kw for higher wind speed than 13 m/s, approximately.
g. Most higher wind speeds have a direction between 170-225° and 15-90°.
h. The best performance metrics were obtained using the GBTs model (R^2 = 0.9768). However, this model could produce inaccurate predictions for wind speeds lower than 5 m/s.
i. The use of an ensemble model is encouraged: theoretical power for wind speed lower than 5 m/s and GBT model for higher.

8. Further works

a. Incorporate extra data of turbine wind as maintenance periodicity.
b. Test the previous models using different outlier handling techniques (e.g. Winsorization, Transformation, and Binning).
c. Hyperparameter tuning of the best model using Bayesian optimization.