Local Wind Forecast Optimization with Machine Learning

Javier Liró Armenteros
Data Science Master's Thesis
Kschool

1. MOTIVATION OF THE PROJECT

Weather forecasting is a long-standing scientific and technological challenge. From the earliest weather forecasting approaches based on human observations dating back to 650 BC to the most modern techniques based on satellite imagery and complex numerical models, the accuracy of these predictions greatly influences a wide range of sectors, from agriculture and energy to tourism and outdoor sports industries.

Despite much effort to improve forecasts, the accuracy of the models is still not as accurate as desired at times, and one of the most difficult variables to predict is wind speed. As an amateur practitioner of water sports that rely heavily on wind and sea conditions, with a professional background in science and technology, I have always been intrigued by how the more experienced practitioners of these sports were able to "interpret" weather charts and based on them make their own customized predictions, in many of the cases much more accurate than those provided by the general forecast weather models.

After a few years of practice, I finally developed that skill that may seem mystical and magical to newcomers, but far from it is based on observation. Observation of weather forecasts and observation of actual conditions on the same day you can show up on the spot, perhaps after a long drive, discovering that the actual conditions were not as good as expected or surprisingly better than predicted.

But that ability is local. What I mean by this is that you can learn what conditions and deviations from the weather forecast will be like in a particular location or region, but each area works differently and the rules used in one place need not apply to another. This local concept is key in this project that aims to improve the forecast accuracy for a particular location. The idea of improving a global weather prediction model developed over many years by a wide range of meteorological experts is not the scope of this project and would be an unreasonable purpose for a Master's project in Data Science.

With these two ideas in mind, the observation and the local component, I decided to develop this project with the following hypothesis: if a human is able to learn to predict based on actual characteristics and observations of wind conditions for a particular location, a machine, with proper guidance, will be able to learn the same skill much faster and with better accuracy. This is a supervised machine learning project.

2. PROJECT OVERVIEW

2.1.Project Objective & Scope

The objective of the project is to make wind predictions for a specific location with higher accuracy than that provided by general weather forecast models.

Datasets of historical weather forecasts and historical weather measurements are going to be processed to train a selection of regression models using supervised machine learning techniques.

After an iterative process of training/testing techniques and feature engineering, the best model obtained is going to be captured to perform real-time wind predictions based on current weather forecast data captured from weather prediction APIs, hoping to improve its accuracy through that machine learning process.

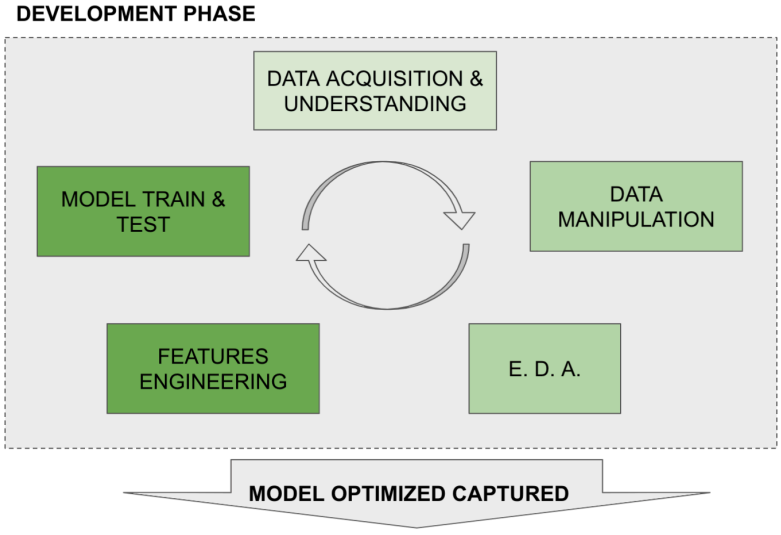
A beta web application has been developed to display current weather forecasts in real time, as well as a dashboard to visualize forecasts versus actual measurements over the historical series.

2.2.Project methodology

Two phases can be distinguished:

1. Development Phase:

The objective of this phase is to apply data science techniques with an iterative methodology to obtain a trained model capable of making predictions based on selected features.



We can subdivide the iterative process into 5 steps:

Data acquisition and understanding: Search for and perform a preliminary assessment of different data sources to evaluate data quality and project feasibility.

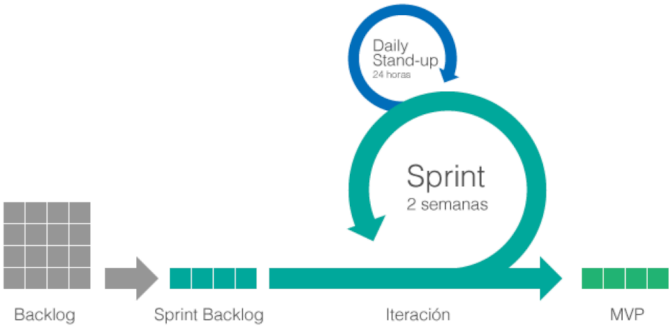
Data manipulation: Prepare and structure the data to enable proper data analysis and deeper understanding of the data.

Exploratory data analysis: Analyze relationships and correlations between features and the behavior and distribution of the data using different data visualizations.

Feature engineering: Perform basic data transformations that allow a machine learning model to understand and interpret the data.

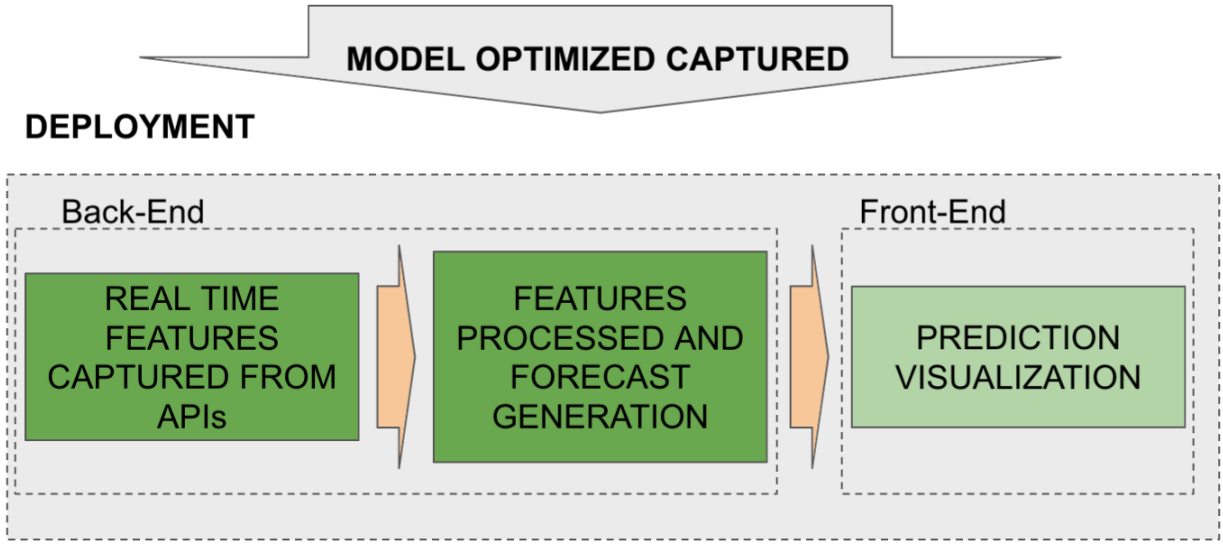
Model training and testing: Train machine learning models with selected features and target values in the training dataset, and measure the quality of the predictions made with the test dataset.

In this type of data projects where each phase affects and is affected by the others and where it is difficult to establish a clear scope and planning at an early stage, the **SCRUM** methodology is optimal. After the preparation of a basic MVP (Minimum Valuable Product), different improvements have been developed in subsequent **SPRINTS**, such as model optimization, feature transformation, improved data analysis or added features of the web application.



2. Deployment Phase:

Once a reliable model has been trained, tested and optimized, a Web App has been created that collects the real-time data from the web APIs and processes it with the transformations inherited from the development phase and the previously captured model. With all this, an improved forecast is generated and displayed through a visual front-end.



2.3. Methodology used in Machine Learning and Feature engineering process

- The methodology followed in the Feature Engineering + Model Training and Testing development phase is explained in more detail below:

- The data set has been divided into training and test subsets.

- The target value to be predicted is the wind speed, which is a continuous variable. We will then train different regression models with the training data and evaluate them with different error metrics for the machine learning regression models.

- The features we use to predict the wind speed at a given location are the weather forecast values given by a weather forecast provider, which provides the wind speed values among other parameters with a certain level of accuracy. We will first evaluate the accuracy of that particular model and that will be the baseline to beat in this project.

- In a first approach, a selection of models has been trained and tested without further hyperparameter optimization and feature tuning. The regressions used are the following

- **Linear regression:** It is an algorithm that assumes linear relationships between target variables and features, and starting from random coefficients/weights for each of the features, improves a cost function by minimizing the error between the regression curve and each of the points of the dataset.

- **K-Neighbors:** KNN is an algorithm that can be used for both classification and regression problems and uses "feature similarity" to predict the values of new points.

- **Decision Tree Regressor:** It is a tree-structured classifier that can be used for both regression and classification problems, dividing the data into different tree levels with simple binary rules to finally obtain the prediction of the target variable.

- **Random ForestRegressor :** It is an algorithm that can solve both regression and classification problems by constructing multiple decision trees and merging them to obtain a more accurate and stable prediction.

• The predictions of the trained model have been evaluated with the test data, using 4 different metrics:

- **Mean Absolute Error:** this is the mean value of the absolute error of all predictions. It provides a good view of how well the predictions fit the actual values.

- **Mean Square Error:** Similar to MAE, but in this case the error value of each prediction is squared before calculating the mean value of the predictions. This gives more weight to outliers.

- **Mean Percent Absolute Error:** The MAPE is the sum of the individual absolute errors divided by the value of each. The result is the mean of the percentage errors.

- **Explained Variance:** This is a metric that gives an idea of what percentage of the variance of the target variable can be explained or predicted from the characteristics. Higher EV values mean better model predictions.

• After training the basic models and evaluating them with the different metrics, an iterative optimization process is initiated in which different modifications will be made to both the model and the features, looking for the best metric score.

• The combination of model modification plus features that obtains the best metrics at the end of the iterative process will be the one implemented to make real-time predictions with the current weather forecast data.

2.4.Technologies used

The development phase has been done in **Python** using the **Jupiter-lab** interphase, generating self-explained .ipynb files combining code, partial code output and text explanation. Data has been stored and manipulated to/from **csv** files. The key Python libraries used in the development phase have been: **Numpy, Pandas, Matplotlib, Seaborn, Scipy, Scikit-learn.**

In the deployment phase, a .py python file has been generated that calls the data in real time using API calls to process them in Python. The optimized model from the development phase captured with the **Pickle** library has been used to make predictions. A visual web interface has been generated using the **Streamlit** library inside a python application.

All the work has been developed within a **GitHub** repository and the control of environments and dependencies has been done using **conda**.

2.5.Data Sources

Two data sources have been merged:

1. **Historical actual weather data:** Historical weather measurements from AEMET weather stations. AEMET is the Spanish Government Weather Forecasting agency that has a wide range of metro Station deployed along the Spanish territory.

Link: https://opendata.aemet.es/centrodedescargas/inicio

2. **Historical Weather Forecasts:** Historical weather predictions from **OpenWeatherData.** OpenWeatherData according to their website generate s predictions based on GFS and ECMWF models applying already some machine learning optimization. Their predictions are going to be the baseline of the project, so the baseline is already going to be a challenge to overcome. Hopefully, as the project is very focused on one of the variables (wind) and on a specific location, there is still room for improvement.

Link: https://openweathermap.org/api

2.6.Dataset variables explanation

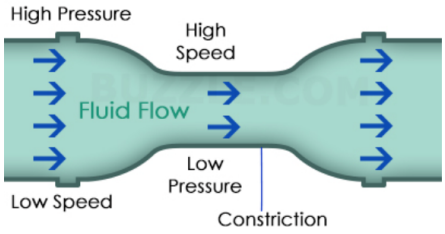
In the following table there is brief description of the different variable that can be found in the dataset used in this project (after dropping unusable columns, windProject_4ML.csv).

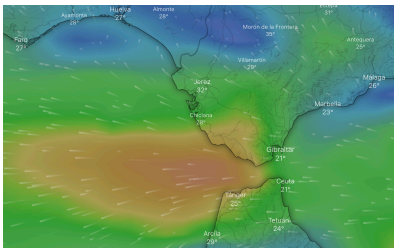
Variable	Description
Forecast dt	Date & time when a prediction has been made
Slice dt	Date & time for when a prediction has been made.
Lat	Geographical coordinates of the location (latitude)
Lon	Geographical coordinates of the location (longitude)
Temperature	Temperature forecasted ºC
Dew point	Atmospheric temperature (varying according to pressure and humidity) below which water droplets begin to condense and dew can form. ºCelsius
Pressure	Atmospheric pressure on the sea level, hPa
Humidity	Humidity, %
Clouds	Cloudiness %
Wind_speed	Wind speed forecasted in knots
Wind_deg	Wind direction forecasted in degrees
Rain	Rain forecasted (mm).
Convective	Convective precipitation (mm) forecasted.
Direction N-S	Vertical linear projection of wind direction modulus forecasted.
Direction E-W	Horizontal linear projection fo wind direction modulus forecasted
Tarifa_wind	Meassured wind (knots)
Tarifa_windGust	Measured max wind speed (knots)
Tarifa_windDirection	Measured wind direction (degrees)
Tarifa_Direction N-S	Vertical llnear projection of measured wind direction
Tarifa_Direction E-W	Horizontal llnear projection of measured wind direction
Delta_Wind	Diference between wind measured and wind forecasted
Delta_N-S	Difference in vertical linear projection of wind forecasted vs wind measured.
Delta_E-W	Difference in horizontal linear projection of wind forecasted vs wind measured.

2.7.Location selected

The location chosen to carry out the project is Tarifa, Spain (Long.: -5.59883, Lat: 36.013985). This place is known for its strong winds derived from its unique geographical characteristics. The distance between Africa and Europe changes drastically from hundreds of kilometers to 14.4 kilometers at the narrowest point, which causes an acceleration of the prevailing winds in the region, the westerly and the easterly wind, due to the Venturi effect.

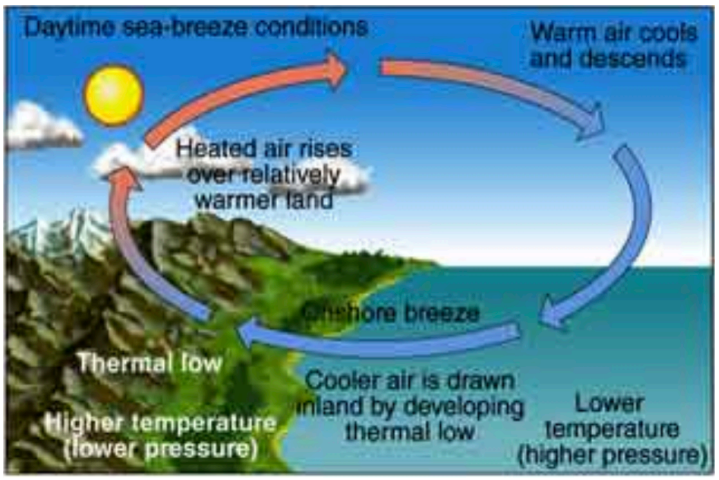
The **Venturi effect** states that in a situation of constant mechanical energy, the velocity of a fluid passing through a constricted area will increase and its static pressure will decrease.





Another local effect that affects this and other coastal locations is thermal winds. In general, winds are generated by pressure gradients between two spatial points. When there is a noticeable temperature gradient between sea and land, for example on sunny spring days when the water is still cold from winter, the warm air from the top of the coastal land tends to expand and rise and cold air from the top of the water, with lower temperature and higher pressure, tends to occupy the space at the top of the land, creating a cycle that can significantly increase wind locally.

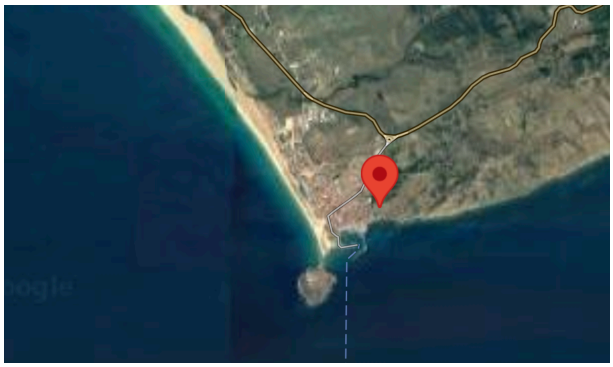
This two effects make specially hard to make accurate wind predictions at this location.



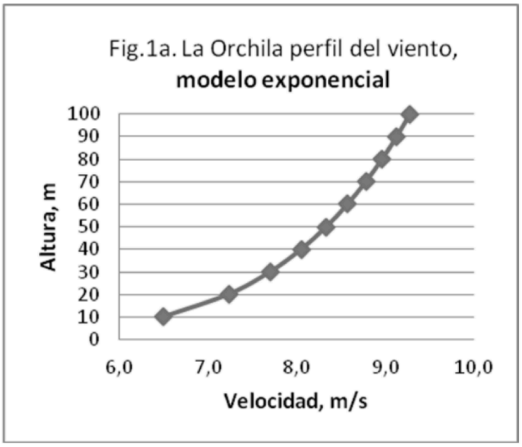
2.8.Hypothesis

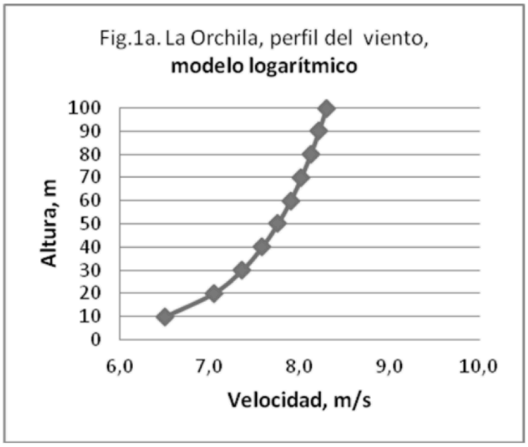
- **Accuracy of weather stations:** The reliability of the predictions made in this project is directly related to the accuracy of the measurements at the weather stations. Any systemic deviation in measurement will be learned by the predictive model. The objective of this project is not to evaluate the accuracy of the AEMET weather station, but to demonstrate that weather prediction models can learn from weather station data and improve their accuracy.

- **Location of the weather stations:** The AEMET Tarifa weather station is located at coordinates lat = 36.01398 lon = -5.59883. The location of the weather station may vary the results in the machine learning project, as the weather station may be better exposed to a certain wind direction. Again, the objective of the project is not to evaluate the accuracy of the weather station, but this observation should be taken into account in case the project wants to be replicated at a specific location, and the exact position of the anemometer can be chosen.



- **Height of the weather station:** The wind speed is significantly affected by the height, especially in the first 100m. There are different empirical and theoretical curves describing this relationship as shown below for another location. The effect of height is outside the scope of this project because the forecast data requested from the weather service API were made at exactly the same location where the weather station is located, and the model predictions are given by the height of the land surface at the specific longitude and altitude used in the API call.





3. WORK FLOW & SUMMARY OF THE RESULTS

The project has been developed within a GitHub repository, with the following files structure:

Variable	Description
Wind Forecast Project Report.pdf	Descriptive report of the project.
README.md	Brief summary and instruction of how to execute the files in the repo
1_DataPreparation.ipynb	Jupyter-lab file. First file of the development phase, data preparation.
2_EDA	Jupyter-lab file. Second file of the development phase, E.D.A.
3 FeatEng.ipynb	Jupyter-lab file. Third file of the development phase, Feature Engineering & Machine Learning.
environment.yml	Environmental file with the dependencies needed to run the project
.gitigonre	Hidden file to avoid uploading unwanted file to GitHub repository.
app.py	Application with web interphase that process information from OpenWeather and AEMET APIs and create realtime weather forecasts for the study case location. **To run the web app, execute from the terminal within the project folder:** **>streamlit run app.py**
app1.py	Sub-app inside app.py. Should not be executed directly.
app2.py	Sub-app inside app.py. Should not be executed directly.
windpredictor.sav	Wind forecast machine learning model captured with pickle library.
Data/StationsRecords/DH-6001.csv	Weather station records located in Tarifa.
Data/StationsRecords/DH-4554X.csv	Weather station closed to the target location.
Data/StationsRecords/DH-6329.csv	Weather station closed to the target location.
Data/historicalForecast.csv	Historical weather forecasts made by https://openweathermap.org/

Let 's navigate along the project to remark the most important insights and difficulties found.

3.1.MODEL DEVELOPMENT PHASE

The development phase has been divided into 3 phases, each of them developed in a different file.

3.1.1.Data Preparation

This phase is developed in the file: "**1_DataPreparation.ipynb".**

The objective on this step is to do a basic inspection of the dataset and manipulate it in order to allow posterior visualizations and analysis.

Our dataset is a compounds of two datasets:

- **Weather Station Dataset:** with actuals values of the target variables (wind, wind gust, wind directions) measured on a site weather stations.

- **Historical Forecast Dataset:** with historical forecasts that will be uses as features to train our forecast model.

The key transformations done to the datasets before merging them are:

- **Forecast dt vs slice dt:**

The historical dataset has two types of dates: forecast dt (date and time at which the forecast was made) and slice dt (date and time at which the prediction was made). For example, a forecast with slice date 2017-10-13 15:00:00 +0000 UTC was made on forecast date 2017-10-09 00:00:00 +0000 UTC,

4 daily forecasts are made at 00.00, 06.00, 12.00, 18.00, which means 4 daily dt forecasts.

For each of these forecasts, 93 dt predictions are made in slices, starting at 3 h intervals and ending at 12 h intervals.

There are then up to 93 forecasts for each hourly date. To avoid having different forecasts for the same date and time, but also to ensure that each date and time has a forecast, the forecasts made by the daily dt 00.00h forecast are the ones to be used, and only the first 8 dt slice of each forecast made by the dt 00.00h forecast will be (8 dt slice x 3 h interval =24 hours).

- **Time granularity of datasets:**

Whether station measurements are provided hourly while the historical forecast are given in three hours intervals. To adapt these intervals before merging the datasets, two options have been evaluated:

- Option 1:Adapt the granularity of the weather forecast to 1 hour, creating artificial values for each hour.
- Option 2: Adapt the granularity of the weather station to 3h, using the mean value of each 3-hour interval.

Option 2 was chosen because it was considered better to have a mean value of real values, which is still a real value, instead of creating 3 records from 1 record by creating artificial values.

- **Components of wind direction:**

The wind direction is given in degrees. The interpretation of this value by a Machine Learning model is a problem. For example angles of 359º and 1º, but a mathematical algorithm interprets these values as very different ones.

To solve this problem, the unit vector of each direction has been decomposed into the linear components x, y, having then the North-South component and the West-East component of each wind direction angle.

- **Merging the data frames:**

To merge the two data frames, the date-time of each prediction/measurement will be used as a key, then the date-time format of the data before merging has been adapted.

3.1.2.Exploratory Data Analysis

This phase is developed in the file: **2_EDA.ipynb**.

In this phase some visualization has been performed to deeply understanding the data and perform further preparation of it before starting the Machine Learning and Feature Engineering phase.

This stage is subdivided in 3 substeps:

1. Data understanding and drop of unusable columns

The objective in this phase was to eliminate some unneeded data.

The reason for eliminating columns has been:

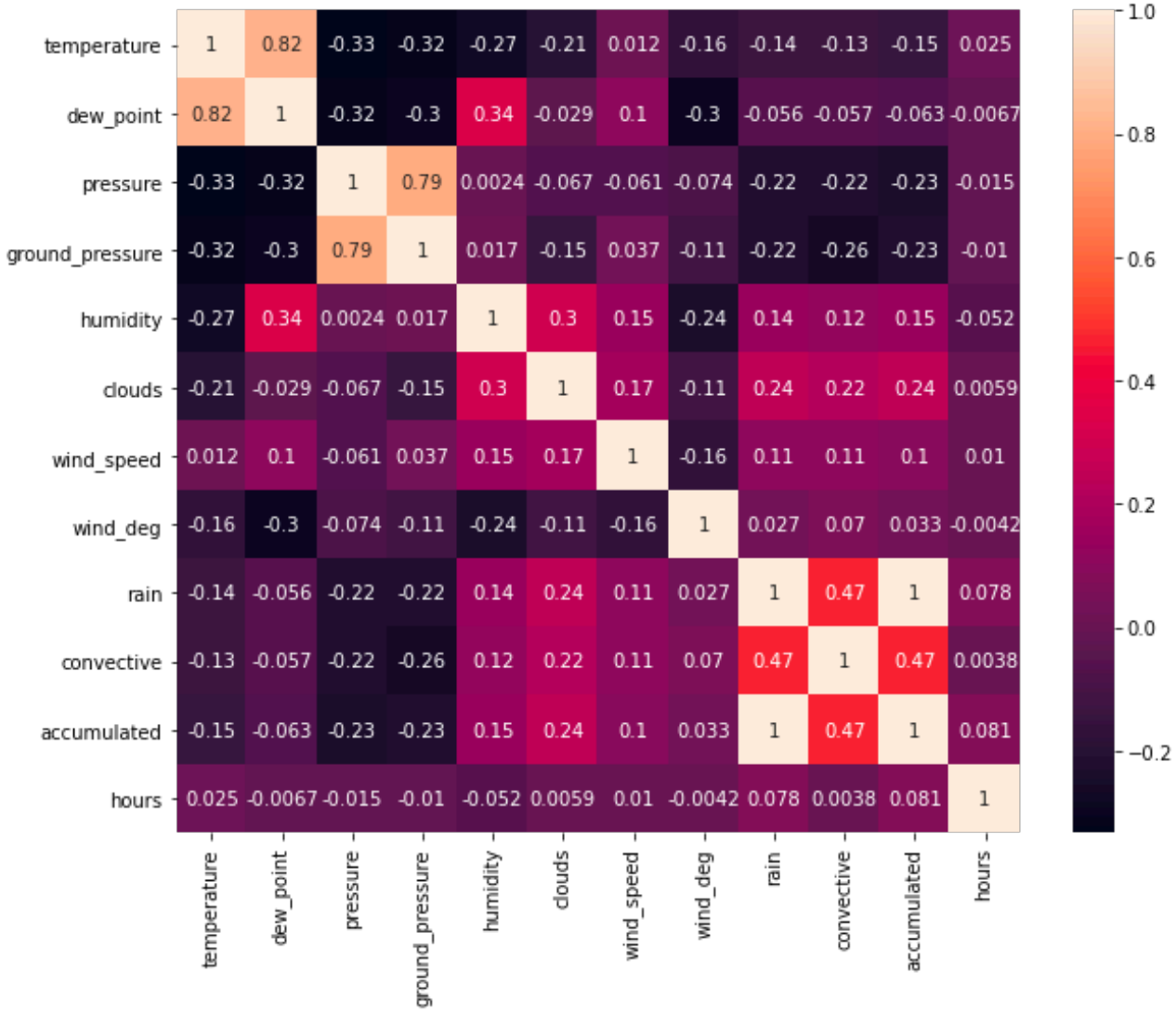
a) Columns data is empty. *Some columns were empty or erratic.*

[6]:
```
print(df['ice'].sum())
print(df['snow'].sum())
print(df['fr_rain'].sum())
print(df['rate'].sum())
```

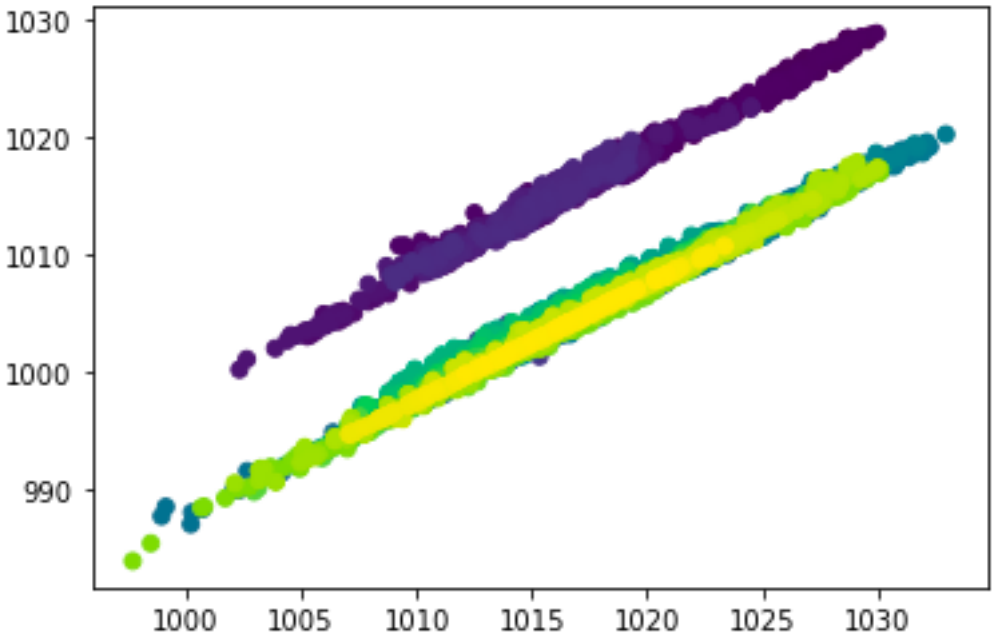
```
0.0
0.0
0.0
0.026
```

This columns are either empty/not relevant /erratic for our project so we drop them

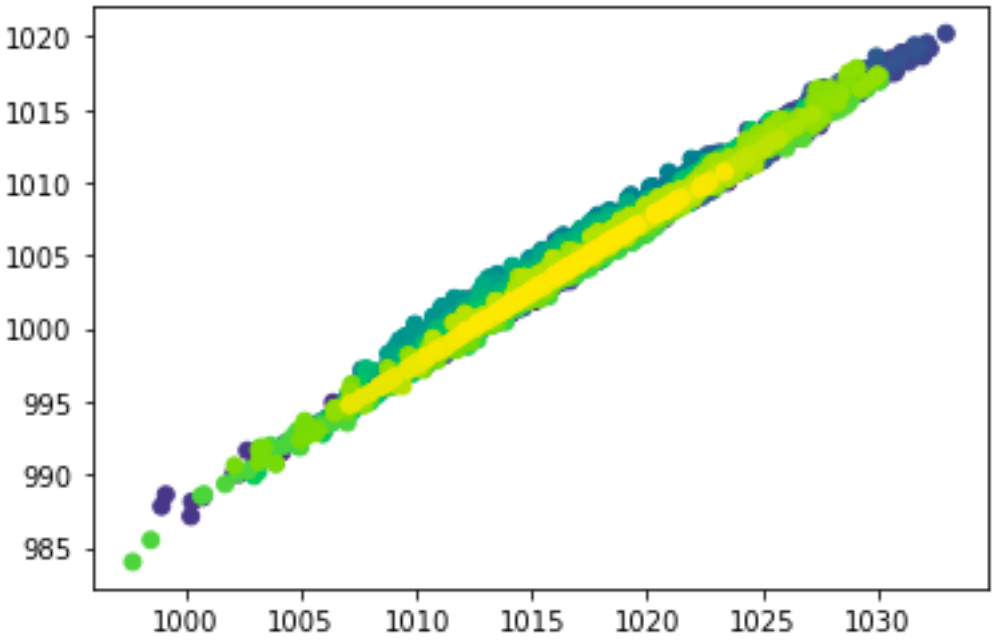
b) ***Correlation between characteristics :*** After performing a correlation analysis, we can eliminate columns that are highly correlated, such as precipitation and accumulated (rainfall).



In some cases there were hidden correlations that needed further investigation such as the relationship between pressure and ground pressure. The correlation plot shows a correlation of 0.79 between the two variables that is relatively high, but they are still independent variables. If we distribute the two variables also taking into account the time at which the data were recorded, we can see the following plot:



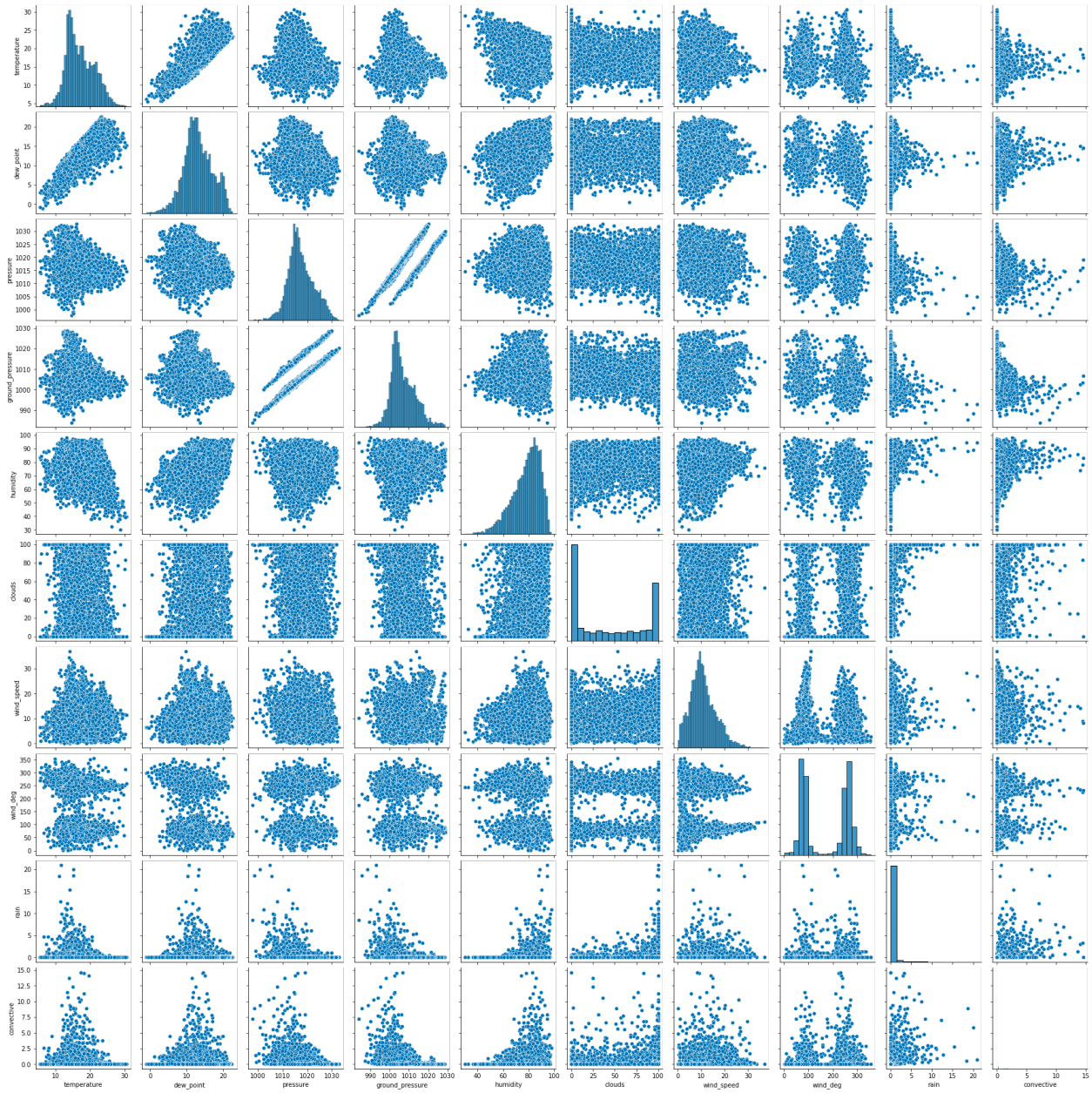
In the previous graph the pressure and ground pressure were shown and the color of the dots is related to the date and time when it was recorded. If we filter out only the distribution of pressure and ground pressure from a particular time on the timeline, we find that there is a perfect correlation between the two variables, as we can see in the graph below, which means that we can eliminate one of them from our set of variables.



2. *Data distribution exploration*

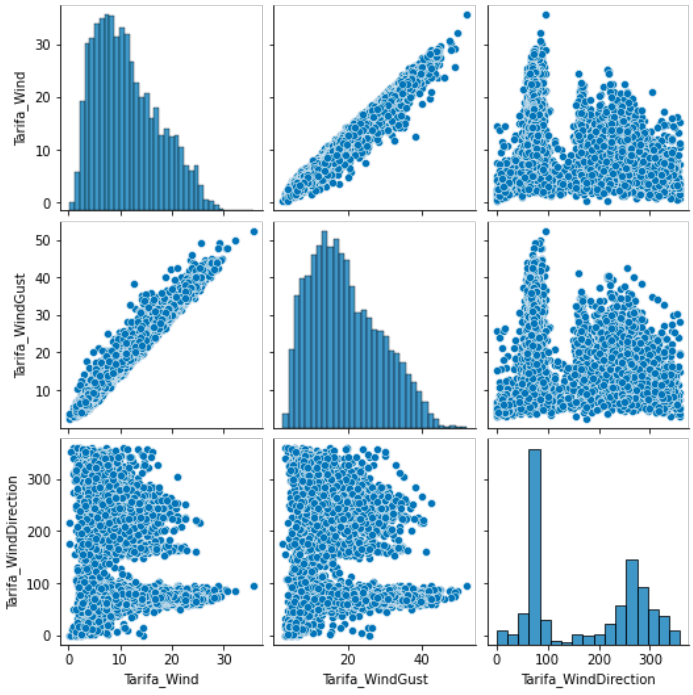
Having a look in an overall pair plot the most important insights are:

- The perfect hidden correlation between pressure and ground pressure mentioned in the previous section.
- The clustering according to wind direction, grouping and dividing the records around the 90º wind (East wind) and around 270º (West wind). These are the dominant winds in the area and we can expect different patterns or behaviors depending on them.
- The cloud distribution shows two peaks which means that there are two important groups of days, clear days with cloud cover below 10% and cloudy days with cloud cover above 90%.



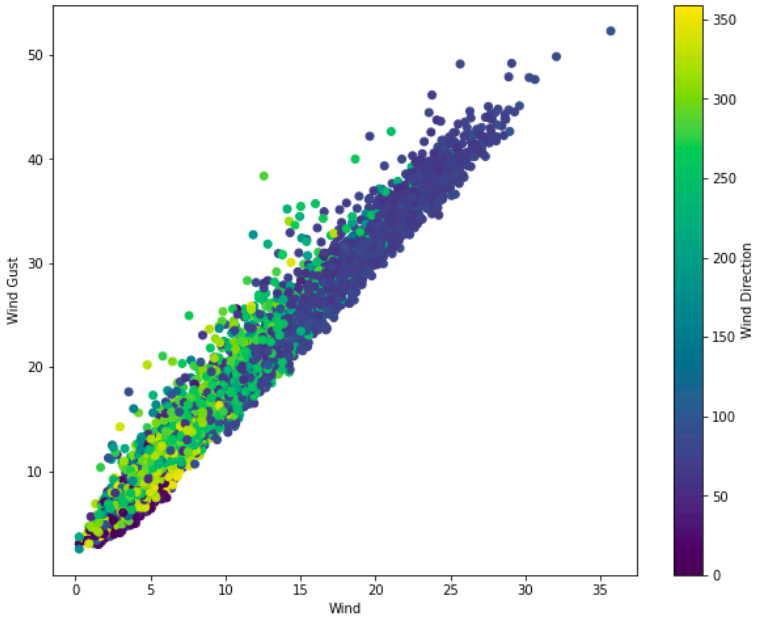
Looking at the anemometer distribution of the weather station we can see:

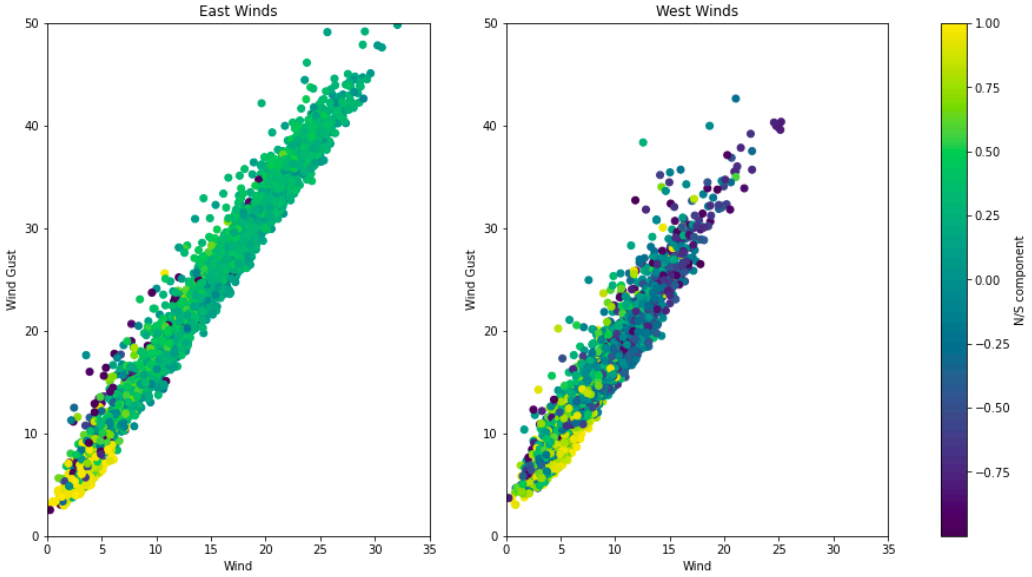
- The clustering around the prevailing easterly and westerly wind is also shown.
- There is a linearity between wind speed and wind gust.
- Winds from the east can reach higher speeds than those from the west.



The relationship between wind and wind gust is relevant as it can be an interesting metric of wind quality. A gusty wind is considered to be of lower quality. It is well-known that easterly winds in this area are gustier than westerly winds and northerly winds (NE, NW) tend to be gustier than southerly winds (SE, SW), so additional data research has been conducted.

In the following figures the relation between wind and wind gust has been shown in color by mapping the wind direction in the first plot and the north-south component in the second by dividing the distribution into two plots, one for easterly winds and one for westerly winds.



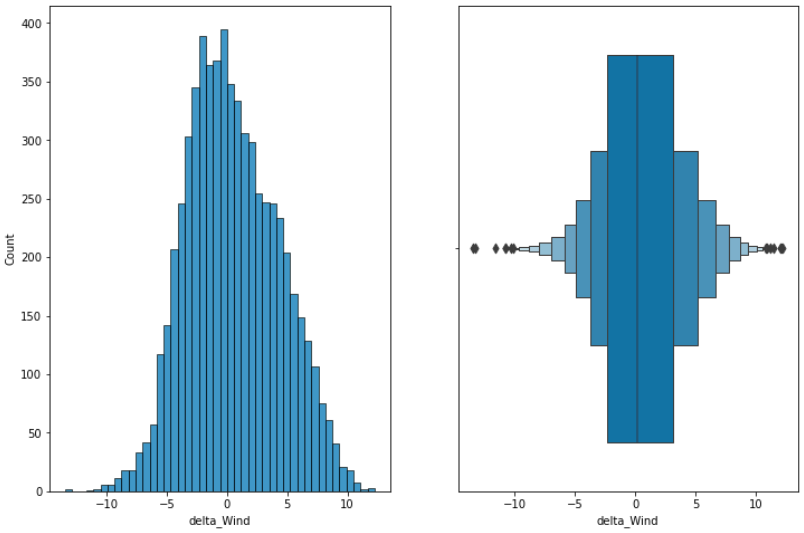


After inspecting these graphs, the conclusion is that we cannot have a clear differentiation in the relationship between wind speed and wind gust as expected, showing in all cases that there is a linear relationship between wind speed and wind gust with a very similar slope in all cases.

The possible reason behind this is that our variables wind speed (average wind speed over a 3-hour time interval) and wind gust (maximum wind speed over a 3-hour time interval) are not really showing the wind gust, understanding gust as the sudden and abrupt change in wind intensity. We would need to measure the ratio of mean wind to maximum wind at shorter intervals to be able to perceive these abrupt changes and consequently have a measure of wind quality, whereas a 3-hour interval does not show how abruptly or progressively the intensity changes.

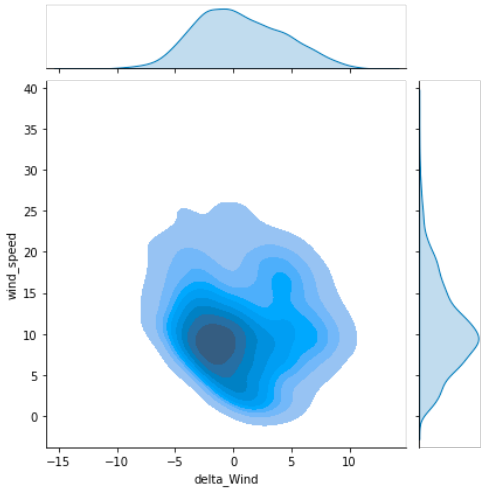
3. Relation between features and actual measurements

In this part of the E.D.A. we are trying to understand overall relations between the features and the target variable. A new variable have been created to better understand of this relation has been created called delta_wind, that is the difference between the forecasted wind and the measured wind.

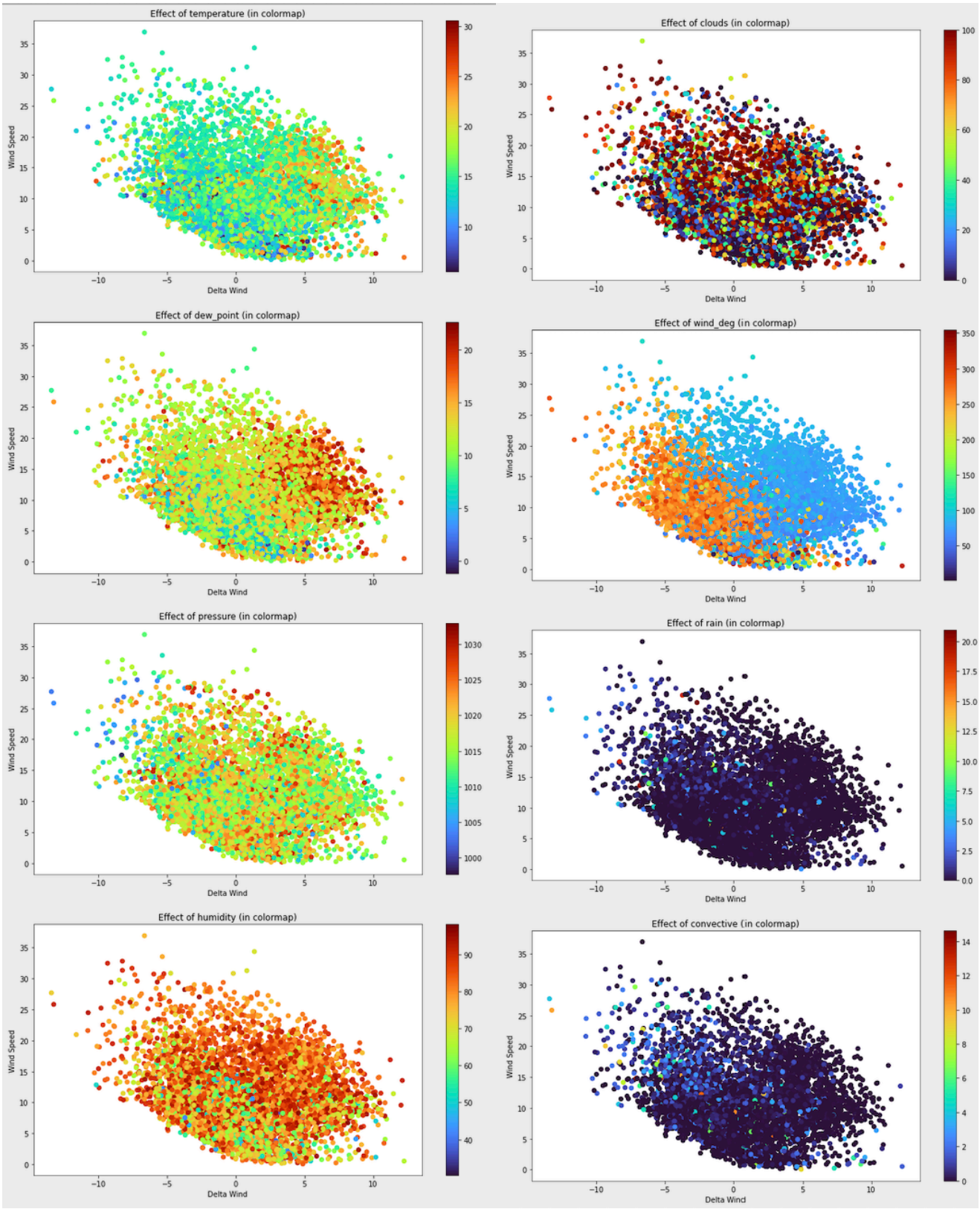


The first thing we can see is that the delta wind is relatively well centered with a mean value of 0.4818 knots, but the standard deviation is relatively high 3.808 for a mean measured wind value of 11.3398.

A first look at the relationship between predicted wind speed and delta wind shows again that, although the distribution is relatively centered around delta wind values close to 0, there is a significant amount of values around the 10 knot wind speed where the wind is underestimated (the actual wind is higher than the predicted wind.



Let’s see how the features variables affect the previous graph, using colormaps to show how each feature affect the data distribution.

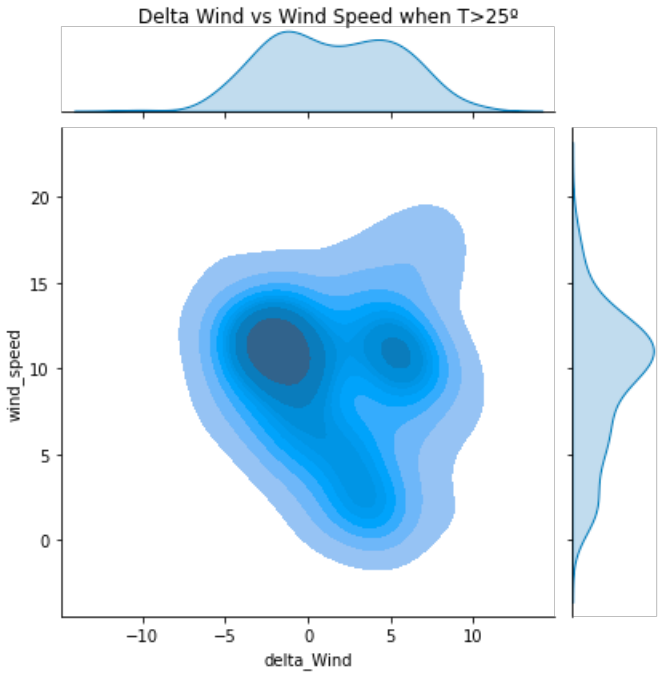


Here we appreciate two important points:

- **Temperature and dew point:** These two variables (which are highly correlated with each other) significantly affect the delta between predicted and measured wind. When temperatures are high, the tendency is for the measured wind to be higher than the predicted wind. This is very interesting because the thermal wind is difficult to predict and our machine learning model can learn this deviation and improve the prediction accuracy.

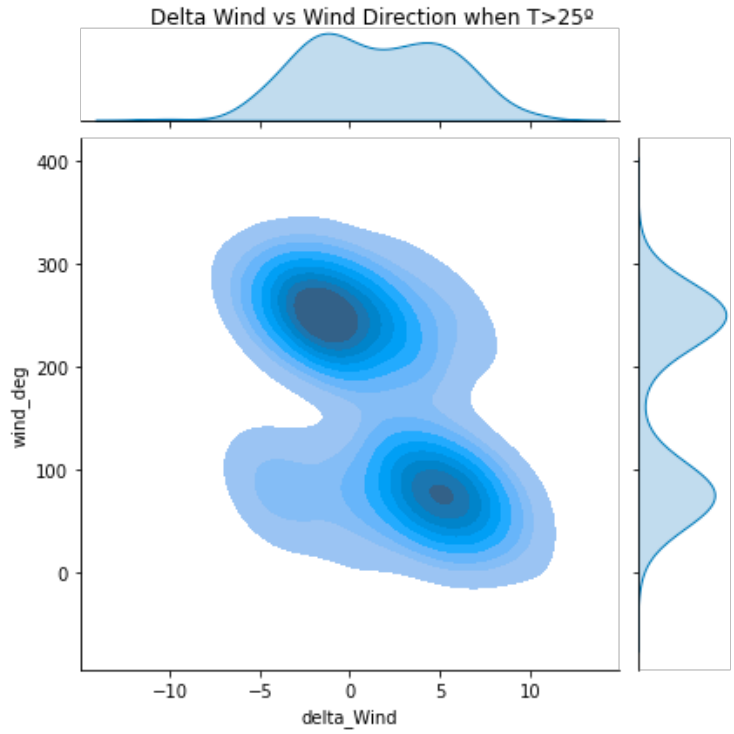
- **Wind direction:** we can see from the color distribution that easterly winds tend to be below the forecast while westerly winds tend to be slightly above the forecast. This may be due to inaccuracy of the forecast model (which would be good room for improvement for our machine learning model) or inaccuracy of the weather station (the station may be more exposed to the east wind than the west wind and that may alter its reliability). Unfortunately, with the data provided it is not possible to determine which of the two effects trumps the other, and assessing the accuracy of the weather stations is beyond the scope of this project.

To dig a little deeper into how temperature may affect the delta between predicted and measured wind we will filter out records where the temperature was above 25 degrees:



An interesting behavior is shown in this figure, with a new cluster appearing at +5 knots of delta wind in the records where the temperature is higher than 25°.

Then, filtering for temperature higher than 25° and plotting the relationship between wind direction and delta wind we obtain the following figure:



In the figure we clearly see that when the temperature is greater than 25° and the wind is coming from the East there is a group of points where the wind is underestimated around 5 knots. This shows that there are different patterns that can be learned by a machine learning model that hopefully can improve the accuracy of wind predictions of global weather forecast models. that the temperature was above 25 degrees:

3.1.3.Machine Learning - Feature Engineering

This phase of the development has been performed in the file: **3 FeatEng-ML.ipynb.ipynb**.

This file has two blocks, first is a Null/NaN values treatment and second one is the Machine Learning & Feature engineering block.

1. Null/NaN values treatment

Different approaches have been done to treat each variable independently as shown in detail in the Jupiter notebook:

- **Measured wind speed:**

Exploration: This is the target value of the project. There is 19 missing values out of 6346 registers(0.2%). The registers missing are consecutive within a 2 days period of time.

Strategy: Because the missing values are consecutive, the variable we are studying is the target itself and there are only 11 missing values, the best option considered is to eliminate the 11 rows with the missing values, there is no added value in creating artificial values for them.

- **Measured wind direction:**

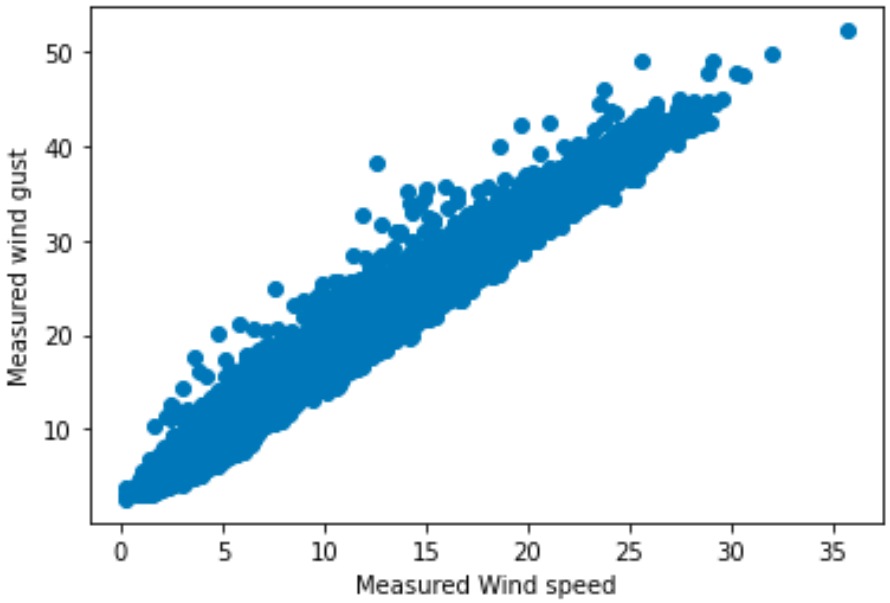
Exploration: The mean deviation between the forecasted wind direction and the measured wind direction is just -2.089º (0,8% of 360º) and there is just 11 NaN values (11/6330 --> 0,1% of the registers).

Strategy: Approximate the 11 missing measured wind direction with the forecasted wind direction.

- **Measured wind gust:**

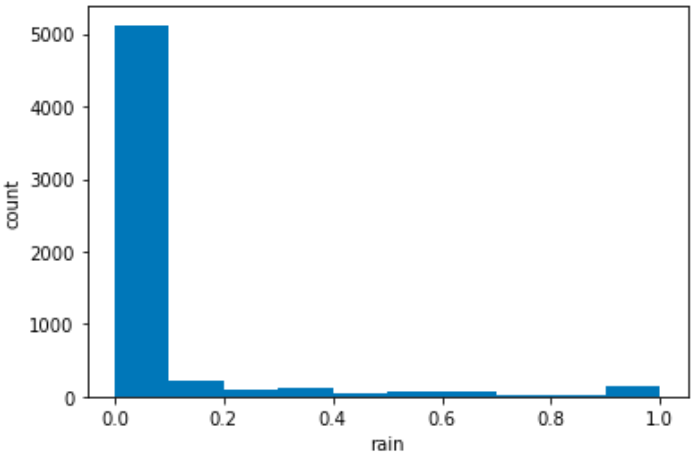
Exploration: As seen in the exploratory data analysis, there is a high linearity between wind speed (average wind speed within 3 hours interval and the wind gust).There are 10 missing values.

Strategy: Approximate the 10 missing values with a linear regression relation between wind speed and wind gust.



- **Forecasted rain:**

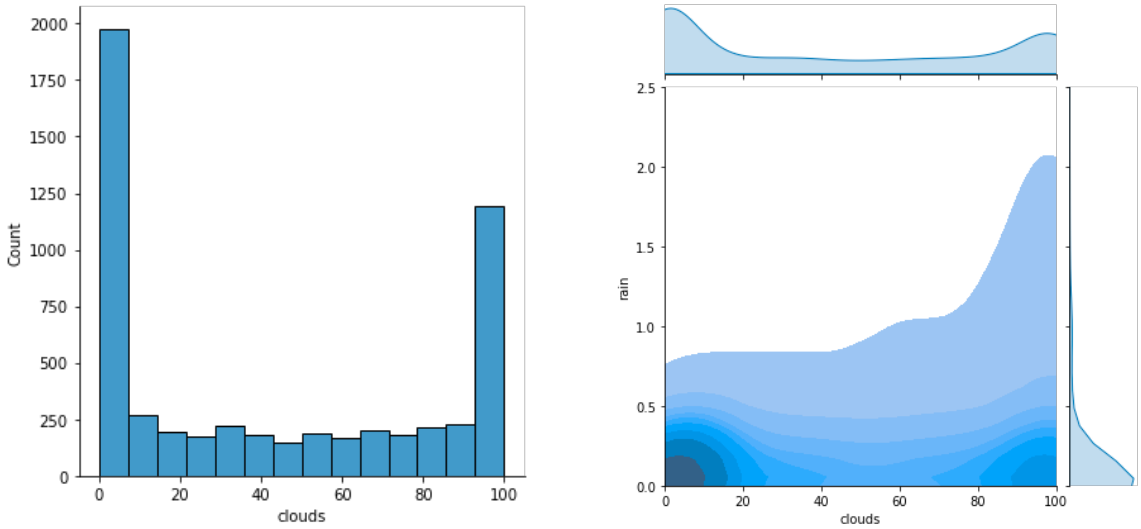
Exploration: The median value of rain is 0.0 mm and the mean value of rain is 0.21 mm. In 77.9% of the time slices there is no rain.



Strategy: After deliberating whether to approximate the missing values with the median value or with the mean value, it has been decided to use the median value of 0.0 mm for the missing values taking into account the shape of the distribution and the fact that 77.9% of the time slices have no rain.

- **Forecasted clouds:**

Exploration:



Exploring the distribution of the variable clouds (left figure) we can identify that cloudiness tends to polarize between clear and cloudy days/intervals.

Exploring the relationship between clouds and rain (right figure) we can observe that clouds and rain have an important relationship: rainy days/intervals tend to be more cloudy than non-rainy days/intervals.

Strategy: Instead of using a single mean value to fill in all missing values of cloud cover, two mean values have been used, the mean cloudiness value of intervals without rain (33% of cloud cover), and the mean cloudiness value of days with rain (75.96% of cloud cover).

- **Forecasted convective rain:**

Exploration:Not clear relation between rain and convective rain have been found apart from the fact that when rain is 0, convective rain mean value is also 0.

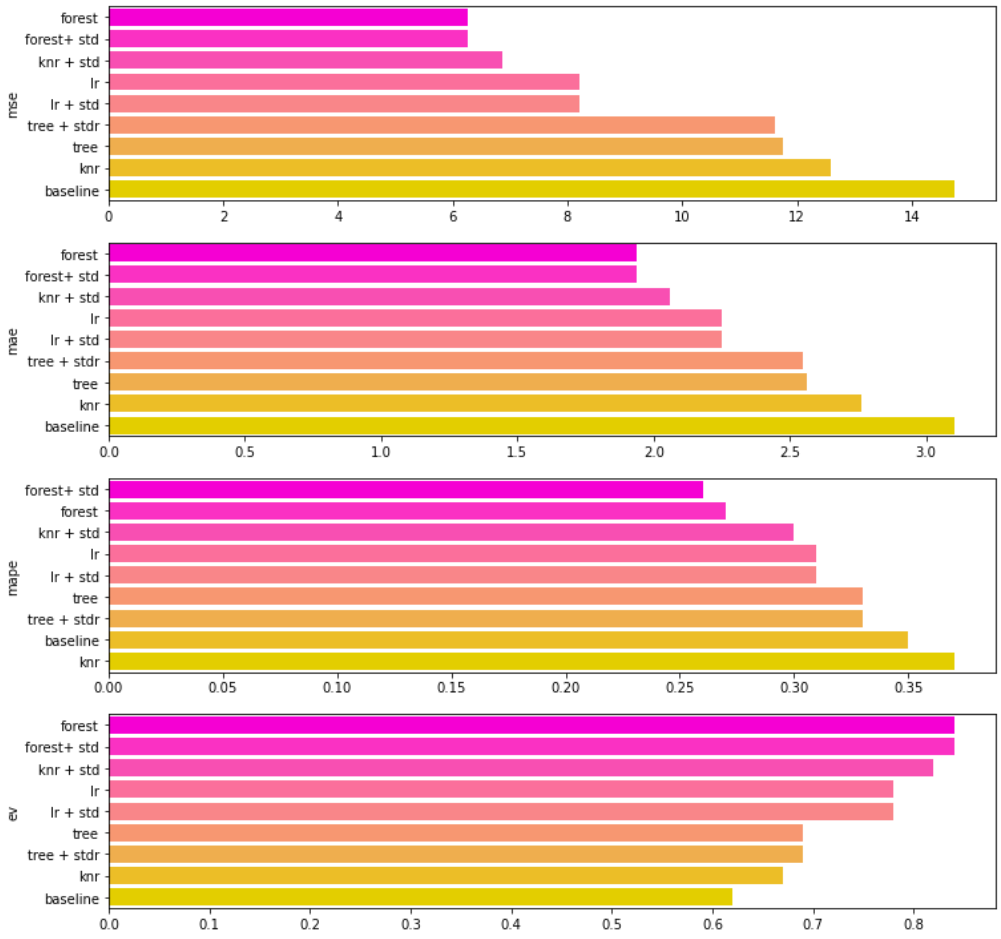
Strategy: convective rain for the days when rain is 0 is approximated to 0, and for the rest of the days, mean convective rain value is used.

2. *Machine Learning and Feature engineering:*

Let's deep dive in the results of the different steps:

I. **Regressor algorithms:** In the first approximation the 4 regressors selected (linear regression, KNN, Decision Tree and Random Forest) have been trained and tested with the train and test datasets and the best metrics have been obtained with **Random Forest.**

II. **Regressors with previous normalization:** Regressors with prior normalization: In the second approach, the same 4 regressors have been trained and tested, this time with a prior normalization of the variables. <u>After studying the result, it was found that in models where the euclidean distance is not relevant, such as Random Forest, there is no significant improvement in the metrics.</u> On the contrary, KNN improves significantly with a prior standardization of the data still with worse metrics than the Random Forest model. From this point onwards the subsequent optimizations to be tested are going to be based on Random Forecast which is the best performing model without prior data standardization as it does not improve the metrics.



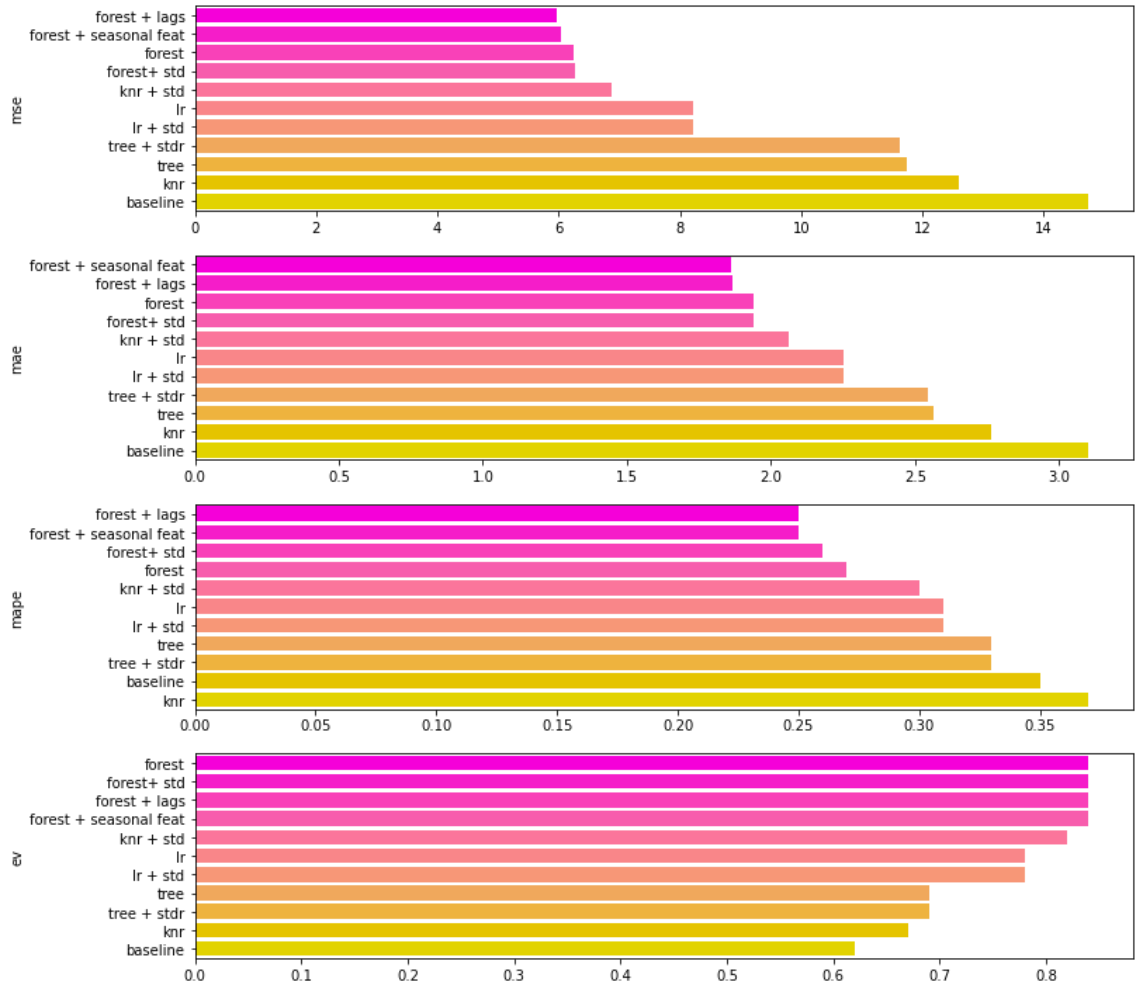
III. Adding lags to features.

Lagging of the most important features has been added, in order for the model to obtain information from the features in the previous time intervals to improve the forecast. Improvements have been obtained in 3 of the 4 metrics.

IV. Adding seasonal features.

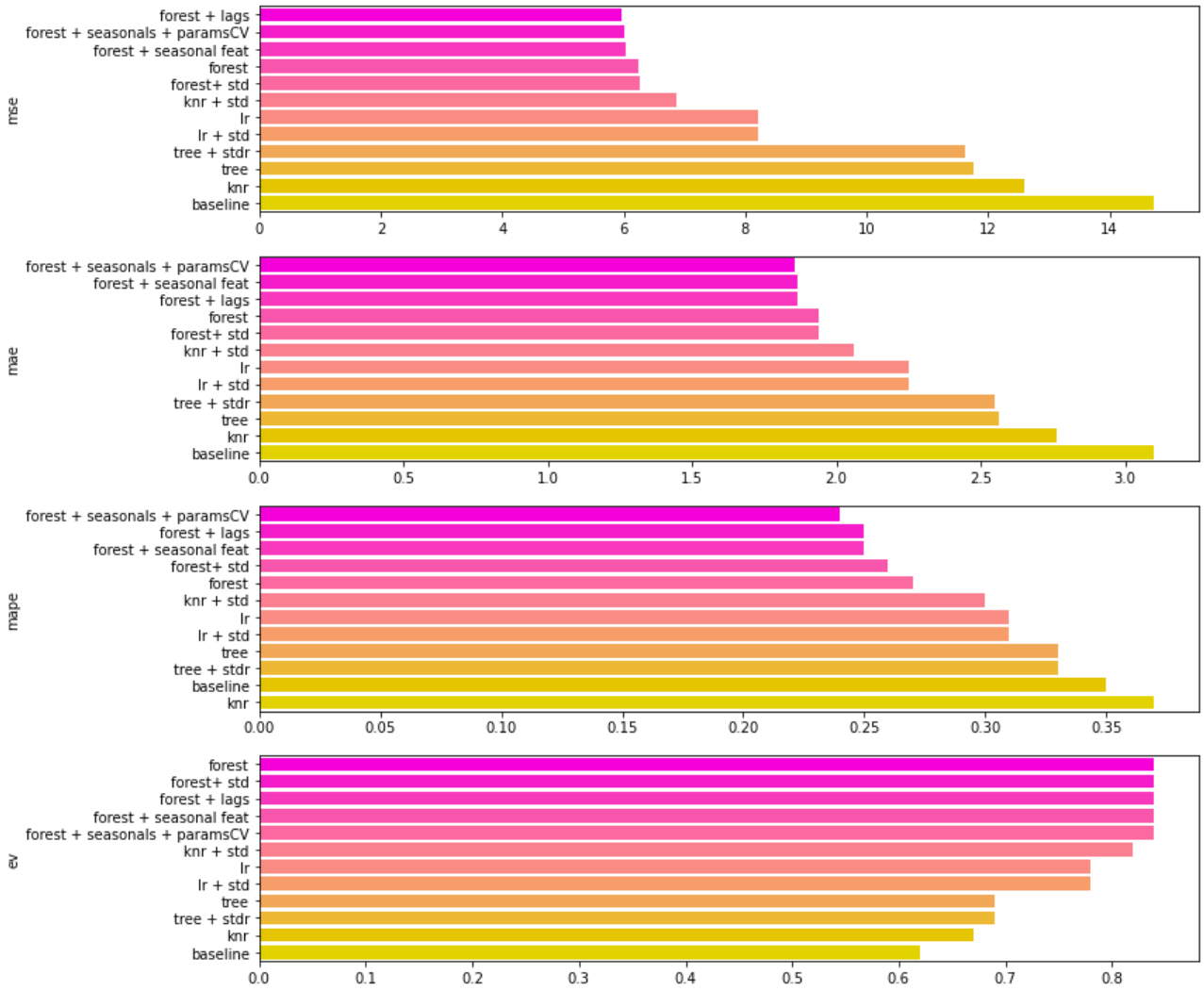
Daily sinusoidal and cosine seasonal features have been added to the dataset to allow the model to interpret the part of the day when the forecast was made. This approach also results in an improvement in 3 of the 4 metrics compared to the simple random forest model and a slight improvement over the lag addition approach.

REVIEW AFTER DEPLOYING THE MODEL: *Adding lags to the features is a problem in the deployment phase, because it implies the generation of NaN on the lagged features in the first records. In the deployment phase we are going to generate predictions for 48 time intervals. Generating 2 lags of each feature means generating 2 rows with NaN. 2 records out of 48 is systematically not efficient, which means that the slight improvement we get by adding lags does not translate into a better prediction. So, let's omit the lags approach and keep only the seasonal features that get similar metric performance.*



V. Optimizing hyper-parameters by means of Grid Search Cross Validation.

The last step performed is the optimization of the most relevant hyper-parameters of the Random Forest, the number of estimators and the maximum depth, obtaining slightly better results in some of the metrics since there should not be a very large margin of improvement at this point.



After the battery of tests the model option to select for deployment will be the Random Forest adding seasonal features with optimization of the parameters obtaining the best MAE MAPE metric. The EV metric is equal for a set of options and the MSE places Random Forest + added Lags in first position, which as discussed above is not optimal for the deployment phase.

Just an observation is that this metric may vary slightly in different runs and with different training and test splits, but overall the ranking of the best options remains similar.

3.2.MODEL DEPLOYMENT PHASE

3.2.1.WebApp

Once the optimized predictive model was obtained, it was captured using the python pickle library.

A new python App.py file has been created that captures the current features for the next 48 hours from the same website where the training-testing dataset was obtained (OpenWeather.org) through via API, perform the same data transformations that were done on the train-test dataset and then apply the model saved with pickle. The result is a real-time wind forecast with an improved forecast for the next 48 hours.

3.2.2.Front End

The library **Streamit** from Python has been used for generating a visual web interphase that provide to the user the forecast for the next 48 hours and other interesting parameters as well as the real wind measurements obtained from the **API** of the weather station used in the project. To run the app and launch the web interphase is necessary to execute the following sentence from the project folder:
> streamlit run app.py.

4. CONCLUSIONS

This project demonstrates that the application of supervised machine learning can significantly improve the accuracy of weather forecasts by training a model with historical data coming from meteorological stations. However, this improvement is highly dependent on the accuracy of measurements of the meteorological stations. For example, in our project focused on the wind variable, there is a large variability of the wind depending on the altitude and the potential obstacles that may be around the anemometer, and obviously the machine learning model will be as good at learning wrong measurements as it is at learning good measurements.

This improvement in weather prediction has local applicability, as the improvement in prediction accuracy is directly related to the distance to the weather station. With this in mind, there is still a wide range of applications in places where accuracy is critically relevant, such as a wind power station, airports or harbors.

With today's computational power, and considering the amount of data to be processed, it is worth evaluating how different models perform and produce better results. In our particular case, Random Forest has proven to be a perfect model in terms of accuracy and computational performance.

Some assumptions coming from domain experience, such as the daily seasonality of the deviation of the actual wind versus the predicted wind or the dependence of that deviation on how other variables such as temperature or precipitation behave in the previous hours, as we saw when adding variable lag features to the dataset.

As a final thought, the only aspect that makes this project theoretical and not practical is the rigor of the weather station measurements, which in the case of this project has not been checked, but the process followed here can be perfectly replicated for any location with any available predictive model, with a high guarantee of success in improving the accuracy of the predictions.