2 December 2024

vazzoler.federico@gmail.com

Efficient Classification of Solar Variability: A Machine Learning Approach for PV Performance Analysis using OSMER data

Federico Vazzoler
Hamburg, Germany

Keywords: Weather data, Classification, Machine Learning

Summary

This study explores weather classification methods using data provided by the Friuli Venezia Giulia Regional Meteorological Observatory. A traditional approach based on binormalized daily solar radiation patterns was applied to classify days from 2012 into distinct categories across three locations. This method served as a baseline for comparison against machine learning (ML) techniques, which utilized daily averaged meteorological features for day-type classification.

The ML approaches demonstrated notable advantages by eliminating the reliance on detailed hourly measurements, thereby offering a more efficient and scalable solution. Among the tested models, the Support Vector Classifier (SVC) outperformed others in distinguishing clear and cloudy days, showcasing robust classification performance. However, all models faced challenges in accurately identifying the intermediate *quasi-clear* day class, likely due to overlapping feature characteristics and potential limitations in the traditional binormalization-based method.

These findings underscore the potential of ML-based approaches to improve day-type classification accuracy and scalability, while also highlighting areas for refinement in traditional methods and feature engineering.

Contents

1 Introduction

As solar photovoltaic (PV) electricity production continues to grow, its inherent variability presents significant challenges for operational management. Efficient distribution network (DN) management requires a thorough understanding of solar irradiation patterns to predict and control the behavior of installed PV systems. A critical aspect of this understanding is identifying specific weather characteristics that differentiate between days with high and low solar electricity production or periods with varying degrees of output variability.

Quantifying the variability of a single PV plant or a fleet of plants over a specific area is essential for informed power system operation decisions. For example, understanding variability aids in determining appropriate levels of regulating reserves. Additionally, classifying days based on their variability allows system operators to evaluate the frequency of different types of variability, helping assess the likelihood of challenging conditions arising from variable renewable energy sources [1].

In the context of home micro generation, accurately assessing the feasibility and performance of PV installations is particularly important. This requires evaluating the solar resource at the specific location of the installation. Solar resources are highly dependent on weather conditions and the time period under consideration. Common measures of solar resources include irradiance and insolation. Irradiance, expressed in watts per square meter (W/m^2), measures solar power on a given plane (e.g., horizontal or plane-of-array [POA]) and is directly proportional to the power output of a PV system. Variations in irradiance provide insights into the variability of PV plant output, making it a critical parameter for analysis.

Analyzing day-to-day variations in solar resources for specific locations and time periods offers valuable insights. A precise categorization of days into clear, cloudy, and intermediate conditions is crucial for understanding and predicting solar resource availability. Traditional methods for day classification, such as the one proposed in [2], rely on clustering hourly solar irradiation data collected throughout the day. While effective, these methods require extensive, fine-grained data collection, which can be resource-intensive.

Machine learning (ML) approaches provide an alternative by using daily average features for classification. These methods eliminate the need for detailed hourly measurements, offering a more efficient and scalable solution. In this study, we evaluate the performance of various ML methods for classifying days based on solar variability, comparing their effectiveness against the standard classification method described in [2].

The data used in this analysis are sourced from real-world PV sites, ensuring the applicability and relevance of the findings to practical scenarios.

2 Input data

2.1 What is OSMER

The Regional Meteorological Observatory (OSMER) is a branch of ARPA FVG, the Agency for Environmental Protection of the region Friuli Venezia Giulia (North-Est Italy), and thus it's an operative branch of the regional Administration.

It's activities are observing, understanding and forecasting weather phenomena, diffusing the resulting products (bulletins, warnings, data) to all the productive sectors and to the population. More into detail, OSMER manages several kinds of meteorological data coming from AWS networks both local and from surrounding regions, from weather radars, satellite, radio-soundings and lighting detection systems. Data are used for real time monitoring and for climatology. It also manages a net of hail-pads, expanded also to the Slovenian area near the border.

2.2 Raw data

The raw data correspond to historical weather information hourly measured taken at several different weather monitoring stations across the Friuli Venezia Giulia region in Italy. They are obtained from the OSMER website, and include the following information:

- *location*: Name of the location where the weather station has been installed;
- *year*: Year number of the corresponding data;
- *month*: Month number of the corresponding data;
- *day*: Day number of the corresponding data;
- *hour*: Hour number of teh corresponding data;
- *rain_mm*: Rain precipitation in mm;
- *temperature_C*: Temperature in °C;
- *humidity_perc*: Percentage humidity;
- *min_leaf_wet*: Minimum leaf wetness;
- *pressure_hPa*: Pressure in hPa;
- *avg_wind_kmh*: Average wind in km/h;
- *wind_dir_N*: Wind direction (angle with respect to the North direction);
- *max_wind_kmh*: Maximum wind in km/h;
- *max_wind_dir_N*: Maximum wind direction (angle with respect to the North direction);
- *solar_radiation_kjm2*: Global solar radiation in kJ/m^{-2}

2.3 Preliminary data selection and manipulation

The following steps are applied prior to any data analysis.

2.3.1 Locations and year selection

It has been decided to focus on one specific year of data taking, namely 2012, under the assumptions that the findings (i.e. the correlation between features) is year independent.

Additionally, only three locations has been taken into consideration for the rest of the analysis, each of them representing a different macro-area of the Friuli-Venezia-Giulia area, namely "Capriva del Friuli" (plain area), "Lignano" (shoreline area), and "Tarvisio" (mountain area).

A more refined analysis which take into more than one year of data-taking and different locations will be considered in future studies.

2.3.2 Convert solar radiation unit

The solar irradiance is given, for each hour, in $\mathrm{kJ/m^{-2}}$. In order to be compared with the output of the solar irradiance theoretical model that will be used in the rest of this work it has been converted to $\mathrm{W/m^{-2}}$. The conversion factor is calculated according in the following way: since a J is a W/s:

$$\frac{\mathrm{kJ}}{\mathrm{m}^2} \cdot \mathrm{h} = 1000 \cdot \frac{\mathrm{watt}}{\mathrm{m}^2} \cdot 1/(3600\mathrm{s/h}) \cdot \mathrm{h}$$

Then simplifying:

$$\frac{\mathrm{W}}{\mathrm{m}^2} = \frac{1}{3.6}\frac{\mathrm{kJ}}{\mathrm{m}^2\mathrm{h}}$$

2.3.3 Compute the sunrise and sunset times for the location

When considering solar radiation data taken at multiple locations, an high correlation is expected also because the set of data considers all the data gathered during time (including the night period). The presence of the night period causes a strong contribution to increase the correlation and need to be eliminated in order to achieve better modeling performances. The night period can be eliminated by deleting the corresponding data. To do so, the *sunrise* and *sunset* times are computed for each location and day of the year and added to the raw dataset. Data are then dropped when the correspondig data-taking time is outside the daylight time window. In order to add compute the sunrise and sunset times, the ASTRAL package [3] has been used.

2.3.4 Compute the theoretical solar radiation expected under clear-sky conditions

Finally, in order to been able to categorize the days according to the weather conditions, the measured solar radiation must be compared to the expected one at each location and day of the year considered.

To obtain theoretical predictions of solar radiation, the Ineichen model, a widely-used approach for clear-sky conditions, has been used. This model estimates direct normal irradiance (DNI) and global horizontal irradiance (GHI) based on solar geometry, atmospheric turbidity (Linke turbidity factor), and site altitude. Its parameterized equations efficiently account for key atmospheric attenuation processes, including Rayleigh scattering, aerosol effects, and water vapor absorption. The Ineichen model's computational simplicity and proven accuracy across diverse locations make it a suitable choice for our application, providing reliable baseline estimates for solar irradiance under clear-sky conditions.

The theoretical predictions are obtained with the *ineichen* model as implemented in the PV_LIB toolbox [4], separately for each location and day of the year by using the same granularity of the raw data (i.e. hourly measurements). A representative result, obtained for "Capriva del Friuli", is presented in Fig. 1, which shows the superposition od the solar irradiance plots obtained for each clear sky day on 2012 with PV modules located on the horizzontal plane.

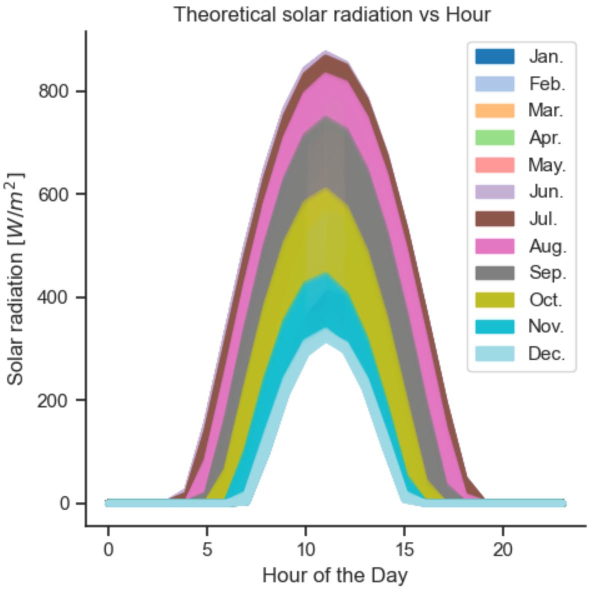

Figure 1: Evolution in time of the solar irradiance in clear sky conditions for "Capriva del Friuli". The theoretical predictions are obtained with the *ineichen* model by using the PV_LIB toolbox [4].

2.4 Classification of days according to measured the solar irradiance

From Fig. 1 it is clear that the clear sky conditions are not represented by the same line trough the year. Instead, two major sources of variations can be observed: one on the horizontal axis, due to the variation of the sunrise and sunset along the year and one on the vertical axis due to the different sun position at noon during the year. To normalize each raw data entry, irrespective of the location and day of the year, a bi-normalization procedure in which both the horizontal and vertical axes are normalized in the $(0, 1)$ range, as proposed in [2], has been used. The binormalised data are then feeded to a clustering procedure in order to split days with similar solar radiation patterns in different categories. On the basis of the clustering results, the attributes associated to the clusters are defined, typically resorting to a terminology that makes it possible to identify for each cluster the characteristics of the days, e.g., clear sky, quasi-clear sky, quasi-cloudy sky and cloudy sky. The starting point for the solar radiation binormalisation is the solar radiation distribution, both measured and expected, for each location and day of the year. An example of the starting distributions is given in Fig. 2 for three representative days in Capriva del Friuli. The solar radiation data binormalisation proceeds as follow:

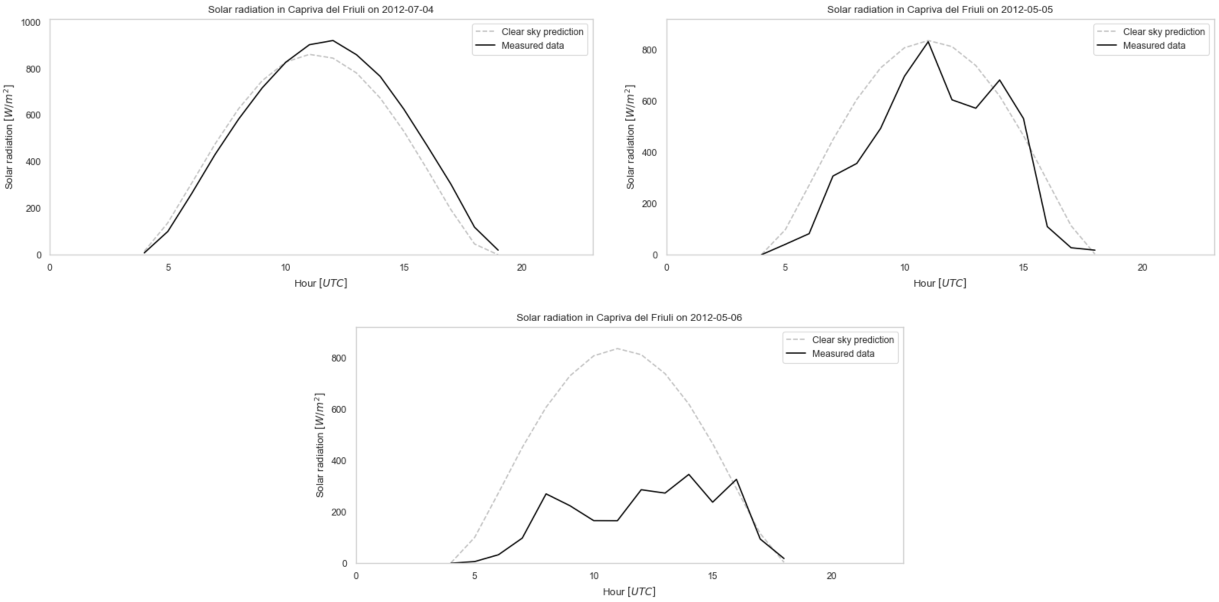

Figure 2: Comparison of measured and predicted solar radiation in Capriva del Friuli for three representative days of 2012. It is clear that the weather conditions were different, in the upper left corresponding to a clear sky day, on the upper right to a sliglty more cloudy one and finally to a cloudy one on the bottom center one.

1. Solar radiation data are divided according to the day and location (cfr. Fig. 2);

2. For each day the measured and theoretical solar radiation are vertically normalised to the maximum predicted value. In this way, for each day the theoretical radiation is now in the range $(0, 1)$ while the measured one can slightly be above 1. This is exepected and can be related to multiple light refractions in the athmosphere;

3. The number of daylight hours is different along the year. To obtain comparable solar radiation patterns the number of points describing the solar radiation has to be the same. In order to to so, a common number of points N has been defined and, for each day, data are interpolated linearly. The same treatment is applied to the measured and predicted solar radiation measurements. The solar radiation patterns are resampled by using $N = 100$. The *numpy.interp* package has been used and the resampled point integral corrected by the γ factor where

$$\gamma = \frac{\int \text{original distribution}}{\int \text{resampled distribution}}$$

. Finally the resampled points are normalised in the $(0, 1)$ interval by using the *sklean.preprocessing.MinMaxScaler* scaler;

Once the solar radiation data has been binormalised the classification proceed as follows. Firstly, the solar radiation patterns are separated in two categories, namely:

- ONP: ordered binormalised patterns, where for each day data are ordered in ascending order with respect to the magnitude of the measured solar radiation;
- OND: ordered binormalised difference patterns, where the absolute difference between the measured and predicted solar radiation is ordered in ascending order;

It has to be noticed that, both ONP and OND patterns completely loose the time information but allows for the clustering model to be agnostic with respect to when, during the day, the clouds appears in the sky (which is considered not to be subject to any specific regularity during the year). The fact that only ONP and OND patterns are considered is motivated by [2] where it has been shown that those are the most informative patterns for a clustering algorithm. Finally, the k-means clustering procedure is used to create the group of days, resulting in K clusters. In this specific case it has been chosen $K = 4$ which corresponds to four day types: *clear*, *quasi-clear*, *quasi-cloudy*, and *cloudy*. After fitting a *sklearn.cluster.KMeans* object to the ONP and OND series, empirical observations on the solar radiation ptterns in the four categories allows to define the following categorization rules:

- A *clear* day is identified by ONP cluster $= 0 \; \vee$ OND cluster $= 0$;
- A *quasi-clear* day is identified by ONP cluster $= 3 \; \vee$ OND cluster $= 2$;
- A *quasi-cloudy* day is identified by ONP cluster $= 2 \; \vee$ OND cluster $= 1$;
- A *cloudy* day is identified by ONP cluster $= 1 \; \vee$ OND cluster $= 3$;

Confidence in the effectiveness of the categorization technique described above can be gained by comparing the solar radiation patterns for the corresponding day categories. In Fig. 2 the upper left figure corresponds to, indeed, a tagged clear day, the upper right figure to a tagged quasi-clear day and the bootom figure to a cloudy one, as expected. The day type categories described above are then added to the raw data. Among the three locations considered, in the year 2012 the followig day type rates are indentified:

- 39% of clear sky days;
- 26% of quasi-clear days;
- 2% of quasi-cloudy day;
- 33% of cloudy days;

3 Machine Learning classification of day type

Once the input raw data are finally created, the subsequent step is trying to understand if a ML approach could be used to categories the day according to the categories defined in the previous section. The advantage of a ML approach can be summarised in the following points:

- Easier implementation, no need to preprocess data (binormalisation);
- No need to collect massive amounts of data: once the model is fitted a daily entry is sufficient to predict the day type;
- The day classification can be done in a more flexible way, accounting for the sensors availabe in the weather station (no need to have a PV unit);

The last point is of particular importance since the main scope of this work is to evaluate if it is possible to train a ML classification algorithm able to predict the day type categories without the information on the solar radiation on the specific location. The idea is to make use only of features correlated to the solar radiation available in the raw data (see Sec. 2.2).

3.1 Exploratory data analysis and data preparation

In order to select the set of features to use in the classification a basic EDA is performed.

The initial dimension of data available is 13482 entries corresponding to 16 features. After checking for the presence of NaN values, several of them (> 8000) has been observed for the *min_leaf_wet*. In order to not heavily limit the data available it has been decided to drop the corresponding feature. The few remaining NaN values correspond to ≈ 200 rows and are smply dropped.

The data are the integrated hourly, such as the final dataframe contains one entry per day and location. All the features values are averaged per day. Finally, since the quasi-cloudy days are very limited $\approx 2\%$, in order not to deal with a to heavily imbalanced multiclass problem they are discarded such as only three data categories remains.

The following features remain available for the classification exercise: *latitude*, *longitude*, *elevation*, *rain_mm*, *temperature_C*, *humidity_perc*, *pressure_hPa*, *avg_wind_kmh*, *wind_dir_N*, *max_wind_kmh*, *max_wind_dir_N*, *solar_radiation*, *month*, and *day*; while the target is represented by the *day_type* string which is in turn one-hot-encoded (by using the *sklearn.preprocessing.label_binarize* object.

The correlation between the varius features and the target is shown in Fig. 3.

The first thing to observe is that the month and day are not particularly correlated with the target thus it has been decided to discard them. This also allow for the model not to be possibly biased by the date when delivering the predictions. Secondly we observe that the solar radiation is highly correlated with the rain, the temperature and the humidity, which is somewhat expected. It is expected that, by not considering the solar radiation as an input feature to the model it should still be able to deliver meaningful day type categorisation. Finally, it can be observed that the pressure is strongly correlated with the latitude, longitude and elevation: it has been decided, despite that, to keep all of these features as input to the model but it can be considered to drop all of them and use only the pressure instead in a future study.

The final set of input features to the classification model is then the following:

- *latitude*;
- *longitude*;
- *elevation*;
- *rain_mm*;
- *temperature_C*;
- *humidity_perc*;
- *pressure_hPa*
- *avg_wind_kmh*;
- *wind_dir_N*;
- *max_wind_kmh*;
- *max_wind_dir_N*;

Most of the features are considerably skewed, so it is expected feature scaling would play a crucial role when applying certain classification models in the rest of this work.

The data are finally split into *train* and *test* sets with a test size of 20%. A stratified splitting is used in order to maintain the different day type proportions among the sets.

3.2 Fitting the models

The train and test sets obtained in Sec. 3.1 are used to train several classification models. Different model where chosen in order to leverage on their characteristics and also to be able to compare their performances. For each model, a *one-vs-rest* classification method is used. The model hyperparameters (and, when needed by the model, the features scaling) has been optimized with Cross-Validation by using 5 Cross-Validation sets and the *f1_weighted* score: using the *f1-weighted* score is advantageous because it accounts for both precision and recall across all classes, while also considering the class imbalance. All the models considered are included in the SCIKIT-LEARN package [5].

3.2.1 SVC model

The first classification model considered is a Support Vector Classification model. The following parameters are tuned via Cross-Validation:

- *C*: Regularization parameter;
- *kernel*: The kernel type;

while the *class_weight* parameter has been set to *balanced.* A pipeline object is created, one for each kind of scalers considered, namely:

- *None*: No feature scaling is applied;
- *StandardScaler*: default standard scaler;
- *MinMaxScaler*: default min-max scaler;
- *RobustScaler*: default robust scaler;
- *QuantileTrasnformer*: quantile transformer with an uniform distribution as output;
- *QuantileTrasnformer*: quantile transformer with a normal distribution as output;

From the Cross-Validation, the following hyperparameters maximise the model performances:

- Best Scaler: quantile transoformer with an uniform distribution as output
- Best Parameters: $C = 1$, *kernel=poly*

3.2.2 RandomForestClassifier model

In order to test how much the need to heavily scale some of the features could impact tje classification, a model which is scale-invariant is used. The following parameters of a *RandomForestClassifier* are tuned via Cross-Validation:

- *n_estimators*: The number of trees in the forest;
- *max_depth*: The maximum depth of the tree;
- *min_samples_split*: The minimum number of samples required to split an internal node;
- *min_samples_leaf*: The minimum number of samples required to be at a leaf node;

while the *max_features* and the *class_weight* parameters are set to *None* and *balanced_subsample*, respectively. From the Cross-Validation, the following hyperparameters maximise the model performances: *max_depth*= 20, *min_samples_leaf*= 10, *min_samples_split*= 2, and *n_estimators*= 2000.

3.2.3 KNN model

Finally, a clustering with very simple interpretability and resilient to new data as much as possible is used. The following parameters of a *KNeighborsClassifier* are tuned via Cross-Validation:

- *n_neighbors*: Number of neighbors to use by default for kneighbors queries;
- *weights*: Weight function used in prediction;

- p: Power parameter for the Minkowski metric. When $p = 1$, this is equivalent to using manhattan distance (l1), and euclidean distance (l2) for $p = 2$;

while the *algorithm* and the *metric* parameters have been set to *auto* and *minkowsky*, respectively. Again, a pipeline object is created, one for each kind of scalers considered, namely:

- *None*: No feature scaling is applied;
- *StandardScaler*: default standard scaler;
- *MinMaxScaler*: default min-max scaler;
- *RobustScaler*: default robust scaler;
- *QuantileTrasnformer*: quantile transformer with an uniform distribution as output;
- *QuantileTrasnformer*: quantile transformer with a normal distribution as output;

From the Cross-Validation, the following hyperparameters maximise the model performances:

- Best Scaler: *MinMaxScaler*;
- Best Parameters: *n_neighbors*$= 10$, $p = 2$, *weights*=*distance*;

3.3 Models results and discussion

The model performances are evaluated, separately for each model, on the test set.

3.3.1 Considerations from the model classification reports

In this section classification reports which summarize the performance of the three different models (SVC, Random Forest, and KNN) on the dataset are compared across different models and classes in the dataset. The precision, recall and f1-score of the models are given in Tab. 1. The following observations can be made based on Tab. 1. The SVC classifier shows:

- Class *clear*: high recall and acceptable precision, which suggest that most (90%) of the clear days are identified correctly at the expenses of ≈ 30% false positives;
- Class *cloudy*: slightly less recall but higher precision, which again suggest the model is able to identify with good precision cloudy days and separate them from the other types;
- Class *quasi-clear*: despite showing a fairly good recall, with 70% of the actual quasi-clear days identified, the precision is not satisfactory, which indicates almost half of the days are categorized in this class despite not being part of it;

	class	precision	recall	f1-score	support
SVC	*clear*	0.72	0.90	0.80	84
	cloudy	0.77	0.83	0.80	72
	quasi-clear	0.50	0.70	0.58	57
	weighted avg.	0.68	0.83	0.74	213
RandomForestClassifier	*clear*	0.70	0.82	0.76	84
	cloudy	0.80	0.76	0.78	72
	quasi-clear	0.46	0.46	0.46	57
	weighted avg.	0.67	0.70	0.68	213
KNN	*clear*	0.73	0.77	0.75	84
	cloudy	0.84	0.67	0.74	72
	quasi-clear	0.48	0.28	0.36	57
	weighted avg.	0.70	0.61	0.64	213

Table 1: The precision, recall, and f1-score for each model considered evaluated for each class on the test dataset.

In summary the SVC perform well overall, especially in identifying and distinguish clear and cloudy days, as can be also seen from the weighted average of the three classes, while it struggles significantly in correctly cataloging the quasi-clear days.

The RandomForestClassifier performances are overall worser than the SVC ones, with the only exception being a sligtly higher precision for identifying cloudy days. It however shows unsatisfactory performances for the quasi-clear days classification, being basically unable to classify them (precision, recall and f1-score are all $< 50\%$). The latter is reflected in the weighted average results which are glovbally worser than the SVC classifier.

Finally, similar considerations can be given for the KNN clasifier, again sligty more precise than the SVC one in identifying cloudy days but even worser than the RandomForestClassifier method when dealing with quasi-clear days, showing a recall of 28%. The KNN method shows the weakest overall performances when considering average between all the classes.

In summary, the relevant takeaways from the classification reports given in Tab. 1 are the following:

- the SVC perform the best overall, with higer recall and f1-scores for class *clear* and *cloudy* while showing some deficits when **quasi-clear** days are classified;

- the RandomForestClassifier provides a still balanced but sligtly weaker perfromance than the SVC, in particular for the *quasi-clear* class;

- finally, the KNN has a strong precision for the *cloudy* class but it struggles with recall for all classes and shows the worst performances for identifying *quasi-clear* days;

To complete the model classification metrics analysis the ROC curves and AUC scores for the three model considered are presented in Fig. 4. From the ROC curve inspection similar consideration can be drawn as before, i.e. the best model being the SVC followed by the RandomForestClassifier and the KNN. When focusing on the ROC curves for the SVC model a steep increase of the true positive rate (TPR) when the false positive rate (FPR) increses can be seen, followed by a plateaux region between $0.3 < \text{FPR} < 0.5$, especcialy for the *quasi-clear* class.

3.3.2 Observed versus predicted values and the confusion matrices

The observed and predicted day types are compared, for each location and model, in Fig. 5. The predicted results are also given in the form of confusion matrices, separately per each location and classification model, in Fig. 6. From Fig. 6 it can be immediately noticed that the same qualitatively behaviour between the different models is similar across every location considered (that is the SVC model is the best classifier among all). When focussing on a single location, for example Capriva del Friuli, the issue with the *quasi-clear* class is evident, that is every model tend to confuse observed *quasi-clear* days with *clear* ones. This is somewhat expected, being the latter expected to be similar but could also point to an imprecise classification according to the binormalised solar radiation pattern (see Sec. 2.4).

3.3.3 Features importance

Finally, for each model, the feature importances and SHAP values were computed separately.

Feature importances were determined using the permutation importance approach and averaged across the three classes considered. The results are shown in Fig.7. From Fig.7, it can be observed that, for all models, the most informative feature is humidity. This finding aligns with the correlation plot in Fig. 3, where it is evident from the EDA that humidity has the strongest correlation with the day type.

The same observation applies to the second most important features: rain (for the SVC model) and temperature (for the RandomForestClassifier and KNN models). In this case, the correlation is less apparent, as these features seem nearly uncorrelated with the day type (see Fig.3). However, they exhibit a strong correlation with the solar radiation variable, which, while not used as an input feature for any model, was employed in Sec.2.4 to cluster the days.

It is worth noting that, for most of the highly ranked features, the boxplots display considerable variability. This suggests potential interactions between features and highlights the need for improved feature selection, such as removing potentially uninformative or correlated features.

To address the assumption of feature independence inherent in the permutation importance algorithm, SHAP values were also computed and are presented in Fig. 8. In this case, SHAP values were computed by merging all locations but separating the different day type classes. The discussion focuses on the SVC model, as it showed the best performance.

Examining the SHAP values, humidity remains the most important feature for identifying *clear* and *cloudy* days. However, wind-related features are more effective in identifying *quasi-clear* days, likely because the presence of wind often indicates changing weather conditions.

For both the RandomForestClassifier and KNN models, wind plays a less significant role in classification. This could explain the comparatively poorer performance of these models in correctly identifying *quasi-clear* days compared to the SVC model.

3.4 Summary

In this study, weather data from the Friuli Venezia Giulia Regional Meteorological Observatory were analyzed. Specifically, a traditional method based on the binormalized daily solar radiation patterns—both measured and predicted—was employed to classify days from 2012 into distinct day-type categories across three locations.

Subsequently, machine learning (ML) approaches were explored as an alternative, using daily averaged features for classification. These ML methods eliminate the need for detailed hourly measurements, offering a more efficient and scalable solution. Among the classification models tested, the Support Vector Classifier (SVC) demonstrated the best performance, particularly in distinguishing between clear and cloudy days.

However, the identification of the *quasi-clear* day class proved challenging for all models. This difficulty is likely due to its intermediate characteristics, which overlap with those of the clear and cloudy classes. Additionally, this may stem from issues such as correlated features in the dataset or potential inaccuracies in the initial classification using the binormalized data. These findings suggest that the traditional binormalization-based method may have limitations, highlighting the potential advantages of ML-based approaches for future applications.

References

[1] Chris Trueblood, Steven Coley, Tom Key, Lindsey Rogers, Abraham Ellis, Cliff Hansen, and Elizabeth Philpot. Pv measures up for fleet duty : Data from a tennessee plant are used to illustrate metrics that characterize plant performance. *IEEE Power and Energy Magazine*, 11(2):33–44, 2013.

[2] Gianfranco Chicco, Valeria Cocina, and Filippo Spertino. Characterization of solar irradiance profiles for photovoltaic system studies through data rescaling in time and amplitude. In *2014 49th International Universities Power Engineering Conference (UPEC)*, pages 1–6, 2014.

[3] Ben Major. Astral: A lightweight python library for calculations involving the position of the sun and moon. https://sffjunkie.github.io/astral/, 2024. Accessed: 2024-11-29.

[4] Kevin S. Anderson, Clifford W. Hansen, William F. Holmgren, Adam R. Jensen, Mark A. Mikofski, and Anton Driesse. pvlib python: 2023 project update. *Journal of Open Source Software*, 8(92):5994, 2023.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

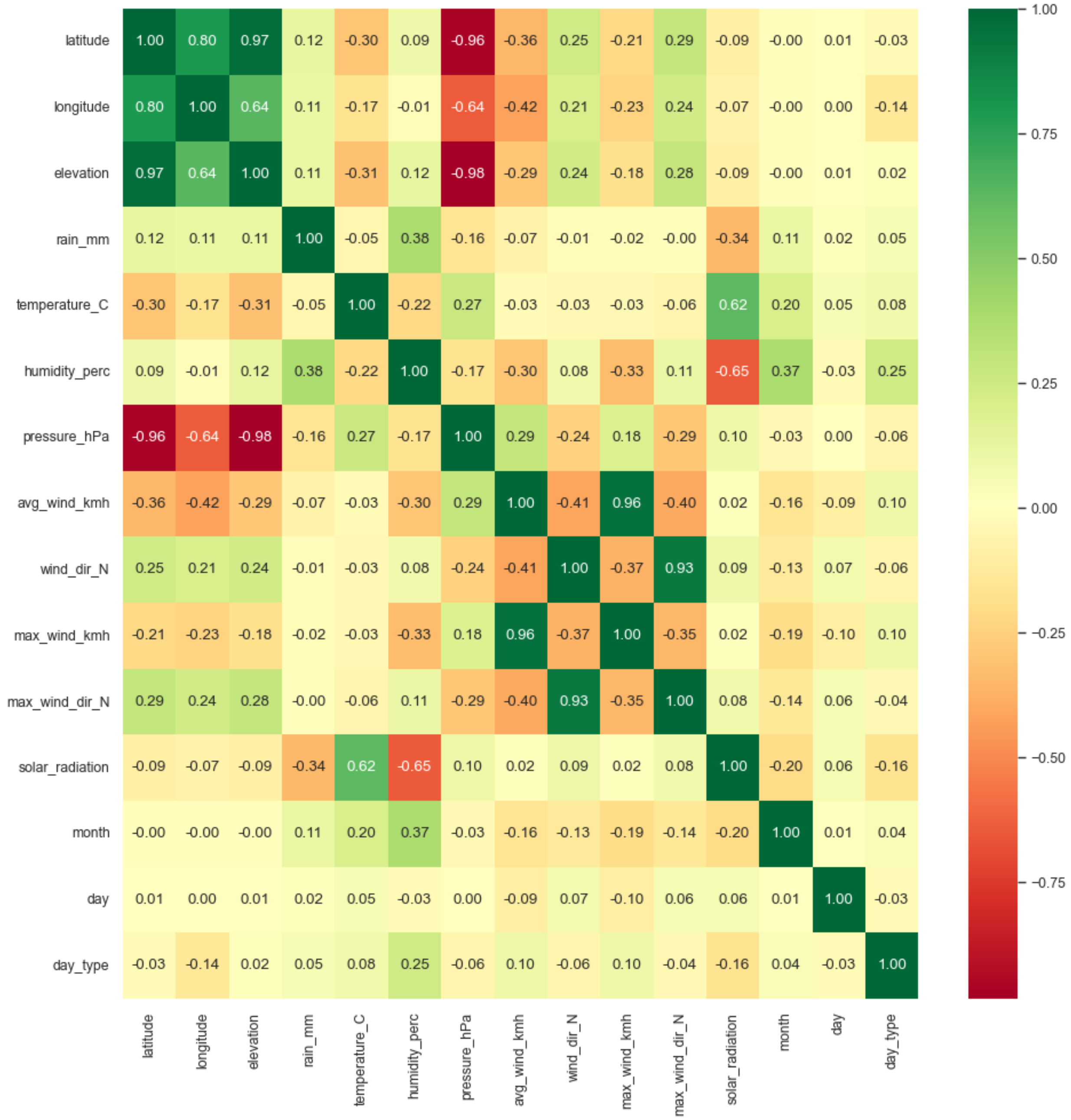

Figure 3: The correlation between all the remaining input features and the day type.

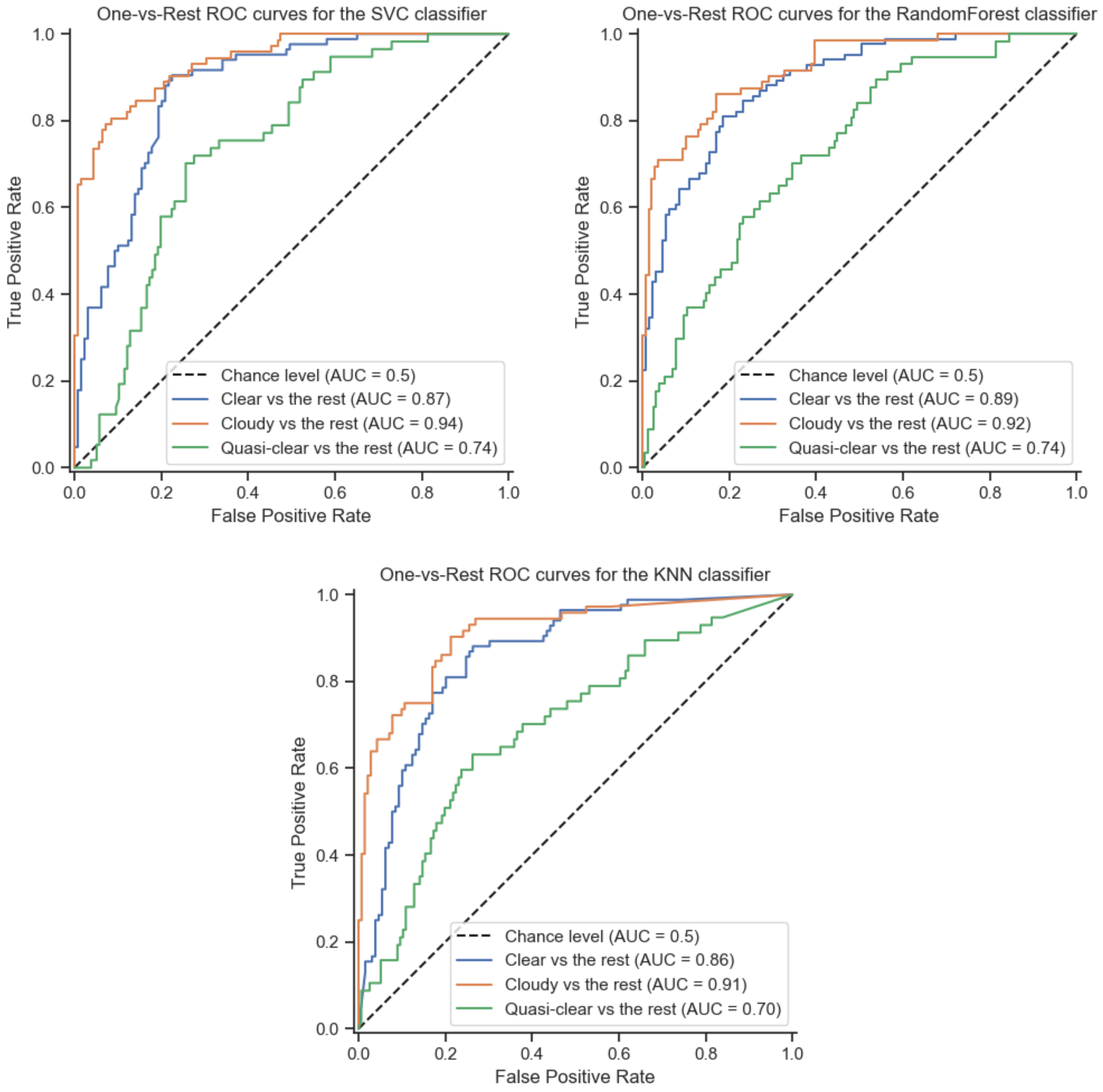

Figure 4: The ROC curves (and correspoding AUCs) for the three classes considered. The results are given for the SVC classifier (upper left), the RandomForestClassifier (upper right) and the KNN classifier (bottom center).

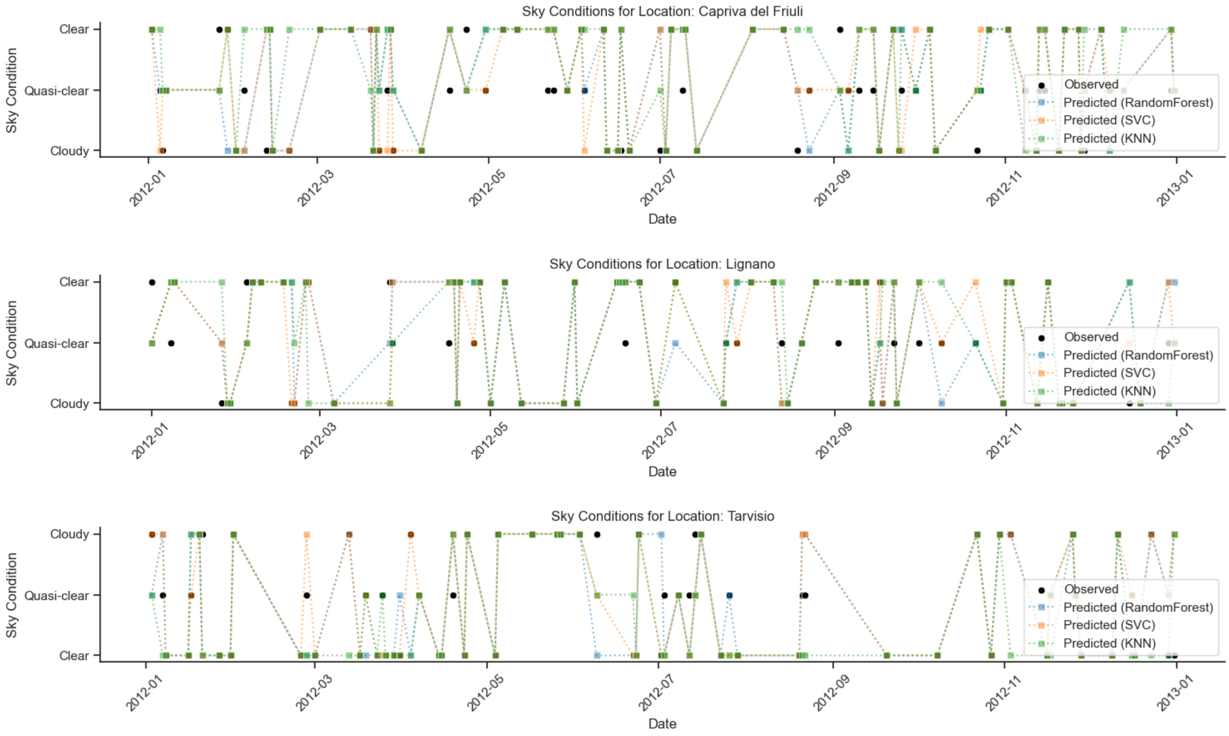

Figure 5: The observed and predicted day types for the three locations considered. Results are given separately for each model.

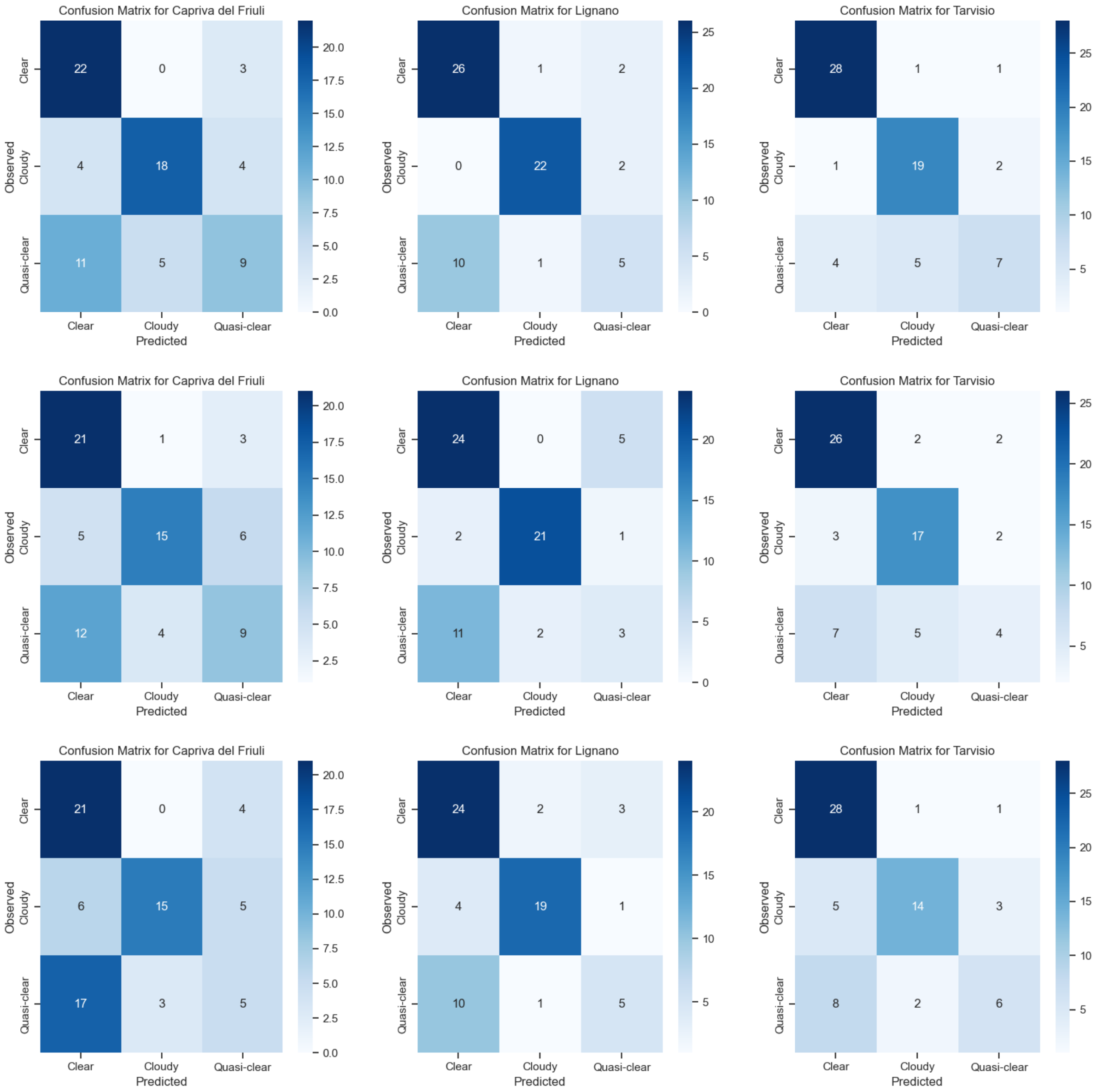

Figure 6: The confusion matrices for the three models considered SVC (up row), RandomForestClassifier (mid row), and the KNN (bottom row). Results are given separately for each location considered, namely Capriva del Friuli (left), Lignano (center), and Tarvisio (right).

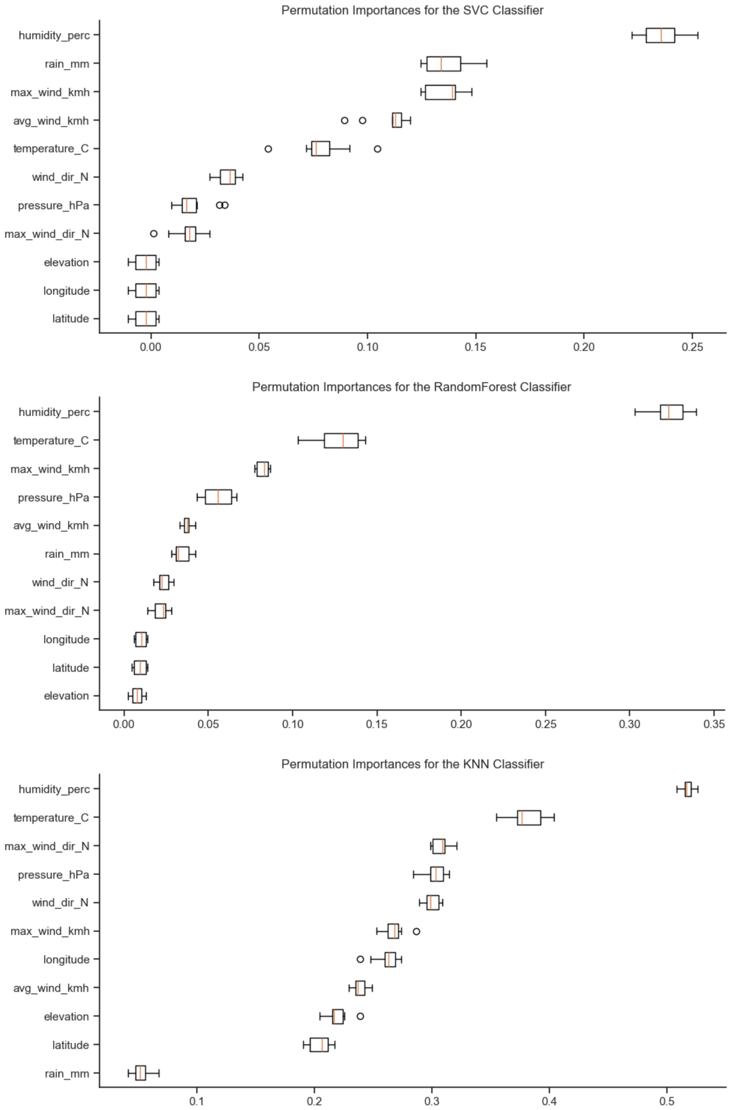

Figure 7: Feature importances for the three classifiers considered: the SVC (top row), the RandomForestClassifier (middle row), and the KNN classifier (bottom row). The importances, averaged across all classes, are obtained via the permutation importance algorithm.

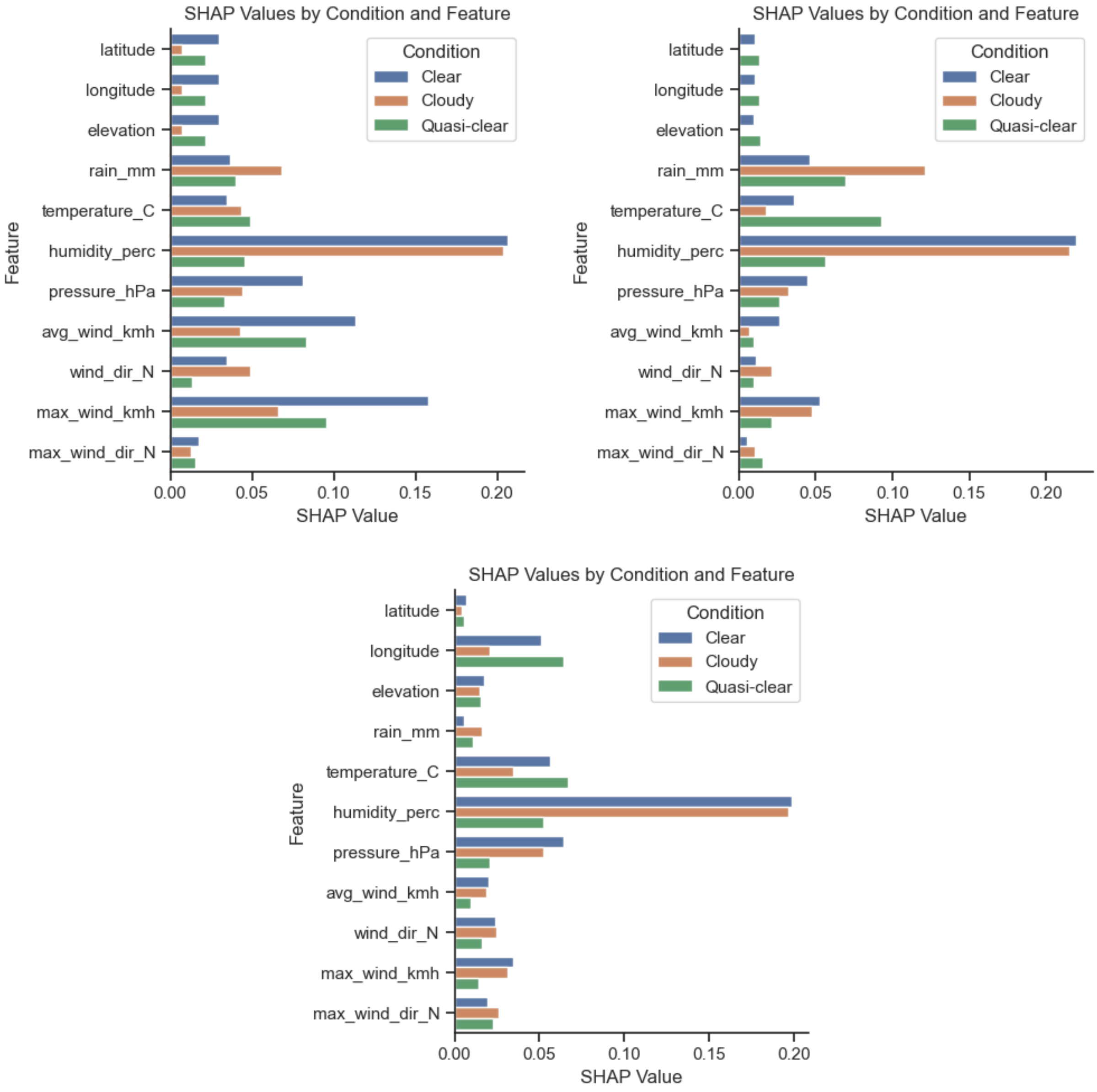

Figure 8: SHAP values for the SVC (upper left), RandomForestClassifier (upper right), and KNN model (bottom). Separate SHAP values were computed for each day type class.