Prediction of Dew and Frost Points Over One Year for Vineyard Protection

Henri Dargent et Paul-Antoine Cousseau
Engineering Student – Data and Artificial Intelligence
ESILV

December 12, 2025

Abstract

Viticulture is highly sensitive to frost events occurring during the early growth stages of the vine. In this project, we use historical meteorological observations and machine learning techniques to forecast daily dew and frost points over one year. The objective is to provide winegrowers with an operational tool to anticipate days with a high risk of frost and prepare adequate protection strategies (e.g., heating, spraying, or crop covers).

The work is strongly connected to my specialization in Data and Artificial Intelligence. It combines classical time series modelling (SARIMAX) with modern machine learning models (Support Vector Regression, gradient boosting with LightGBM and XGBoost, and ensembles). After a broad model exploration, we show that models not explicitly designed for temporal data tend to overfit, whereas a well-specified SARIMAX model offers a robust compromise between accuracy, interpretability and temporal consistency.

The report describes the dataset, the feature engineering process, the formalisation of the prediction problem, the different models tested, the difficulties encountered (in particular overfitting and the handling of seasonality), and a comparison of the final results.

Contents

1 Business Case

Late frost is a major threat for vineyards, especially in regions where winter temperatures can still drop below freezing during the beginning of the growing season. When the temperature falls below the frost point, crystallization occurs and young buds and leaves can be severely damaged, leading to important yield losses and economic impacts for winegrowers.

Figure 1: Image of a frozen bud

The business objective of this project is therefore:

> **To predict daily frost point over a one-year horizon in order to identify dates with a high risk of frost, so that winegrowers can anticipate protection measures.**

From a Data and Artificial Intelligence perspective, this problem is interesting because:

- It is inherently **temporal**, with strong **seasonality** and **autocorrelation** structures.
- It requires combining **physical understanding** (relationship between temperature, humidity, dew point and frost) with **statistical learning**.
- It provides an opportunity to compare **time series models** (SARIMAX) and **generic machine learning models** (SVR, gradient boosting, ensembles) on the same task.

The final goal is not only to obtain good numerical performance, but also to design a model that remains stable over time and can realistically be deployed to support decision-making in vineyards.

2 Dataset Description

2.1 Source and Scope

The dataset used in this project consists of daily meteorological observations spanning from January 1, 2009 to July 28, 2020. The data contains 3,902 daily records with 23 features capturing

various atmospheric conditions. The dataset provides comprehensive measurements of temperature, humidity, pressure, wind, and precipitation. The dataset was downloaded from Open Data Bay, and its geographical scope is Estespark, a city from Colorado, USA. The dataset used in this project is a historical daily weather dataset. It contains meteorological measurements aggregated per day.

2.2 Raw Variables

The raw dataset contains the following main daily variables:

- **Maximum temperature** (°F)
- **Minimum temperature** (°F)
- **Average temperature** (°F)
- **Maximum humidity** (%)
- **Minimum humidity** (%)
- **Average humidity** (%)
- **Average windspeed** (mph)
- **Average barometer** (inches)
- **Average dewpoint** (°F)
- **Rainfall for day / month / year** (inches)
- **Average wind direction** (degrees)
- **Date** (used as time index)

The dew point point and temperature are closely related to frost point: frost occurs when the dew point and the temperature are close to 0 °C, and when the temperature decreases below the frost point. Predicting the dew point and the average temperature is therefore directly relevant for frost risk assessment.

2.3 Data Quality

The dataset exhibits excellent quality with no missing values across all 3,902 records and 23 features. This completeness eliminates the need for imputation strategies and ensures that all available information can be used for model training. However a total of 325 days are missing from the dataset, meaning new rows need to be added and features filled with the proper method to retain maximum information for the model prediction.

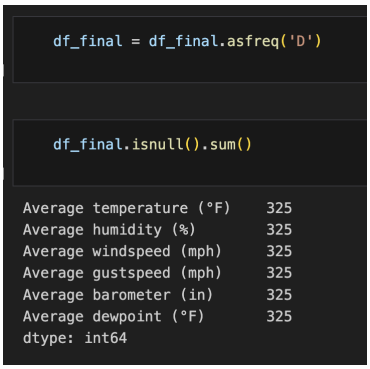

Figure 2: The days that didn't appear in the dataset

2.4 Engineered Features

To improve the predictive power of the models, several additional features were constructed:

- **Temperature range**:

$$\texttt{diff\_temperature} = \texttt{Maximum temperature} - \texttt{Minimum temperature}.$$

  This captures daily thermal amplitude and provides additional information beyond a simple average.

- **Humidity range**:

$$\texttt{diff\_humidity} = \texttt{Maximum humidity} - \texttt{Minimum humidity}.$$

  This describes the variability of humidity over the day.

- **Cyclical encoding of the day of year**: if d denotes the day of year (from 1 to 365), we define:

$$\texttt{day\_sin} = \sin\left(2\pi\frac{d}{365}\right), \qquad \texttt{day\_cos} = \cos\left(2\pi\frac{d}{365}\right).$$

  This encoding preserves the cyclical nature of the calendar: day 1 and day 365 are close in the encoded space, which is not the case if we only use the raw integer index.

These engineered features were motivated by exploratory data analysis and by domain knowledge: the thermal and humidity ranges summarise intra-day variability, and the sine/cosine transform allows models to capture seasonality smoothly.

2.5 Final Feature Set

After cleaning and initial correlation analysis, the following features were kept for modelling the frost point:

- Average humidity (%)
- Average windspeed (mph)
- Average barometer (in)
- Average dewpoint (°F)

- `diff_temperature`
- `diff_humidity`
- `day_sin`
- `day_cos`

Some features such as yearly rainfall were discarded because their information was already encoded in the seasonal variables (`day_sin`, `day_cos`) and their correlation analysis suggested redundancy.

3 Exploratory Data Analysis

In this section we summarise the main exploratory analyses that were carried out to understand the dataset before modelling.

3.1 Time Series Visualisation

The first step consisted in plotting the raw and smoothed time series of the target variable (dew-point or temperature) together with some key meteorological features such as average barometer and humidity.

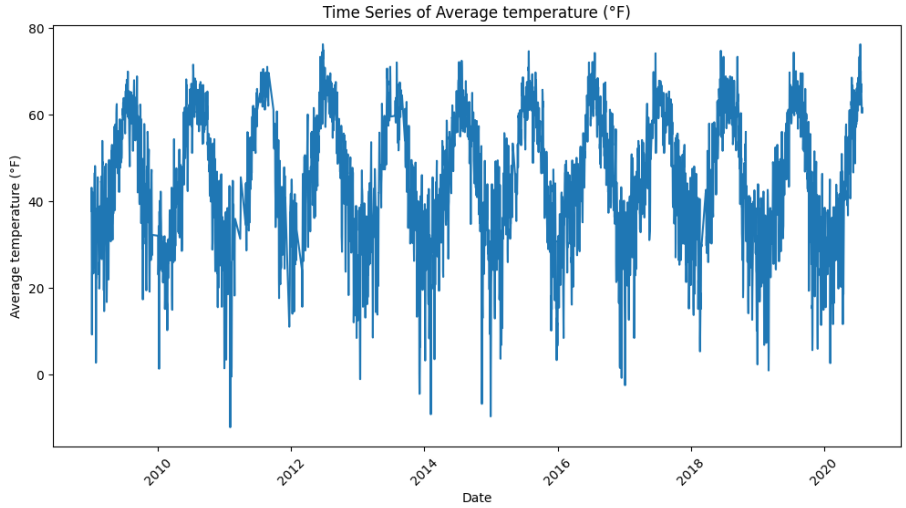

Figure 3: Example of daily Temperature over several years with visible seasonality.

The plot clearly shows yearly seasonality: temperature points are high in summer and low in winter, with pronounced cycles corresponding to the annual climatic rhythm. This observation justifies the use of sinusoidal features (`day_sin`, `day_cos`) and seasonal time series models.

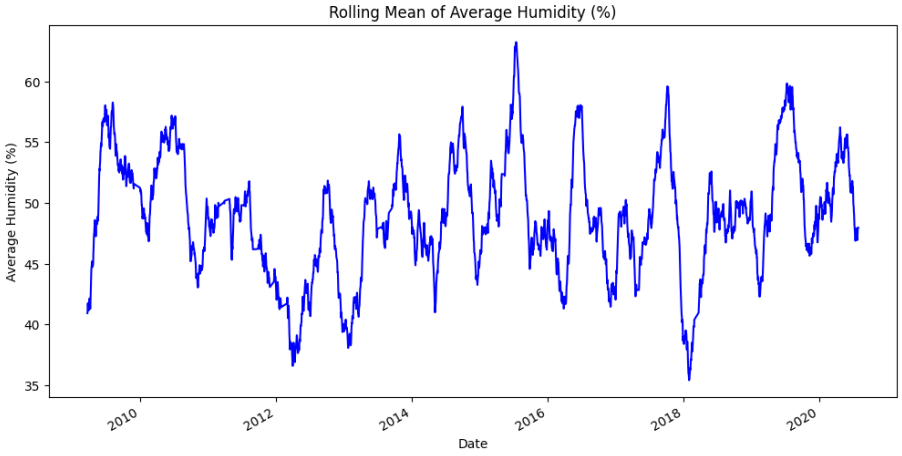

Figure 4: Use of rolling mean strategy to showcase better the trend

3.2 Correlation Analysis

We computed Pearson correlation coefficients between the target and the candidate features. For instance, the day-of-year encoding showed a moderate to strong correlation with temperature-related variables, confirming the importance of seasonality.

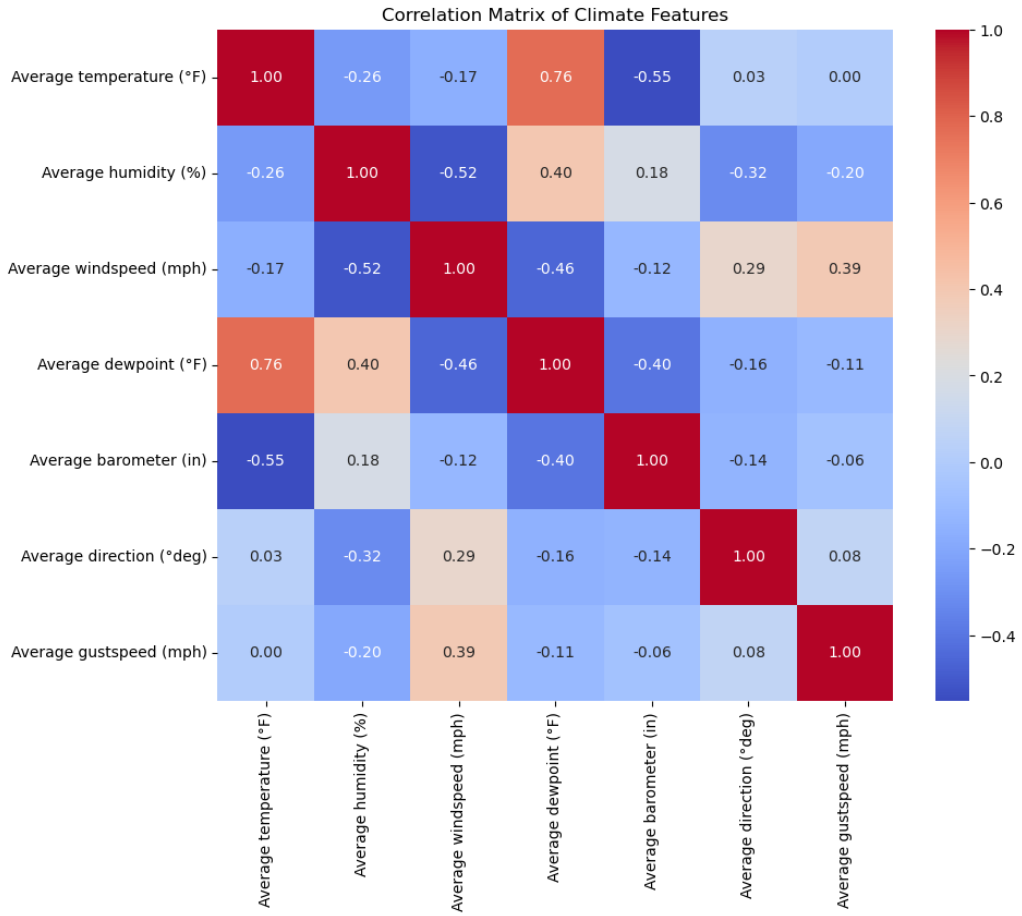

Figure 5: Correlation matrix between the target and selected meteorological features.

Some key observations:

- Maximum and minimum temperature are highly correlated with each other (correlation above 0.9), which suggests that using both brings almost the same information and may create multicollinearity.
- The derived feature `diff_temperature` has a lower but still meaningful correlation with the dew/frost point, and summarises daily thermal amplitude, which is physically relevant.
- Yearly rainfall is strongly linked to the time of year. Since seasonality is already captured by `day_sin` and `day_cos`, keeping rainfall for the year would mainly duplicate seasonal information, so it was removed from the final feature set.

We also plotted each variable we were looking to analyze with the target one. A simple but efficient method that can help to witness if a correlation or trend happens between the two

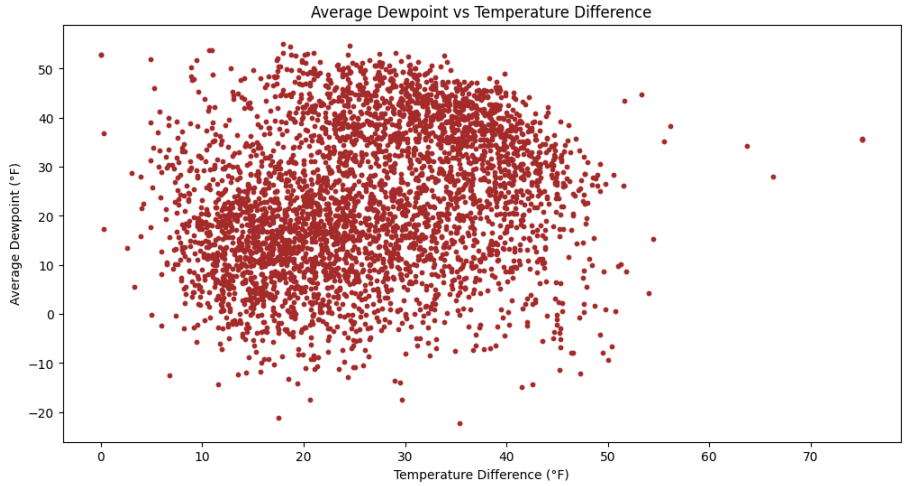

Figure 6: Scatter plot of the dew point and temperature difference.

The temperature difference is a new feature that was created using the "maximum temperature" and "minimum temperature" features. From the scatter plot, we can clearly see there is no correlation or trend that stands out, just a big regrouping of point that doesn't hold any meaningful structure.

3.3 Distribution of Frost Days

To connect the project to the business case, frost days (when the frost point is higger than the temperature) were identified, and their distribution across the calendar year was analysed.

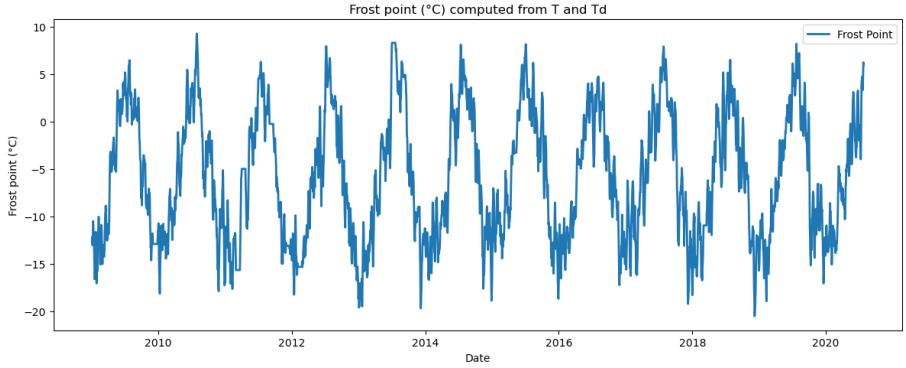

Figure 7: Histogram of frost point.

Most frost events occur during winter and early spring, which is consistent with domain knowledge. The vine is particularly sensitive during the transition from dormancy to budburst, so early spring frosts are of highest concern for vineyard managers.

3.4 Seasonal Patterns

The data exhibits strong seasonal patterns consistent with a temperate climate. Monthly average temperatures show clear variation throughout the year:

- **Winter months** (December-February): 28-30°F average
- **Spring months** (March-May): 36-48°F average
- **Summer months** (June-August): 60-64°F average
- **Fall months** (September-November): 36-56°F average

The coldest month is February (28.50°F average) while the warmest is July (63.78°F average). This 35°F seasonal range indicates a continental climate with significant winter-summer temperature differences.

Precipitation patterns also show seasonal variation, with the highest rainfall occurring in spring and summer months (May and July), while winter months typically receive less precipitation.

4 Problem Formalisation

4.1 Prediction Target

Let y_t denote the dew point or the average temperature at day t. Depending on the model and experiment, y_t may represent:

- the daily average dew point,
- the daily average temperature.

The main task is to predict y_t based on past meteorological observations and exogenous features.

4.2 Supervised Learning Setting

We consider a supervised regression setting. For each day t we observe a feature vector x_t:

$x_t = (\text{Average humidity}_t, \text{Average windspeed}_t, \text{Average barometer}_t, \texttt{diff\_temperature}_t, \texttt{diff\_humidity}_t, \texttt{day\_s}$

and the corresponding target y_t. The goal is to learn a function f such that

$$\hat{y}_t = f(x_t, \text{history up to } t-1),$$

and $\hat{y}_t$ is as close as possible to y_t.

4.3 Evaluation Metrics

We evaluate models using three standard regression metrics:

- Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{N}\sum_{t=1}^{N}|y_t - \hat{y}_t|.$$

- Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}.$$

- Coefficient of determination (R^2):

$$R^2 = 1 - \frac{\sum_t (y_t - \hat{y}_t)^2}{\sum_t (y_t - \bar{y})^2}.$$

Lower MAE and RMSE indicate better predictive accuracy, while R^2 values closer to 1 indicate that a large proportion of the variance is explained by the model.

5 Models

5.1 Baseline Model: Linear Regression

Linear regression serves as our baseline model, assuming a linear relationship between features and temperature:

$$y = \beta_0 + \sum_{j=1}^{d}\beta_j x_j + \epsilon$$

where β_j are learned coefficients and ϵ represents residual error.

Advantages:

- Simple and interpretable
- Fast training and prediction
- Provides insight into feature importance through coefficients
- We've already been dealing with this kind of model for a long time

Limitations:

- Assumes linear relationships
- Cannot capture complex interactions between features
- Sensitive to outliers
- Isn't too adapted to time series prediction

5.2 SARIMAX: A Time Series Baseline

The Seasonal AutoRegressive Integrated Moving Average with eXogenous variables (SARIMAX) model explicitly accounts for temporal dependencies and seasonality. It is well suited for our problem because average temperature and dew point are strongly seasonal and autocorrelated.

A SARIMAX$(p, d, q)(P, D, Q)_s$ model can be written as:

$$\Phi_p(L)\Phi_P(L^s)(1-L)^d(1-L^s)^D y_t = \Theta_q(L)\Theta_Q(L^s)\varepsilon_t + \beta^\top x_t,$$

where:

- L is the lag operator ($Ly_t = y_{t-1}$),
- (p, d, q) describe the non-seasonal autoregressive, integration and moving average orders,
- (P, D, Q) describe the seasonal part with period s (here $s = 365$ for yearly seasonality),
- x_t is the vector of exogenous regressors,
- ε_t is a white noise error term.

In practice, we trained a SARIMAX model on the training period using the dew point and average temperature as target and the engineered features as exogenous variables. The time index is the calendar date, and a chronological split (80% train / 20% test) was used to mimic realistic forecasting conditions.

5.3 Machine Learning Models

In order to compare SARIMAX with more flexible non-linear models, we trained several machine learning regressors on the same task. Because these models do not internally model temporal dependence, the time dimension is only provided through the exogenous features (especially `day_sin`, `day_cos`).

5.3.1 Support Vector Regression (SVR)

Support Vector Regression seeks a function that deviates from the observed targets by at most an ϵ margin while being as flat as possible. In this project we used an RBF kernel and optimised hyperparameters (such as C, ϵ and the kernel width) via grid search on the training set.

5.3.2 Gradient Boosting: LightGBM and XGBoost

Gradient boosting builds an ensemble of decision trees sequentially, each new tree correcting the errors of the previous ensemble. We used two popular implementations:

- **LightGBM**, which is efficient and supports advanced histogram-based splitting.
- **XGBoost**, another robust and widely-used implementation of gradient boosting.

Hyperparameters such as learning rate, maximum depth, number of estimators and regularisation terms were tuned through cross-validation on the training set.

5.3.3 Ensemble Models

To further exploit model diversity, we also combined SVR, LightGBM and XGBoost through simple averaging ensembles:

- An ensemble combining SVR + LightGBM + XGBoost.
- An ensemble combining only SVR + XGBoost.

The idea is that different models may capture different aspects of the relationship between meteorological variables and dew/frost points, and averaging their predictions may reduce variance.

6 Training Procedure and Addressing Obstacles/Challenges

6.1 Train–Test Split and Temporal Consistency

A crucial aspect of this project was to avoid information leakage from the future into the past. Therefore, we used a chronological split: the first 80% of the time series was used for training, and the remaining 20% for testing. This is more realistic than a random split for time series data.

6.2 Data Preprocessing

Missing values were handled by forward filling when necessary (for instance, for some meteorological variables). Features used by machine learning models were standardised or scaled when appropriate (SVR is sensitive to feature scaling).

6.3 Hyperparameter Optimisation

For all machine learning models (SVR, LightGBM, XGBoost), we performed hyperparameter tuning using grid search with cross-validation on the training set. The search focused on parameters that have a strong impact on overfitting, such as regularisation coefficients, maximum tree depth and learning rate.

6.4 Overfitting and Underfitting Prevention

One of the main obstacles encountered was overfitting for flexible models such as gradient boosting. To diagnose it, we systematically compared performance on the training and test sets.

For example, for the SVR model we obtained the following overfitting test:

- Train MAE: 0.2786; Test MAE: 0.4541
- Train RMSE: 0.4752; Test RMSE: 0.6916

The difference is moderate, and the interpretation was “little overfitting”. On the contrary, for LightGBM and XGBoost we observed:

- LightGBM – Train MAE: 0.3386; Test MAE: 0.7802; Train RMSE: 0.4438; Test RMSE: 1.0931 (clear overfitting).

- XGBoost – Train MAE: 0.4834; Test MAE: 0.7700; Train RMSE: 0.6272; Test RMSE: 1.0448 (overfitting).

Despite tuning regularisation and limiting model complexity, these boosted tree models remained quite prone to overfitting on this relatively smooth time series.

By contrast, SARIMAX, which is constrained by its parametric structure and directly models autocorrelation, showed much more stable behaviour.

6.5 Choice of the Final Model

Even though some machine learning models achieved very low test errors (especially the ensembles combining SVR and XGBoost), their tendency to overfit and the lack of explicit temporal structure raised concerns about their robustness when deployed on future unseen years.

For this reason, SARIMAX was selected as the **main model** for the final frost point forecasting system. The machine learning models are still useful as benchmarks and to show the potential of more complex approaches, but the time series model is more aligned with the physics of the problem and easier to interpret by domain experts.

7 Results and Model Comparison

7.1 Performance of Individual Models

Table 1 summarises the main performance metrics obtained on the test set for the different models.

Table 1: Test set performance of the different models for dew/frost point prediction.

Model	**MAE**	**RMSE**	$\boldsymbol{R^2}$
SARIMAX (temperature target)	1.8207	2.5117	0.9749
SARIMAX (dew/frost point target)	1.8433	2.5630	0.9677
SVR (optimised)	0.5627	0.8830	0.9967
LightGBM (optimised)	0.7802	1.0931	0.9949
XGBoost (optimised)	0.7700	1.0448	0.9953
Ensemble (SVR + LightGBM + XGBoost)	0.5656	0.7959	0.9973
Ensemble (SVR + XGBoost)	0.5372	0.7574	0.9976

At first sight, the ensembles reach the best numerical performance with very low MAE and RMSE and R^2 close to 0.998. However, these values must be interpreted with caution, because the models are trained on a relatively smooth time series and may overfit specific patterns of the training period.

The SARIMAX models achieve slightly higher errors, but still reach R^2 above 0.96, which is very satisfactory for a meteorological application. Moreover, their predictions respect the temporal structure of the data (seasonality and smoothness) and are easier to explain.

7.2 Visual Comparison of Predictions

To complement the numerical metrics, we plotted the predicted and true dew point for a subset of the test period.

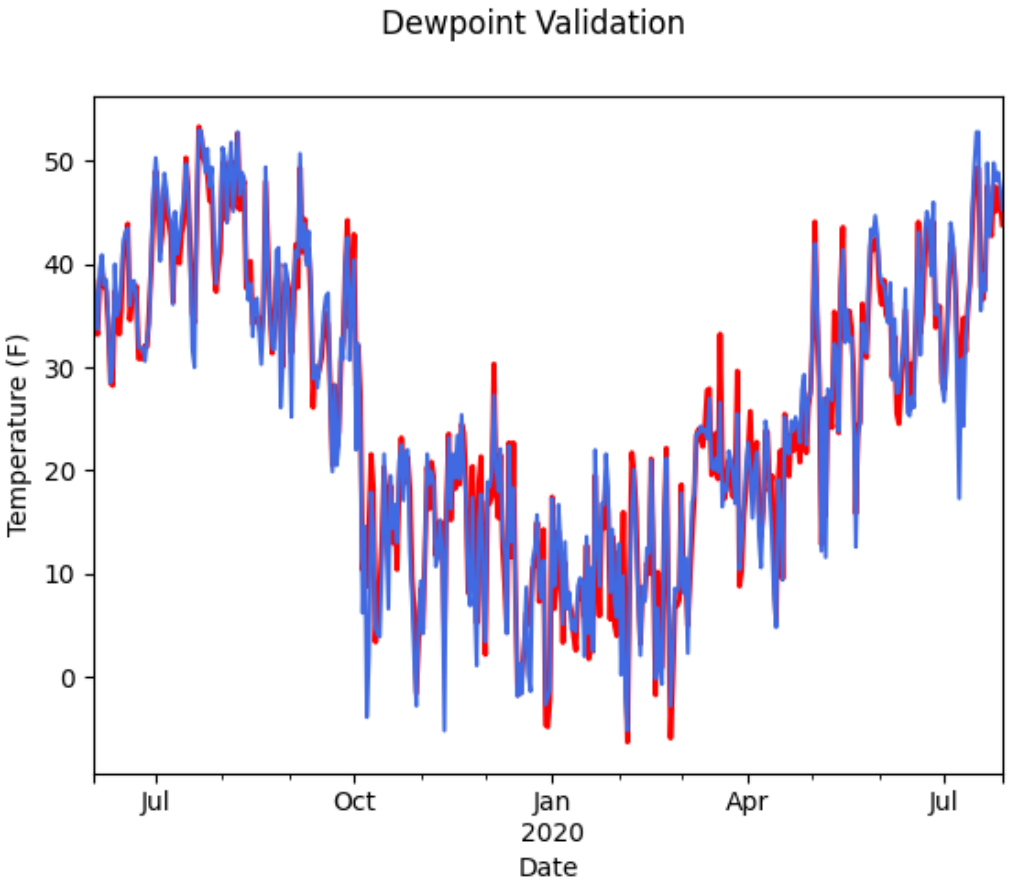

Figure 8: SARIMAX predictions (red) vs. true dew point on the test period (blue).

The SARIMAX predictions follow the general shape of the time series with some moderate local errors, while the ensemble predictions can sometimes be closer numerically but slightly less smooth. For an operational tool used by winegrowers, stability and interpretability are as important as few-tenths of a degree of error.

7.3 From Frost Point to Frost Risk

Using the forecasts of frost point, it is straightforward to derive a binary "frost risk" indicator. For example, a day is flagged as "frost risk" if:

$$\hat{y}_t \leq 0\ ^\circ\mathrm{C},$$

or if $\hat{y}_t$ is below a slightly higher threshold to account for uncertainty. The predicted frost risk calendar can then be communicated to winegrowers, allowing them to focus their preparation on the most critical days.

```
Detected frost periods:
- From 2020-10-05 to 2020-10-09
- From 2020-10-13 to 2020-10-13
- From 2020-10-29 to 2020-10-29
- From 2020-11-10 to 2020-11-10
- From 2020-11-16 to 2020-11-16
- From 2020-12-26 to 2020-12-26
- From 2021-03-01 to 2021-03-01
- From 2021-03-12 to 2021-03-12
- From 2021-04-01 to 2021-04-01
- From 2021-04-09 to 2021-04-09
- From 2021-04-21 to 2021-04-21
- From 2021-04-28 to 2021-04-29
- From 2021-05-07 to 2021-05-08
- From 2021-05-19 to 2021-05-22
```

Figure 9: Prediction of frost over the next year

```
Frost periods longer than 3 days:
- From 2020-10-05 to 2020-10-09  (5 days)
- From 2021-05-19 to 2021-05-22  (4 days)
```

Figure 10: Periods of frost longer than 3 day

8 Conclusion

This project addressed the prediction of average temperature and dew point over one year using historical meteorological data to calculate frost point, with the practical objective of helping winegrowers anticipate frost events. From a Data and Artificial Intelligence perspective, it involved:

- Building a clean and informative feature set, including engineered variables such as barometer and humidity ranges and cyclical encodings of the calendar.
- Exploring multiple modelling families, from classical SARIMAX time series models to nonlinear machine learning models (SVR, LightGBM, XGBoost) and their ensembles.
- Analysing overfitting and understanding the limitations of models that do not explicitly represent temporal dependence.

Even though ensembles of SVR and gradient boosting achieved excellent numerical accuracy on the test set, they showed signs of overfitting and lacked explicit temporal structure. In contrast, SARIMAX provided a good trade-off between accuracy, robustness and interpretability, which is crucial when the model is meant to support real-world decisions in viticulture.

In future work, several extensions could be considered:

- Incorporating spatial information from multiple weather stations or vineyards.

- Using a dataset that contains surface and radiation data for a more accurate prediction.
- Testing deep learning models designed for sequences (e.g., LSTM or temporal convolutional networks) with careful regularisation.
- Integrating probabilistic forecasting (prediction intervals) to better quantify the uncertainty around frost risk.

Overall, the project demonstrates how data-driven methods can concretely support agriculture, while also illustrating the importance of choosing models that are aligned with the structure of the data.

References

[1] Rob J. Hyndman and George Athanasopoulos, *Forecasting: Principles and Practice*. OTexts, 2nd edition, 2018.

[2] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung, *Time Series Analysis: Forecasting and Control*. Wiley, 5th edition, 2015.

[3] Alex J. Smola and Bernhard Schölkopf, "A Tutorial on Support Vector Regression", *Statistics and Computing*, 14(3), 199–222, 2004.

[4] Jerome H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine", *Annals of Statistics*, 29(5), 1189–1232, 2001.

[5] Tianqi Chen and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System", In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[6] Fabian Pedregosa et al., "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, 12, 2825–2830, 2011.