
# Predicting Droughts in Uganda

from meteorological satalite data

*Author:*
Behrouz Eslami and
Misha Klein
behrooz.eslami@gmail.com

May 10, 2020

# 1 Summary

Here we train and optimize a logistic regression model to predict the occurrence of droughts in Uganda based of meteorological data. The precipitation and the **E**nhanced **V**egitation **I**ndex (EVI) turn out to be the most important drought predictors among all the meteorological quantities that we studied, implying that the drought reports are triggered by both the hydrological and agricultural factors. The model is able to capture 70 percent of the reported droughts in our historical data. Based on our model, we derive a "drought score" which would serve as a potentially useful metric to monitor and predict droughts in Uganda.

## 2 Data collection

Our aim is to construct a model to predict the occurrence of droughts in Uganda, using meteorological satellite data. The meteorological parameters that serve as drought indicators, and to which we refer as **features**, are collected using the Google Earth engine by this python script, and are listed bellow

- **NDVI:** '**N***ormalized* **D***ifference* **V***egitation* **I***ndex*' (scale by 0.0001).

- **EVI:** '**E***nhanced* **V***egitation* **I***ndex*' (scale by 0.0001).
- **precipitation:**: Rainfall in mm/hrs. Name in our dataset: '`precipitation per hour v1`'.
- **hourlyPrecipRate:** Rainfall measured in mm/hrs. Name in our dataset: '`precipitation per hour v2`'.
- **LST_Day_1km:** Land surface temperature during daytime in 50 Kelvins. Name in our dataset: '`surface temperature daytime`'.
- **LST_Night_1km:** Land surface temperature during daytime in 50 Kelvins. Name in our dataset: '`surface temperature nighttime`'.
- **Evap_tavg:** Evapotranspiration measured in kg m^{-2} s^{-1}. Name in our dataset: '`evapotranspiration`'.
- **Rainf_f_tavg:** Precipitation rate measured in kg m^{-2} s^{-1} Name in our dataset: '`rainfall`'.
- **SoilMoi00_10cm_tavg:** Soil moisture (0 - 10 cm underground) in m^3 m-3. Name in our dataset: '`SoilMoisture00 10cm`'.
- **SoilMoi10_40cm_tavg:** Soil moisture (10 - 40 cm underground) in m^3 m-3. Name in our dataset: '`SoilMoisture10 40cm`'.
- **SoilMoi40_100cm_tavg:** Soil moisture (40 - 100 cm underground) in m^3 m-3. Name in our dataset: '`SoilMoisture40 100cm`'.
- **SoilMoi100_200cm_tavg:** Soil moisture (100 - 200 cm underground) in m^3 m-3. Name in our dataset: '`SoilMoisture100 200cm`'.
- **SoilTemp00_10cm_tavg:** Soil temperature (0 - 10 cm underground) in K. Name in our dataset: '`SoilTemperature00 10cm`'.
- **SoilTemp10_40cm_tavg:** Soil temperature (10 - 40 cm underground) in K. Name in our dataset: '`SoilTemperature10 40cm`'.
- **SoilTemp40_100cm_tavg:** Soil temperature (40 - 100 cm underground) in K. Name in our dataset: '`SoilTemperature40 100cm`'.
- **SoilTemp100_200cm_tavg:** Soil temperature (100 - 200 cm underground) in K. Name in our dataset: '`SoilTemperature100 200cm`'.
- **Tair_f_tavg:** Near surface air temperature in K. Name in our dataset: '`air temperature`'.
- **Wind_f_tavg:** Near surface wind speed in m/s. Name in our dataset: '`wind speed`'.

- **SPEI** columns: We calculated the '**S***tandardized* **P***recipitation*-**E***vapotranspiration* **I***ndex*' from the columns '`rainfall`' and '`evapotranspiration`' in our dataset, using the published R package. We have used different *scale* parameters from 1 to 12 months, generating the columns '`SPEI 1month`' to '`SPEI 12month`'. The R script to perform this calculation can be find here.

All features were averaged over each district (i.e. admin. level 1), and either averaged or summed in successive one-month intervals (see the python script for details).

The target variable that we are trying to predict, to which we refer to as the **label**, is the occurrence of droughts at a certain time point/period in a particular district. This would be a binary (i.e. Boolean) variable, taking the value 'True' when a drought has occurred, and 'False' when no drought is reported. This information comes from two different sources, the Red Cross desinventar and news articles, both can be found here. We chose to build the label variable by combining the two sources together. This is represented by the column '`drought reported`' in the dataset.

We merged all the features and labels into a single dataset using this Jupyter notebook. The resultant dataset can be found here. Each row of the dataset corresponds to a unique pair of date and district.

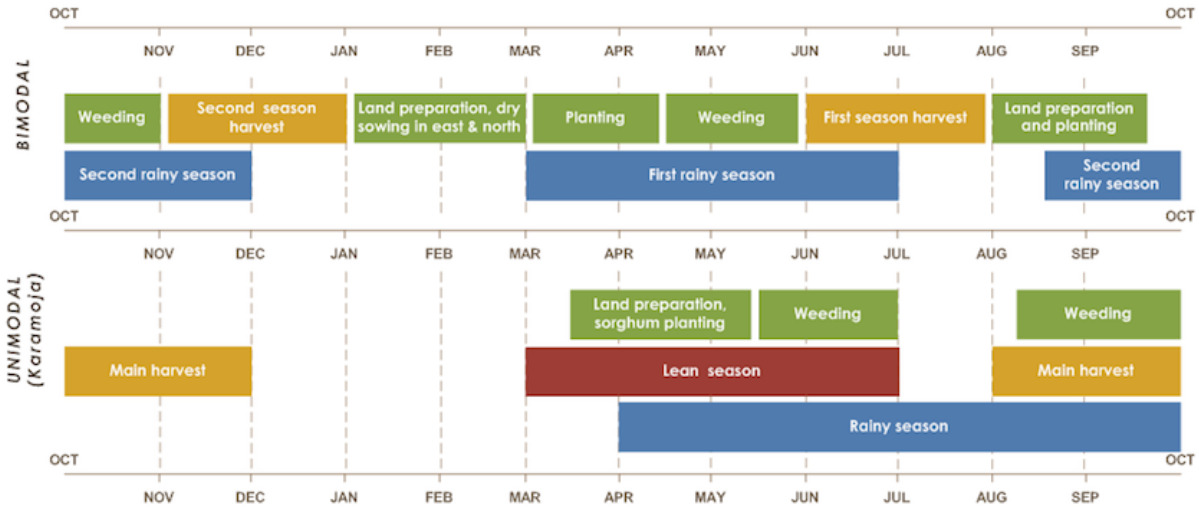


***Figure 3.1:*** agricultural calendar for Uganda

# 3 Data preprocessing

We consider the bimodal agricultural calendar for Uganda (Figure 3.1, top panel) which has two harvest seasons per year: the first one during June and July, and the second one in November and December.

We set as our goal to predict whether or not a drought occurs within each harvest season, based on the meteorological data in the three-month period preceding the season. To this end we redefine the label variable to have the value `True` if a drought is reported in any of the two months of the season, and `False` otherwise. We also average all the meteorological data, except SPEI, over the preceding three-month period and use them as features for our model. For SPEI we use the column '`SPEI 3month`' and take the data points that correspond to the three-month period before each harvest season. We then normalize all features except `SPEI` per district and season. This means that for any given district and season, we calculate the mean and the standard deviation of each feature during all years, and then standardize the feature by subtracting the mean and dividing by the standard deviation. The `SPEI` feature is already normalized by construction.
The transformed dataset can be downloaded from here. Each row is indicated by a unique triple of district, year and season. The first harvest season is indicated by `6_7` and the second one by `11_12`.

As a final step of preparing the training data, we decided to only keep those rows in the dataset that correspond to either a year with a reported drought or a year preceding/following a drought. This will potentially reduce the noise in the data due to the droughts that have actually happened but remained unreported.

# 4 Data exploration

## 4.1 Correlations among the features

Many of the features in our dataset are strongly correlated. To investigate this, we used a biclustering algorithm to divide the features into three separate groups. Features within a group correlate strongly with each other, while the correlation between features from different groups is weaker. Figure 4.1 Shows the resultant correlation matrix, where the colors represent the absolute value of the correlation coefficient. We can see that the soil moisture and soil temperature features all correlate strongly with each other, as expected. Vegetation indices EVI and NDVI correlate well with each other, with air and surface temperatures, and with the soil temperature (but not with the soil moisture). Surprisingly the features related to precipitation, namely '`rainfall`', '`precipitation_per_hour_v1`' and '`precipitation_per_hour_v2`', do not correlate well with each other. The reason for this is not clear for us.

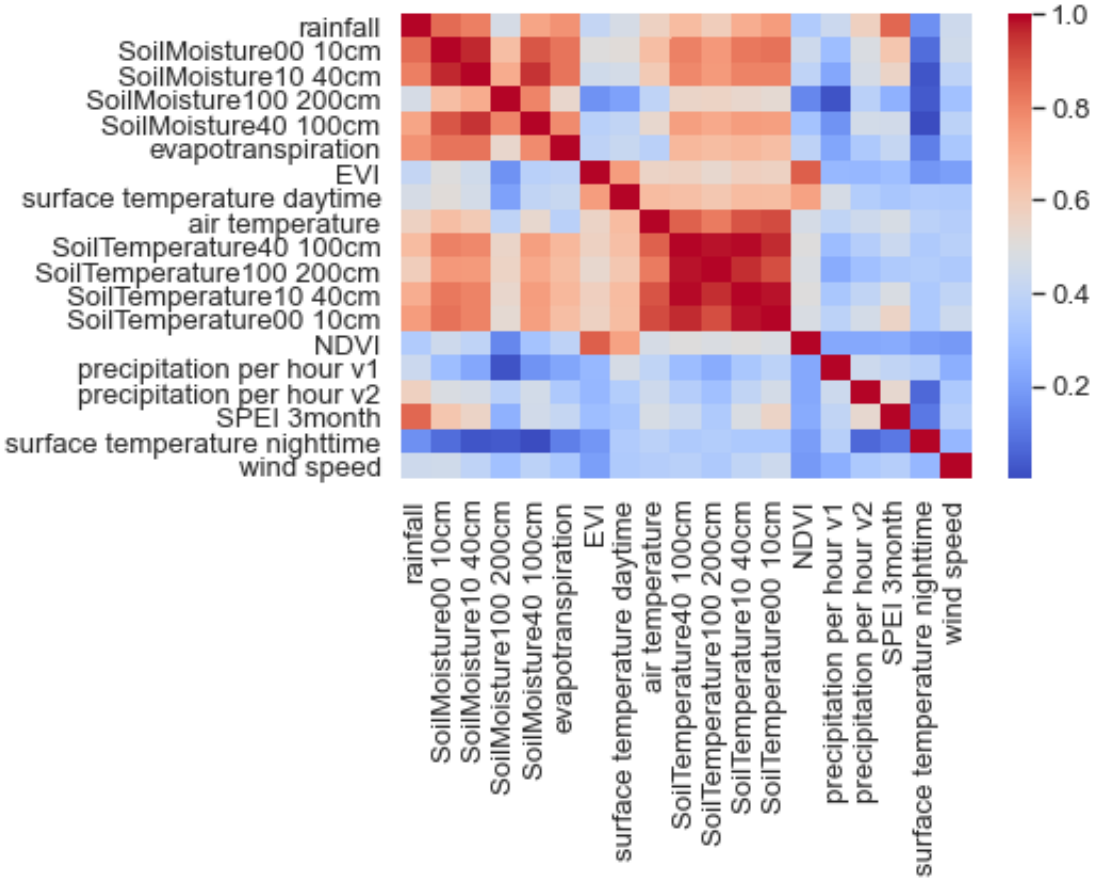


***Figure 4.1:*** Correlations matrix for all pairs of features. The colors represent the absolute value of the correlation coefficient.

The correlation plots among the features within each of the three groups are shown in Figures 4.2 to 4.4.

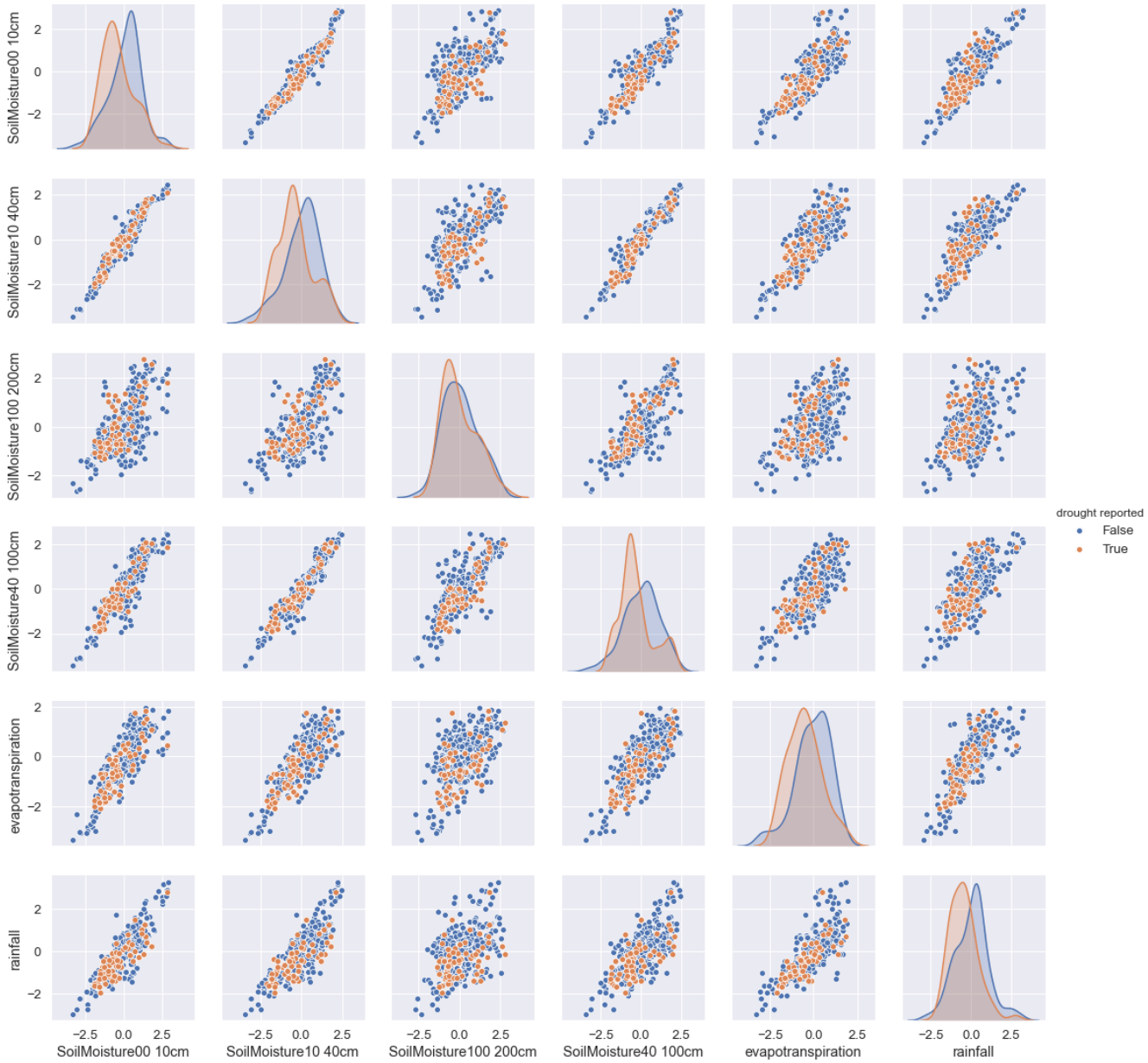


***Figure 4.2:*** Correlation plots for the features in group 1. The points are colored based on their label values.


***Figure 4.3:*** Correlation plots for the features in group 2. The points are colored based on their label values.

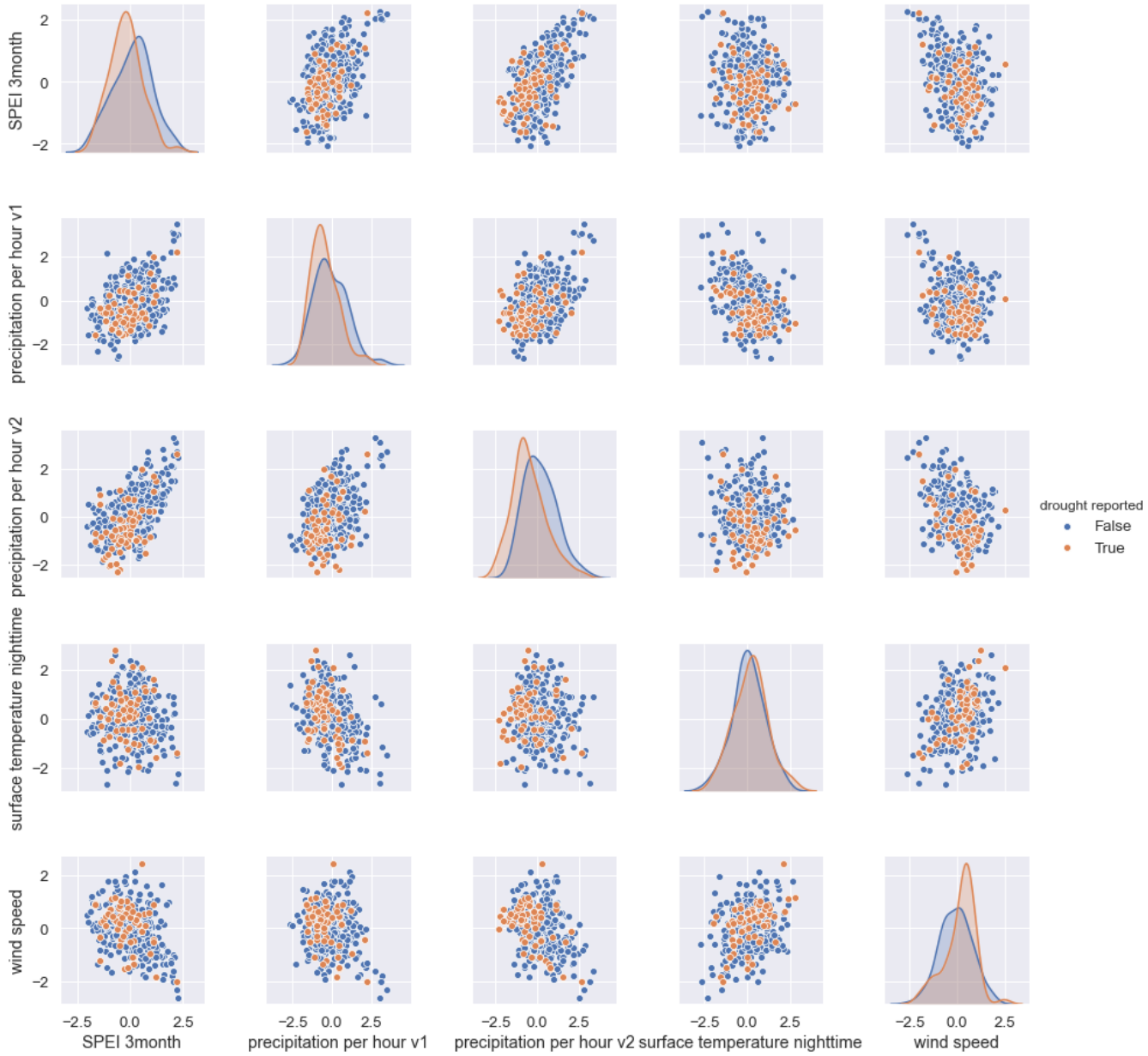


*Figure 4.4:* Correlation plots for the features in group 3. The points are colored based on their label values.

## 4.2 Correlation with the target

Figures 4.5 to 4.23 show the distribution of each feature separately for '`drought reported =False`' (blue) and '`drought reported = True`' (orange). The medians of the distributions, as well as the absolute values of the difference between them, are reported in Table 4.1. The features are ranked in descending order according to the difference in medians. From Table 4.1 it is clear that the features that are related to precipitation, either directly or indirectly, are important drought predictors.

***Table 4.1:*** The median values of the features grouped by the label. The column 'difference' shows the absolute values of the difference between the two groups.

| | feature | no drought | yes drought | difference |
|---|---|---|---|---|
| 0 | SoilMoisture00 10cm | 0.280299 | -0.570226 | 0.850525 |
| 1 | precipitation per hour v2 | 0.063213 | -0.698218 | 0.761430 |
| 2 | SoilMoisture10 40cm | 0.216020 | -0.530722 | 0.746742 |
| 3 | SoilMoisture40 100cm | 0.203412 | -0.531513 | 0.734925 |
| 4 | evapotranspiration | 0.044880 | -0.602795 | 0.647675 |
| 5 | rainfall | 0.158049 | -0.486975 | 0.645024 |
| 6 | SoilTemperature100 200cm | -0.268359 | 0.320006 | 0.588365 |
| 7 | SoilTemperature40 100cm | -0.224721 | 0.361192 | 0.585913 |
| 8 | SoilTemperature10 40cm | -0.211171 | 0.302455 | 0.513625 |
| 9 | SoilTemperature00 10cm | -0.144578 | 0.340174 | 0.484752 |
| 10 | EVI | 0.214773 | -0.196214 | 0.410986 |
| 11 | SPEI 3month | 0.211545 | -0.192696 | 0.404242 |
| 12 | wind speed | -0.024963 | 0.329311 | 0.354275 |
| 13 | precipitation per hour v1 | -0.240641 | -0.549282 | 0.308641 |
| 14 | NDVI | 0.141345 | -0.127157 | 0.268502 |
| 15 | SoilMoisture100 200cm | -0.036950 | -0.292951 | 0.256001 |
| 16 | surface temperature daytime | -0.092329 | 0.155518 | 0.247847 |
| 17 | air temperature | -0.153588 | 0.028358 | 0.181946 |
| 18 | surface temperature nighttime | 0.090008 | 0.235946 | 0.145938 |

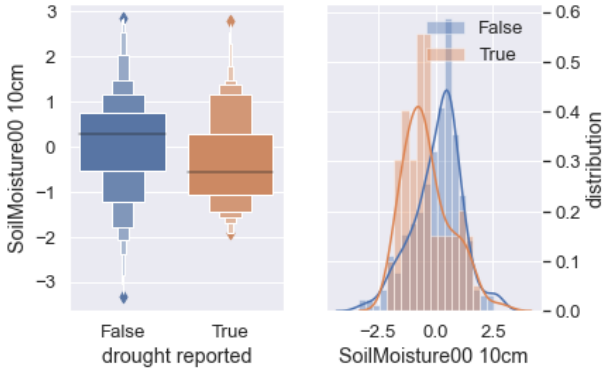


***Figure 4.5:*** Distribution of '`SoilMoisture00 10cm`' separated by the label values.

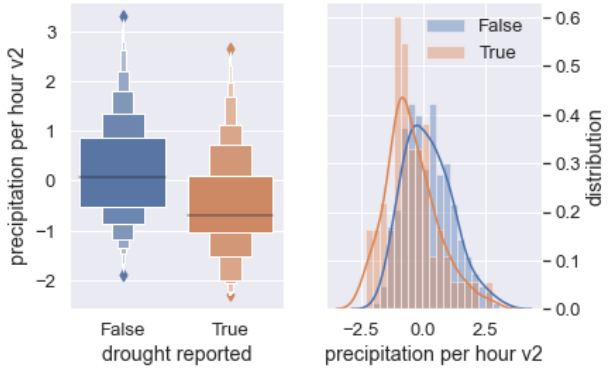


***Figure 4.6:*** Distribution of '`precipitation per hour v2`' separated by the label values.

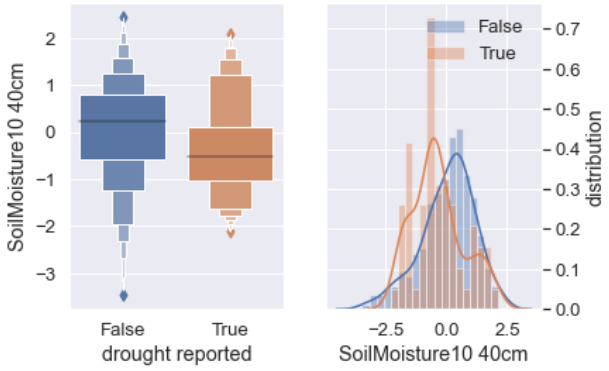


*Figure 4.7:* Distribution of '`SoilMoisture10 40cm`' separated by the label values.

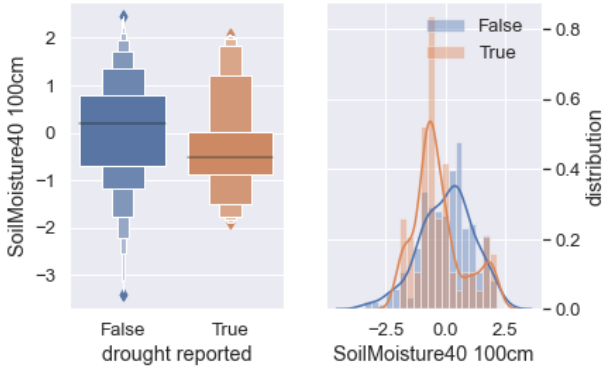


*Figure 4.8:* Distribution of '`SoilMoisture40 100cm`' separated by the label values.

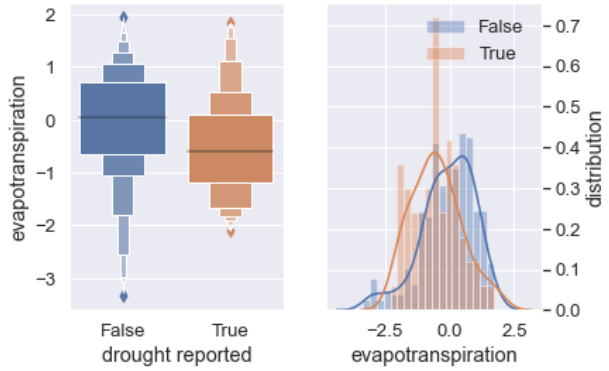


*Figure 4.9:* Distribution of '`evapotranspiration`' separated by the label values.

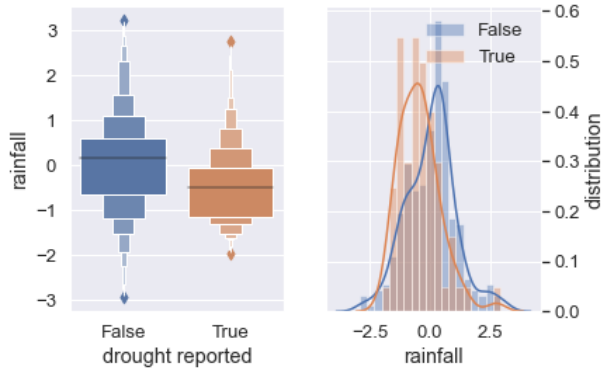


***Figure 4.10:*** Distribution of '`rainfall`' separated by the label values.

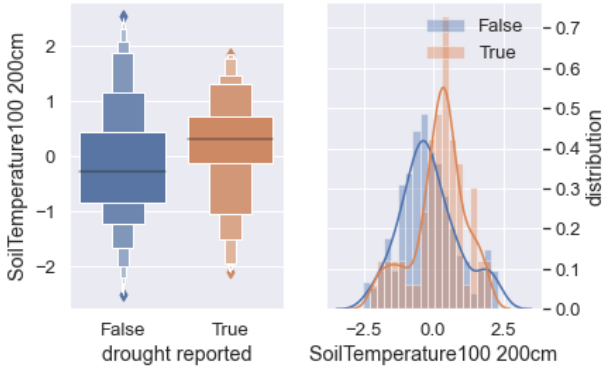


***Figure 4.11:*** Distribution of '`SoilTemperature100 200cm`' separated by the label values.

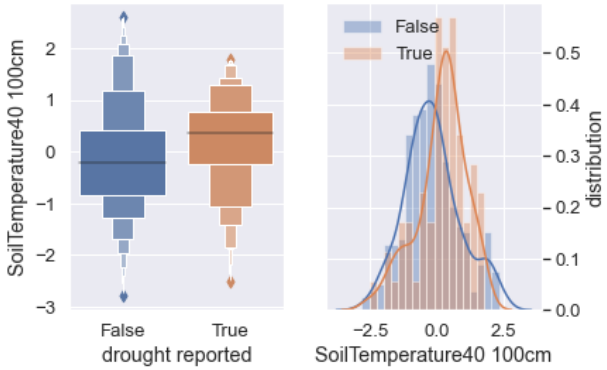


***Figure 4.12:*** Distribution of '`SoilTemperature40 100cm`' separated by the label values.

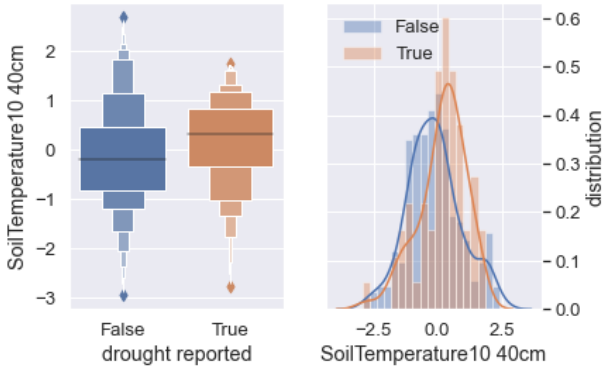


*Figure 4.13:* Distribution of '`SoilTemperature10 40cm`' separated by the label values.

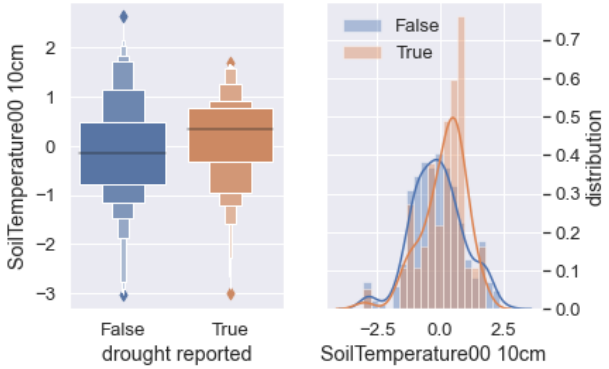


*Figure 4.14:* Distribution of '`SoilTemperature00 10cm`' separated by the label values.

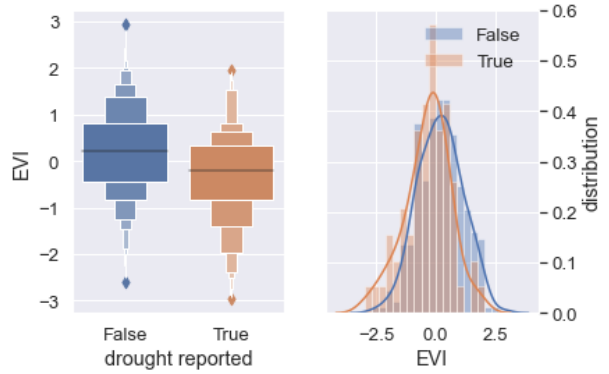


*Figure 4.15:* Distribution of '`EVI`' separated by the label values.

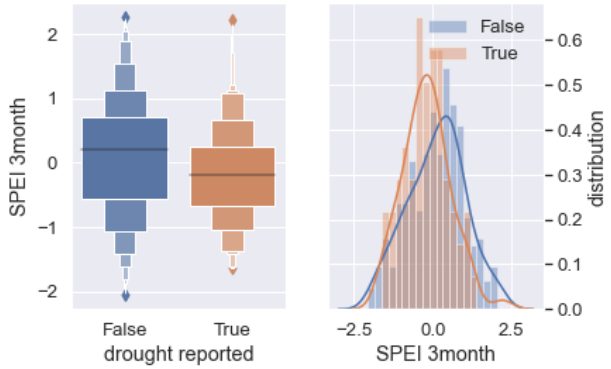


*Figure 4.16:* Distribution of ‘`SPEI 3month`’ separated by the label values.

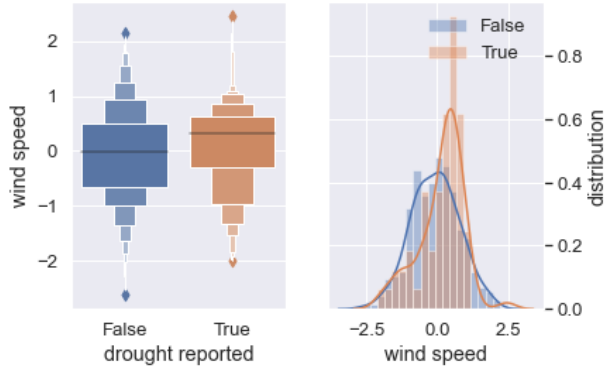


*Figure 4.17:* Distribution of ‘`wind speed`’ separated by the label values.

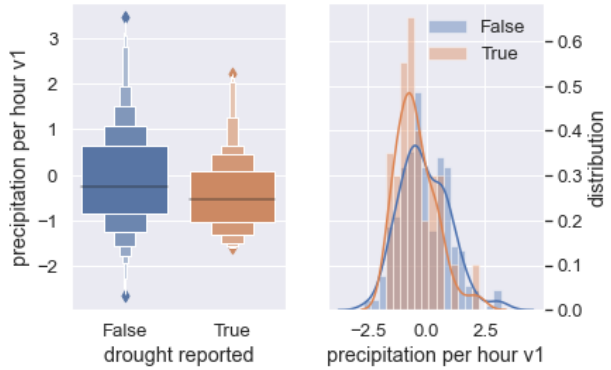


*Figure 4.18:* Distribution of ‘`precipitation per hour v1`’ separated by the label values.

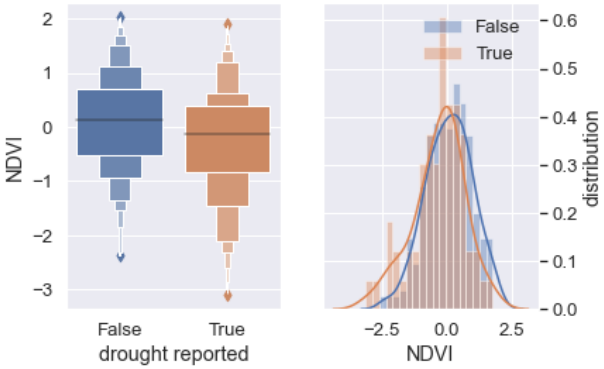


*Figure 4.19:* Distribution of '`NDVI`' separated by the label values.

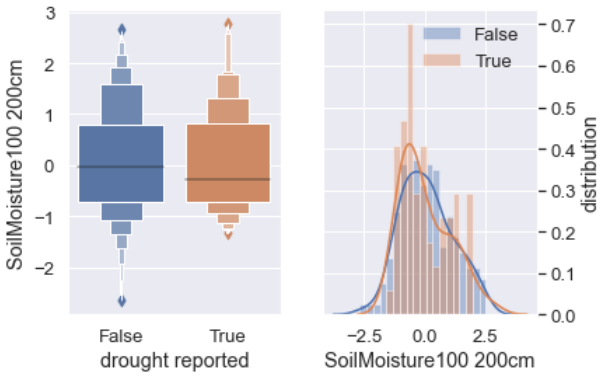


*Figure 4.20:* Distribution of '`SoilMoisture100 200cm`' separated by the label values.

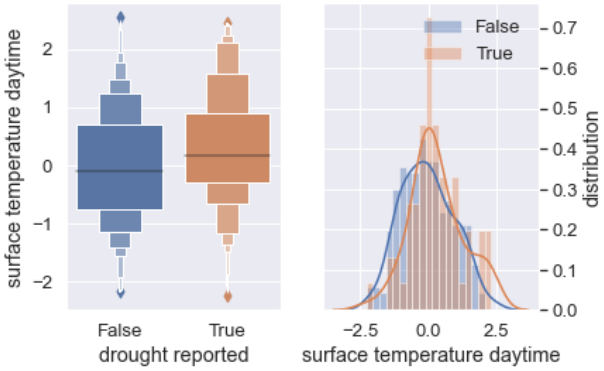


*Figure 4.21:* Distribution of '`surface temperature daytime`' separated by the label values.

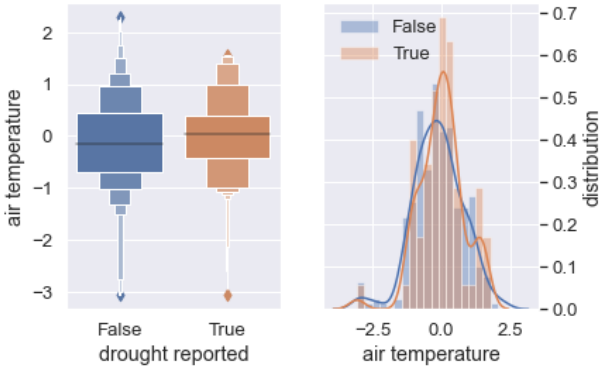


*Figure 4.22:* Distribution of ‘`air temperature`’ separated by the label values.

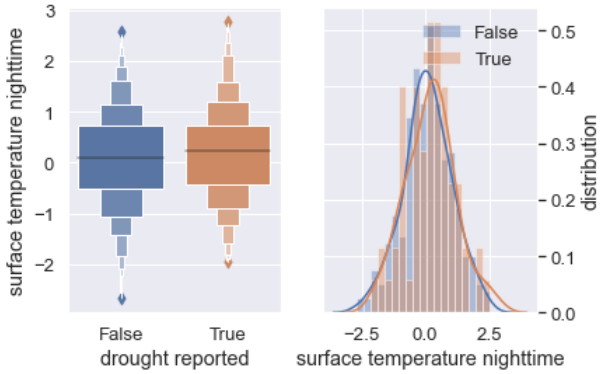


*Figure 4.23:* Distribution of ‘`surface temperature nighttime`’ separated by the label values.

# 5 Fitting and optimizing a Logistic Regression model

We train a logistic regression classifier with $l_1$ penalty to predict the occurrence of droughts in Uganda. Our goal is to separate the rows of the dataset into two classes: a positive class which represents the occurrence of a drought, (`drought = True`), and a negative class where the model predicts that no drought will occur (`drought = False`). It is important to note that the population of the positive class $n^{pos} = 65$ in our training data is about 5 times smaller the population of the negative class $n^{neg} = 297$. The negative class is therefore largely overrepresented, and a model trained on such data will be biased towards predicting the negative class correctly, without caring so much about capturing the positive class. Since having a good predicting power for the positive class is equally important for us, we need to correct for this imbalance in the class populations. We do this by assigning unequal weights $W^{pos}$ and $W^{neg}$ to the positive and the negative classes, which are inversely proportional to the class populations:

$$W^{pos} = \frac{\frac{1}{n^{pos}}}{\frac{1}{n^{pos}} + \frac{1}{n^{neg}}}, \tag{5.1}$$

and

$$W^{neg} = \frac{\frac{1}{n^{neg}}}{\frac{1}{n^{pos}} + \frac{1}{n^{neg}}}. \tag{5.2}$$

Feeding these weights into the logistic regression model guarantees that both classes are represented equally when the model is trained on the data.

The $l_1$ penalty in the logistic regression model is associated with a regularization parameter $C$, which tunes the model complexity by controlling the number of features that are included in the model. In the framework of logistic regression, the contribution of each feature is determined by a coefficient, and a zero coefficient means that the corresponding feature does not enter the model. The regularization parameter determines how many of these coefficients are different from zero. Figure 5.1 shows the model coefficients for all features as a function of the regularization parameter $C$. We can see that for large values of $C$ all coefficients are non-zero. As $C$ decreases, the coefficients gradually shrink to zero.

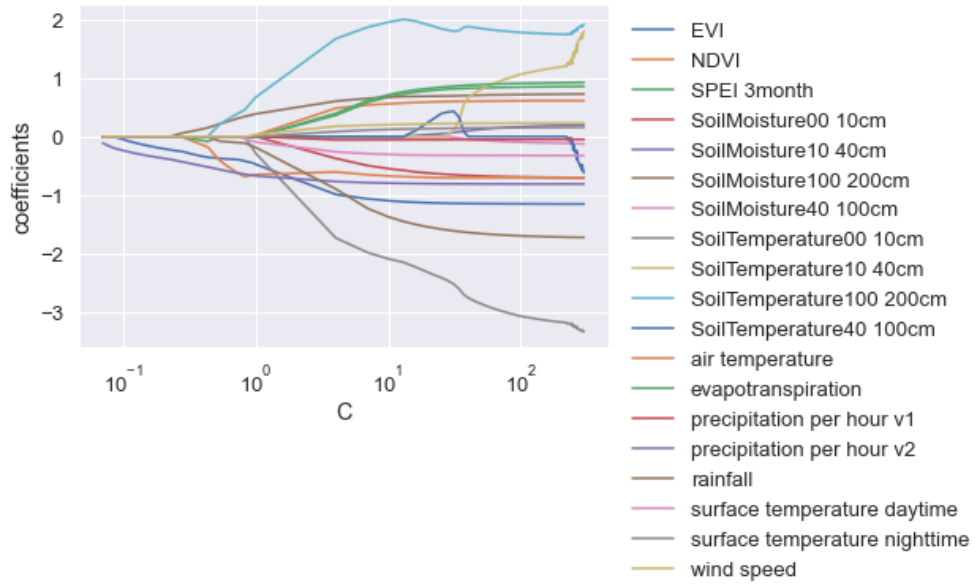


***Figure 5.1:*** Model coefficients for all features as a function of the regularization parameter $C$.

Choosing an appropriate value for the regularization parameter is important, as this systematically eliminates unimportant and redundant features, so that only the most important features remain in the model. We performed 3-fold cross-validation to find the best value for $C$, where we used the average F1 score as the goodness of fit measure. The average F1 score is given by the weighted sum of the F1 scores for the positive and negative classes:

$$F1 = W^{pos} F1^{pos} + W^{neg} F1^{neg}, \tag{5.3}$$

Where $F1^{pos}$ and $F1^{neg}$ are the F1 scores for the positive and negative classes, and $W^{pos}$ and $W^{neg}$ are given by equations 5.1 and 5.2 respectively.

To be sure that the outcome of the cross-validation is robust and generalizable, we repeated the procedure 10 times where we randomly shuffled the data before each round of cross-validation. The outcome is demonstrated in Figure 5.2, where the curves in pail colors show the cross-validation average F1 scores as a function of $C$ for each round of shuffling, and the mean of all these curves is shown in dark black. The training average F1 score is also shown in the Figure (dashed-dot curve, green).

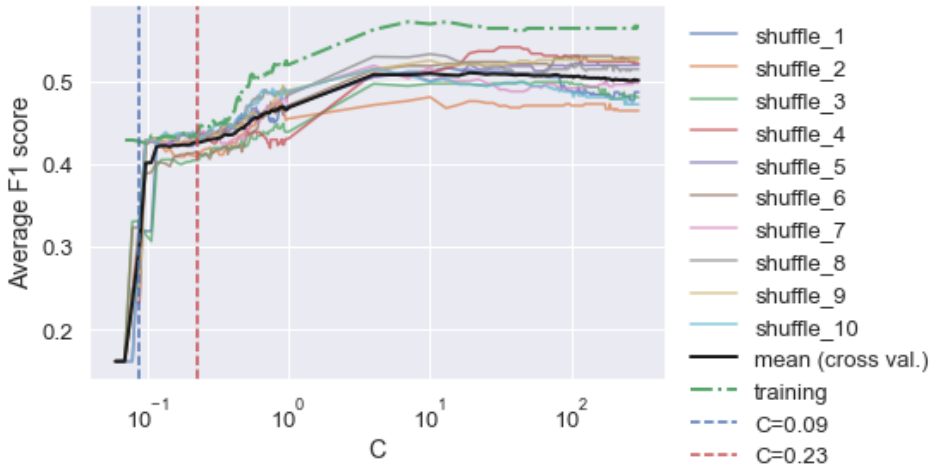


***Figure 5.2:*** Average F1 score as a function of the regularization parameter $C$. The curves in pail colors show the cross-validation average F1 scores after randomly shuffling the data, and their mean is shown in dark black. The dashed-dot curve (green) corresponds to the average F1 score for the training data. The blue and red vertical dashed lines indicate $C = 0.09$ and $C = 0.23$ respectively.

Looking at the dark black curve in Figure 5.2, it is clear that for $C < 0.09$ (blue vertical dashed line) the cross-validation F1 score rapidly grows by increasing $C$. For values between 0.09 and 0.23 (red vertical dashed line), the cross-validation average F1 score remains relatively constant around 0.42, and the same happens for the training average F1 score (see dashed-dot curve, green). For $C > 0.23$, although the training score continues to grow and saturates at 0.57, the cross-validation score always remain below the training scores and quickly saturates at 0.48 after a mild increase. This implies that for values of $C > 0.23$ the model stops to nicely generalize to new data. In other words, the model would have the tendency to overfit to the training data.

***Table 5.1:*** Nonzero model coefficients for $C = 0.3$. The model overfits to the training data.

| feature | coefficients |
|---|---|
| precipitation per hour v2 | -0.45 |
| EVI | -0.27 |
| SoilMoisture100 200cm | 0.07 |
| air temperature | -0.04 |
| evapotranspiration | -0.02 |

In the case of our drought model, detecting the overfitting is rather easy, as it leads to intuitively nonsensical models. See, for example, the model coefficients for $C = 0.3$ which are listed in Table 5.1. The coefficient for precipitation is negative, which makes sense, as the precipitation should anticorrelate with droughts. However, the positive coefficient for soil moisture and the negative coefficient for air temperature do not make sense intuitively, indicating that the model is fine-tuning itself to the training data. We find that we get more and more of such nonsensical coefficients as we increase $C$ further. Since we inferred from Figure 5.2 that the model tends to overfit for $C > 0.23$, and the data in Table 5.1 confirms that the overfitting already happens for values as small as 0.3, In the rest of this analysis we use the optimal value of $C = 0.23$ for the regularization parameter.

# 6 Model evaluation

A logistic regression model produces a score as a linear combination of the features, where the contribution of each feature is determined by its associated coefficient. Table 6.1 lists the nonzero coefficients for our optimal drought model with $C = 0.23$. Interestingly, only two features, '`precipitation per hour v2`' and '`EVI`', enter the optimal model, implying that the drought reports are triggered by both the hydrological and agricultural factors. According to Table 6.1, the score produced by our optimal drought model can therefore be written as

$$\text{score} = -0.38 \times \text{'precipitation per hour v2'} - 0.23 \times \text{'EVI'}. \tag{6.1}$$

***Table 6.1:*** Nonzero model coefficients for the optimal model with $C = 0.23$.

| feature | coefficients |
|---|---|
| precipitation per hour v2 | -0.38 |
| EVI | -0.23 |

Note that, as mentioned earlier, in equation 6.1 both features are normalized per district and per season. The score can be positive or negative, and is a measure for the likelihood of droughts. The more positive the score is, the higher is the chance that a drought occurs. Figure 6.1 shows the distribution of this "drought score" separately for '`drought reported =False`' (blue) and '`drought reported = True`' (orange), confirming that the score correlates with the target.

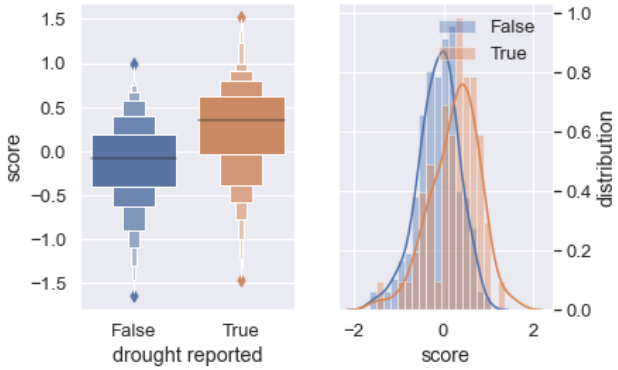


***Figure 6.1:*** Distribution of the drought score separated by the label values.

The confusion matrix and the accuracy associated with the optimal model, as well as precision, recall and the F1 scores for positive and negative classes, are given in Text 6.1. The model reaches the overall accuracy of 60%, and has an impressive recall of ~ 70% for the positive class (which means that the model captures 70% of the reported droughts). On the other hand, the precision for the positive class is rather low: about %25. This is partially due to the overrepresentation of the negative class, which leads to many false positives, and lowers the precision for the positive class as well as the recall for the negative class. We note however that there is a rather high chance that an actual drought in Uganda remains unreported. We therefore expect that the number of false positives presented in Text 6.1 is an overestimation. The actual predictive power of the model can only be assessed after testing it in the field.

***Text 6.1:*** Confusion matrix, accuracy, and the goodness of fit metrics for the optimal model.

```
                Confusion matrix
                Score positive    Score negative
Actual positive            45                20
Actual negative           127               170

Accuracy  0.59

           Positive       Negative
Num case         65            297
Precision      0.26           0.89
Recall         0.69           0.57
F-score        0.38           0.70

Weighted Average F-score  0.44
```

To have a baseline with which we can compare the performance of our model, we consider a "random model" which classifies the data by "coin flipping": irrespective to the features values, the model assigns `True` and `False` labels to the data points with equal probability. Text 6.2 lists the fit quality metrics for such a random model (averaged over 100 runs). We see that our model clearly outperforms the random model in every aspect, especially in the case of the positive class.

***Text 6.2:*** Goodness of fit metrics for a random model.

```
           Positive       Negative
Precision      0.18           0.82
Recall         0.50           0.50
F_score        0.26           0.62
```

Figures 6.2 and 6.3 show the precision-recall curve and the receiver operating characteristic (ROC) for the optimal model. The area under the ROC curve (auc) is a metric commonly used to assess the predictive power of binary classifiers. A perfect classifier has an auc of one, while for a random classifier auc equals 0.5. Our model reaches auc of 0.7.

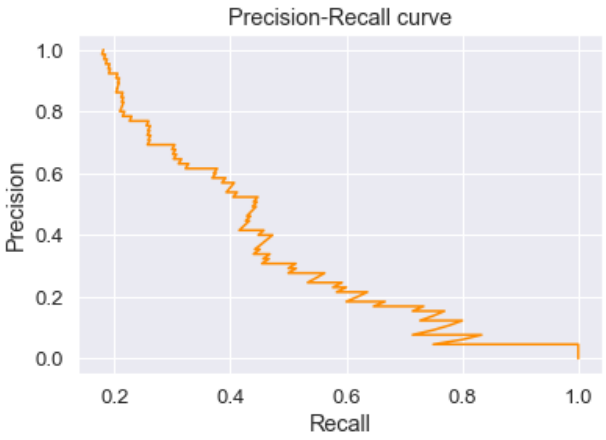


***Figure 6.2:*** Precision-recall curve for the optimal model.

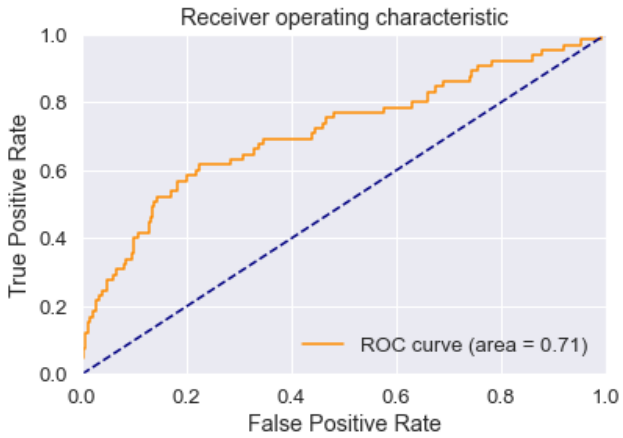


***Figure 6.3:*** receiver operating characteristic for the optimal model.

# 7 Drought prediction: successes and failures

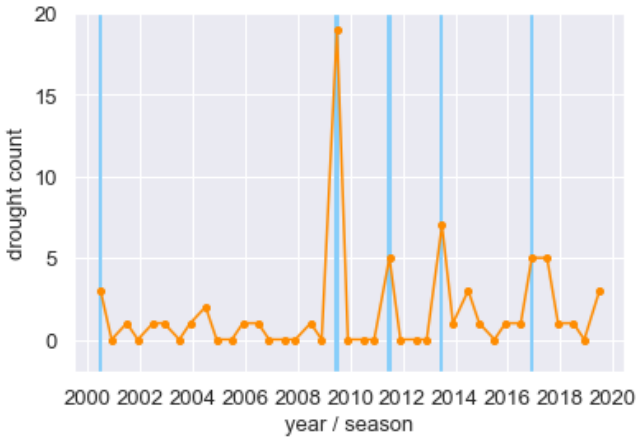


***Figure 7.1:*** Number of reported droughts in Uganda during each harvest season between 2000 and 2019. Major droughts are highlighted in blue (Related to Figures 7.2 to 7.8).

Figure 7.1 shows the number of reported droughts in Uganda during each harvest season between 2000 and 2019. Specifically, a large number of reports have been appeared in June-July 2009. Figure 7.2 shows the distribution of our drought score (equation 6.1) over the whole contry during this period: the score is generally positive and large. Conversely, in the same harvest season at 2010 no drought has been reported, and Figure 7.3 shows that the score is clearly lower. The model similarly captures the reported droughts (highlighted in Figure 7.1) in 2000 (Figure 7.4), 2011 (Figure 7.5) and 2016 (Figure 7.6).

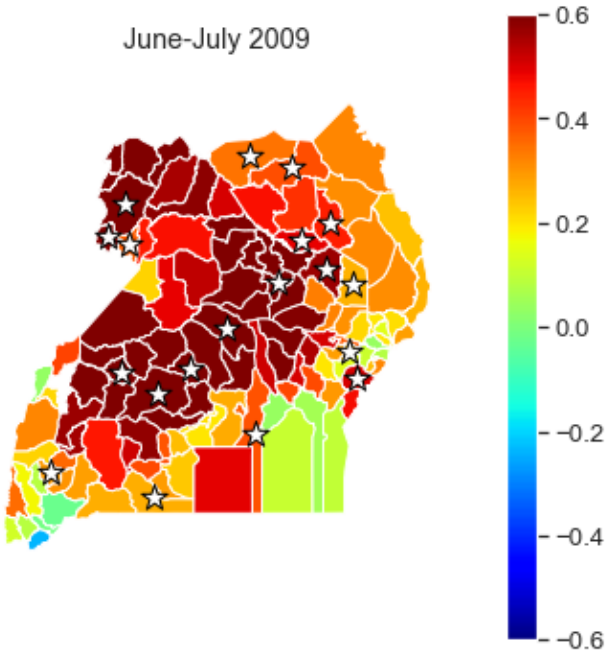


***Figure 7.2:*** Model scores for Uganda districts in the first harvest season in 2009. The colorbar is capped at +/- 0.6. Stars mark districts in which a drought has been reported. The model successfully captures the droughts.

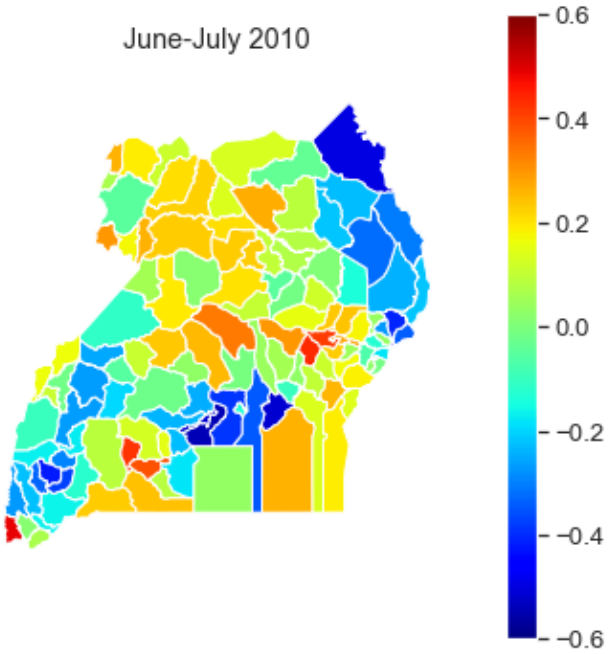


*Figure 7.3:* Model scores for Uganda districts in the first harvest season in 2010. The colorbar is capped at +/- 0.6. No drought has been reported, and the model scores are generally lower compared to the same period in 2009 (see Figure 7.2).

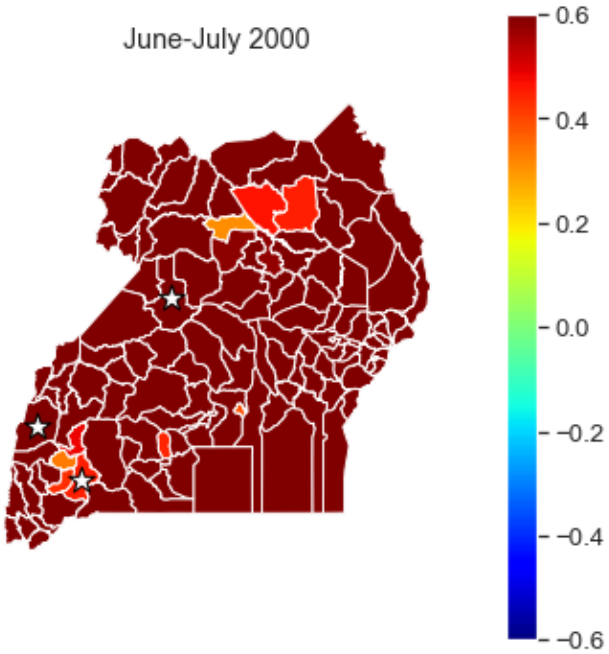


*Figure 7.4:* Model scores for Uganda districts in the first harvest season in 2000. The colorbar is capped at +/- 0.6. Stars mark districts in which a drought has been reported. The model successfully captures the droughts.

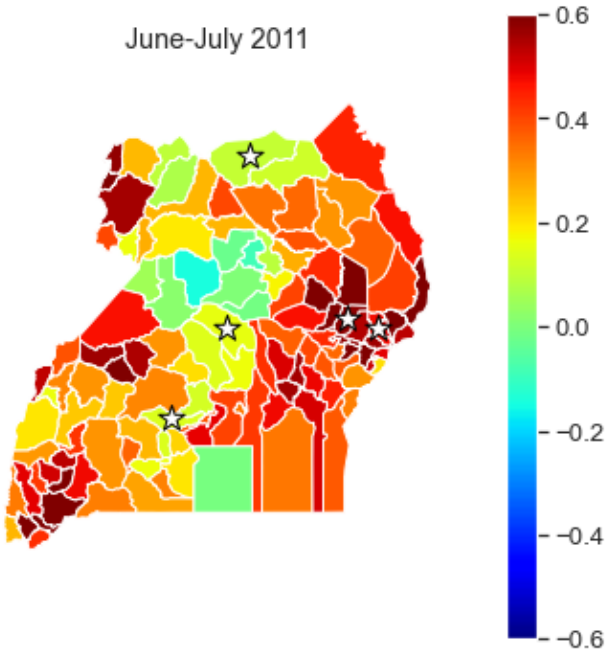


*Figure 7.5:* Model scores for Uganda districts in the first harvest season in 2011. The colorbar is capped at +/- 0.6. Stars mark districts in which a drought has been reported. The model successfully captures the droughts.

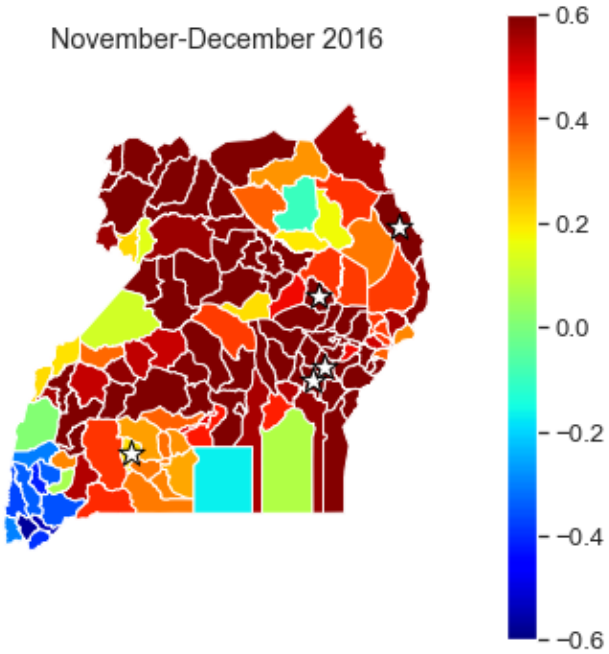


*Figure 7.6:* Model scores for Uganda districts in the second harvest season in 2016. The colorbar is capped at +/- 0.6. Stars mark districts in which a drought has been reported. The model successfully captures the droughts.

One notable exception is the period of June-July 2013 where the model fails drastically (see Figure 7.7): it produces a low score anywhere within the country while seven droughts has bin reported. In an attempt

to rationalize this, we slightly modified the drought score: instead of considering the three-month period preceding June 2013, we calculated the score over the period of May to June 2013. As Figure 7.8 shows, this modification improves the model performance in this particular instance. We therefore suggest that the failure of the model might be due to an unusually rapid change in the meteorological factors.

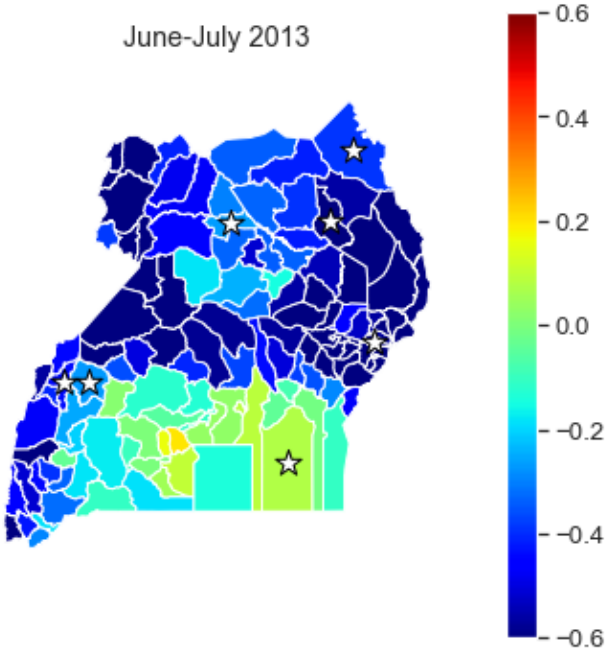


***Figure 7.7:*** Model scores for Uganda districts in the first harvest season in 2013. The colorbar is capped at +/- 0.6. Stars mark districts in which a drought has been reported. The model fails to capture the droughts.

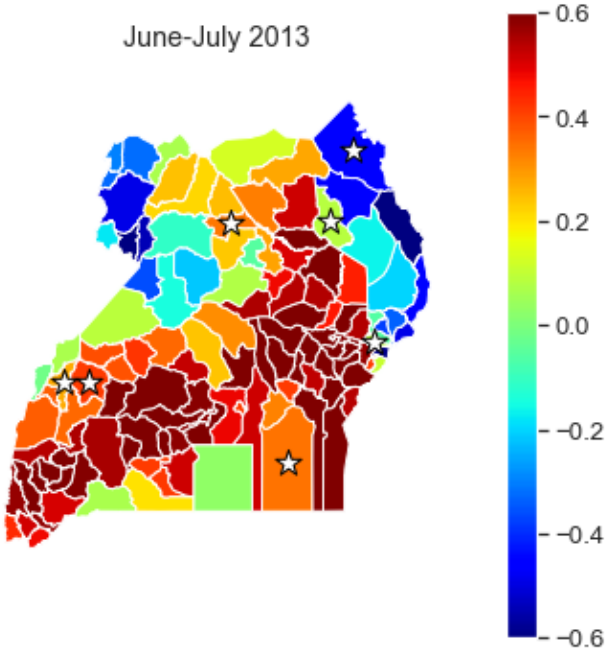


***Figure 7.8:*** The same as 7.7, but the score is calculated by averaging the features over the three-month period of May-July 20013. The model performance is improved.

# 8 Generalizing the model towards a drought-monitoring tool

Finally, we looked into the potential of our model as a drought-monitoring tool. To this end, we generalize our drought score in such a way that it can be calculated continuously over time, instead of being limited to just a three-month period prior to any harvest season. This is done by averaging the features using a sliding temporal window with a period of three months, normalizing them per month and district as described in section 2, and calculating the drought score using equation 6.1.

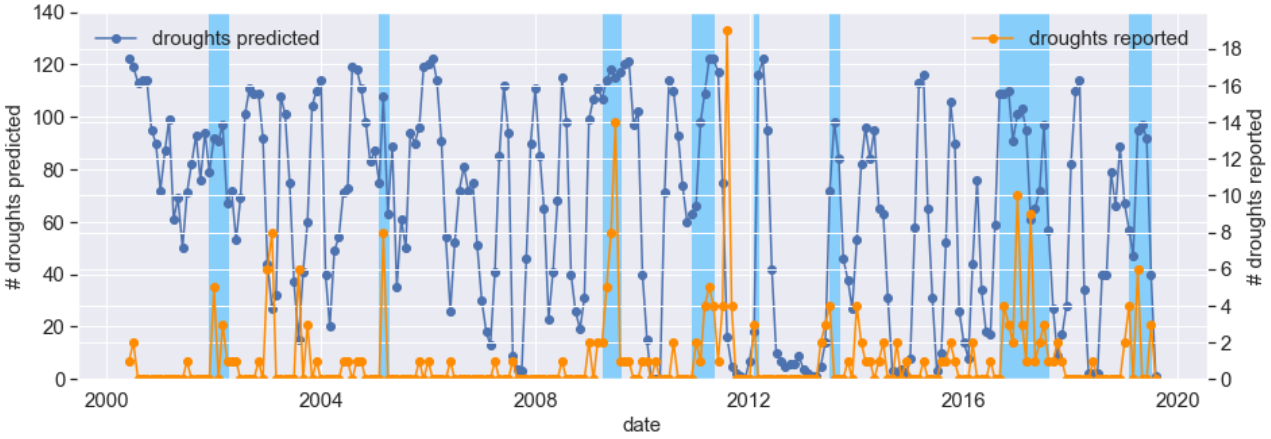


***Figure 8.1:*** Number of droughts in Uganda during each month between 2000 and 2019. Orange curve: number of reported droughts. Blue cure: predicted number of droughts by thresholding the draught score at zero. Some instances of successful performance of the model are highlighted in blue.

Figure 8.1 shows the total number of reported droughts in Uganda (orange curve), for every month between 2000 and 2019, compared with our model prediction (blue curve) obtained by thresholding the score at zero (i.e. positive score: drought occurs; negative score: drought does not occur). As expected from the large number of false positives in Text 6.1, our model generally overestimates the number of droughts. Notably, however, there are distinct time points where the model predicts a "global drought" in Uganda, where 100 or more districts out of the 120 districts of Uganda (i.e. more than 80%) simultaneously have positive drought scores. Interestingly, in many cases these predicted global droughts coincide with a large number of reported droughts (several examples are highlighted in blue in Figure 8.1). We therefore suggest that these timepoints are of particular importance where our drought score, if implemented in a drought monitoring tool, can serve as an warning for extensive droughts in Uganda. A crude example of such a monitoring system is implemented in the end of this Jupyter notebook (see Figures 8.2 and 8.3), where users can explore the distribution of the drought score at any date they chose, along with any reported drought if exists.

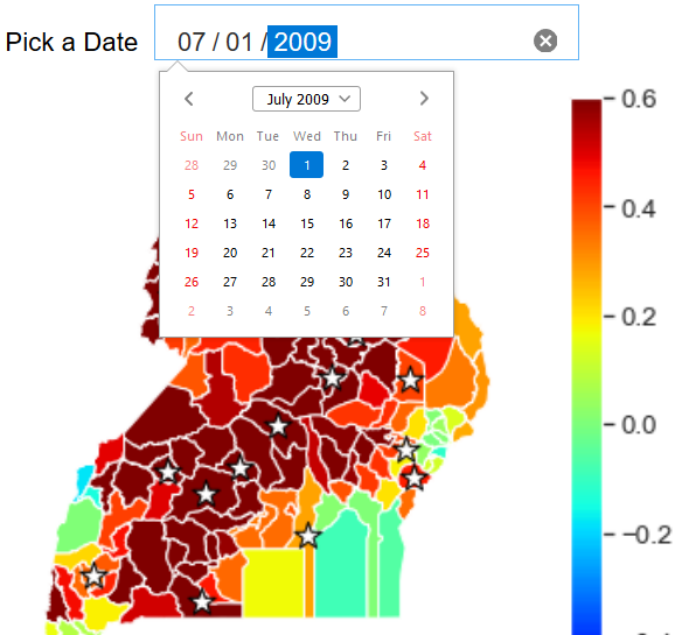


*Figure 8.2:* A crude example of a simple drought monitor with a date picker.

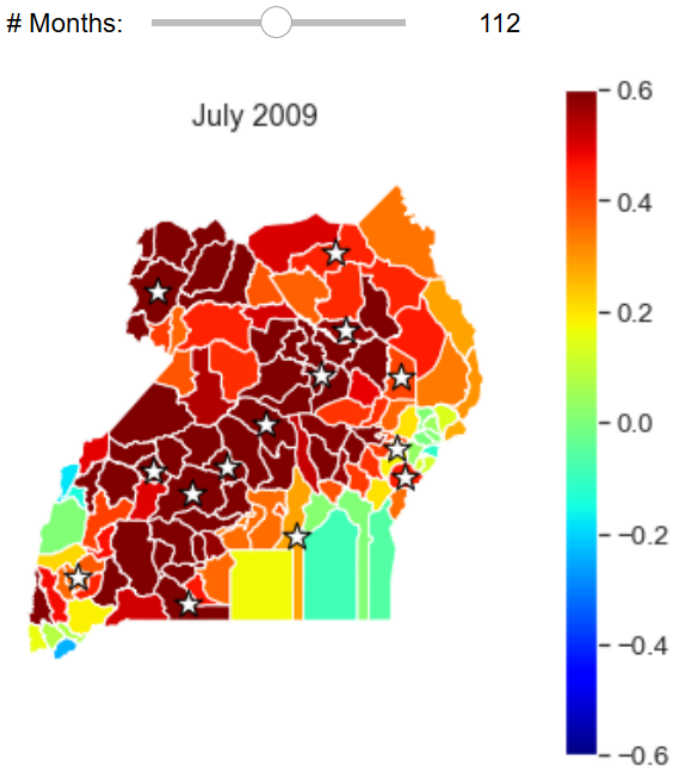


*Figure 8.3:* Another example of a drought monitor with a date slider.

# 9 Resources

This report has been generated from a Jupyter notebook. All the associated datasets, python scrips and the LaTeXsource file of this report can be accessed via the Rodekruis GitHub.