Predicting future droughts in the United States

Photo by Bob Nichols, U.S. Department of Agriculture, Austin, Texas (US)

EAEEE4000
Machine Learning for Environmental Engineering and Science
Ema FAGOUL

Table of Contents

GitHub : https://github.com/emafagoul/EAEEE4000

Introduction

Since the 1970s, the area affected by drought has doubled to the point where droughts claim more victims and displace more people than cyclones, floods and earthquakes combined. The media pays less attention to these natural disasters because they are less spectacular. Drought is generally defined as "a deficiency of precipitation over an extended period of time (usually a season or more), resulting in a water shortage." (National Drought Mitigation Center, 2022). Lack of water combined with extreme temperatures will exacerbate drought as more evaporation and transpiration of plants will occur (evapotranspiration), which dries the soil. Natural causes of drought are a lack of water and high temperatures, but human activities exacerbate the situation. Agricultural activities, factories, and homes require large amounts of water and place a strain on water reserves. The deficit in the reserves combined with a poor water resource management make drought even more severe.

The United States is one of the countries that is and will be most prone to drought periods, especially the western side in the states of California, Texas, Oregon, Nevada, Utah, and New Mexico. These states are regularly affected by drought due to their geographical position which gives them a hot and dry climate (Vatter, 2019). Many consequences can result from droughts, such as water scarcity, crop losses, local food shortages, and forest fires, as well as indirect consequences, such as migration, unemployment, and social unrest. Predicting droughts poses a high challenge in order to address these issues. Drought prediction can enable better anticipation and responsiveness.

Drought is typically recognized as an attribute of nonlinearity and unstableness. Statistical and probabilistic models cannot accurately capture the characteristics of meteorological and hydrological data for drought monitoring and are computationally expensive to handle complex and heterogeneous data sets (Prodhan et al., 2022). However, machine learning (ML) has shown excellent performance when it comes to modeling nonlinear and dynamic time series. Machine learning models can perform preprocessing and data preparation tasks. As a complex and devastating natural hazard, drought monitoring is of the utmost importance. The ability to monitor and forecast drought is essential for planning and decision-making. Mitigation planning and adaptation strategy development are primarily dependent upon the efficiency of drought forecasting models or methods. In this context, ML can provide accurate and efficient drought forecasting and can be applied to drought disaster risk management.

The purpose of this project is to predict drought using meteorological data. Various Random Forest models will be trained and compared to a baseline neural network model.

Data Presentation

About the data sources

We will be looking at the US Drought Prediction Data Set from Kaggle, a resource providing interesting data sets. The data are split into three parts:

- train_timeseries.csv is the training set, which contains data through from 2000 to 2016
- valid_timeseries.csv is the validation set, which contains data from 2017 to 2018
- test_timeseries.csv is the test set. It contains data from 2019 to 2020

The reason I chose this dataset is that it is very comprehensive. The purpose of this study is to determine whether droughts can be predicted using only meteorological data in hot climate counties, potentially leading to generalizations of US predictions to other regions of the world with different climates.

To split the data by climate zone, I took another dataset found on GitHub, that gives climate for each county in the US.

Variable to predict and features

The variable to predict is the US drought monitor. It is a measure of drought across the US manually created by experts using a wide range of data. This is a classification dataset at a county level over six levels of drought, based on the US Drought Monitor's Classification. The classification is as followed [Appendix 1] :

- 0 : No drought
- 1 : D0=Abnormally dry
- 2 : D1=Moderate drought
- 3 : D2=Severe Drought
- 4 : D3=Extreme drought)
- 5 : D4=exceptionally dry

Given that we are attempting to predict a drought category, we are exploring a multiclass classification problem.

Each entry is a drought level at a specific point in time in a specific US county, accompanied by 18 meteorological indicators: 8 indicators on wind, 7 indicators on the temperature, 1 indicator on the humidity, 1 indicator on the pressure level, and 1 indicator on the precipitation [Appendix 2].

Evaluation Metrics

We have different scores to get an estimation of the quality of a classifier. The most used for problems of our type are the accuracy, the recall and the precision. Overall, the recall and precision scores are in general the most informative ones regarding the quality of a classifier. These two scores are often aggregated into a single F1-score. F1 score is often preferred. It is a number between 0 and 1 and is the harmonic mean of precision and recall. It maintains a balance between the precision and recall for your classifier. Our goal is to get the F1-score value as close to 1 as possible but we will have a look at the other metrics to draw better conclusions.

Preprocessing data for modeling

Preliminary analysis

The validation and the train sets are combined in one data frame, *df_raw*, in order to tackle the processing task at once.

The key fields in the data frame are:

- *fips*: the unique identifier of a US County
- *score*: the US Drought Monitor Indicator (the value to predict).
- *date*: the date the weather data were collected

Summary of the initial findings:

- There are 21,569,520 rows (entries) and 21 columns (features)
- There are missing values only for score feature, as it is mentioned that only weekly data is available.
- The date is in the object type. We'll have to format it into a datetime object to be able to work with it.
- The score is in float type. We need to format it as a category.

Data processing

In order to make the data usable, it must first be cleaned. Preprocessing steps included:

- Convert the date column to a datetime type
    - To convert our date column into a datetime data type, we used the parse_dates parameter.
- To enrich our data, create new columns from the date column.

    - We are seeking to make a time-based prediction. The more information we have about the dates, the better our algorithm will be able to learn. The attribute .dt is used to provide additional information about the date.
- Fill in our missing values in the score column and convert them to integers.
    - There are only a few missing values in the score feature. The level of drought is reported on a weekly basis. The lines without the weekly results are deleted.
    - The column is transformed into categorical data by using astype('category').
- Separate the data by date and climate zone.
- Normalize with a standardization method (remove the mean and scale to unit variance)

Machine Learning Model

Choice of the model & Fine-Tuning

Vanilla Neural Network

I first trained a Neural Network.

```
# set hyperparameters
n_neuron       = 50
activation     = 'relu'
num_epochs     = 40
learning_rate  = 0.0001
minibatch_size = 64
```

```
nn_model = Sequential()

nn_model.add(Dense(n_neuron,  activation=activation,input_shape=(X_train.shape[1],))) #  the 1st hidden layer
nn_model.add(Dense(n_neuron,  activation=activation)) # the 2nd hidden layer
nn_model.add(Dense(n_neuron,  activation=activation)) # the 3rd hidden layer
nn_model.add(Dense(y_train.shape[1],  activation='softmax')) # the output layer


nn_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['acc'])
```

Parameters for the vanilla Neural Network model

I tried different parameters but overall, the same results were obtained. The Neural Network model didn't perform well as the F1 score on the testing data set is 0.3066.

1st Random Forest

I then trained a fine-tuned Random forest by running a grid search and to fine the best paramerts.

```
random_grid1 = {'n_estimators': [15,20,30,45,60],
                'max_features':[None,'sqrt'],
                'max_depth':[None,10, 20, 25],
                'min_samples_leaf':[3,7,10,16],
                'max_samples':[None,0.5,0.25],
                'min_samples_split':[2,5,10,15,25],
              }
```

```
class1 = RandomForestClassifier(random_state=42)

rf_random = RandomizedSearchCV(estimator = class1, param_distributions = random_grid1,
                               n_iter = 5, cv = 3, verbose=2, n_jobs = -1)

%time rf_drought = rf_random.fit(X_train,y_train)

print("The best hyperparameters: \n",rf_drought.best_params_)
```

Random grid search for fine-tuning

```
model_rf = rf_drought.best_estimator_
model_rf
```

```
RandomForestClassifier(max_features='sqrt', min_samples_leaf=3, n_estimators=60,
                       random_state=42)
```

1st Random Forest model

The results were much better as I obtained a F1 score of 0.7588.

2nd Random Forest with feature selection

As the Random Forest performs very well, I was interested to know how it is making its predictions. The best way to see this is with feature importance [Appendix 3]. The most important feature is the year. Overall, temperature predictors play a big role in the model performance while wind predictors seem to be less important.

I then decided to do a new fine-tuned Random Forest model, by discarding the less important features.

```
to_keep = feat_importance[feat_importance.imp > 0.02].cols
len(to_keep)
```

```
14
```

Final number of kept features for the new model

I loaded a new grid search that generated new candidates from a grid of parameter values.

```
model_rf_features = rf_drought4.best_estimator_
model_rf_features
```

```
RandomForestClassifier(max_features='sqrt', min_samples_leaf=10,
                       min_samples_split=5, n_estimators=30, random_state=42)
```

New Random Forest model with the most important features after fine-tuning

A 0.68 F1-Score is obtained for the new model. We can conclude that keeping only the best features doesn't improve the model.

3rd Random Forest with under-sampling

During the pre-processing I realized that my classes within my training samples were really unbalanced. We have indeed more days without droughts than with extreme droughts.

```
1  np.unique(y_train,return_counts=True)
```

```
(array([0, 1, 2, 3, 4, 5]),
 array([241895,  83261,  64459,  49965,  29646,  15076]))
```

Number of data by class in the training set

Class-imbalanced data arise from many fields of recent scientific discoveries where the prevalence is typically very low (Guo et al., 2017). As these events are rarely observed in daily life, the prediction task suffers from a lack of balanced data. To date, class-imbalanced learning is a relatively new challenge.

Does this unbalance avoid the model to perform correctly on less represented category? Can we improve the prediction if we balance the training data?

To address the problem of imbalance, the first strategy is the "under-sampling". This technique removes examples from the training dataset that belong to the majority classes. I import the imblearn library and use Random Under Sampler technique, which randomly removes examples.

```
1  from imblearn.under_sampling import RandomUnderSampler
```

```
1  rus=RandomUnderSampler(sampling_strategy='auto',random_state=42)
2  X_train_new,y_train_new=rus.fit_resample(X_train,y_train)
3  print(X_train_new.shape,y_train_new.shape)
```

```
(90456, 24) (90456,)
```

Under-sampling method

After creating a new random grid, the following Random Forest model was found to be the most effective:

```
model_rf_balancedcat = rf_drought2.best_estimator_
model_rf_balancedcat
```

```
RandomForestClassifier(max_features='sqrt', min_samples_leaf=7, n_estimators=70,
                       random_state=42)
```

Random Forest with an under-sampled training data set

The F1-score is 0.53. Removing examples doesn't improve the prediction.

4th Random Forest with over-sampling

The other strategy is "over-sampling". It consists in increasing the examples from the training dataset that belong to the minority classes. I imported the SMOT technique from the imblearn library, which creates synthetized examples from the existing ones.

Finally, a new Random Forest was fine-tuned from this new training dataset.

```
from imblearn.over_sampling import SMOTE
```

```
smote=SMOTE(sampling_strategy='auto',random_state=42)
X_train_new,y_train_new=smote.fit_resample(X_train,y_train)
print(X_train_new.shape,y_train_new.shape)
```

```
(1451370, 24) (1451370,)
```

Over-sampling method

```
model_rf_balancedcat = rf_drought2.best_estimator_
model_rf_balancedcat
```

```
RandomForestClassifier(max_features='sqrt', max_samples=0.5,
                       min_samples_leaf=16, min_samples_split=15,
                       n_estimators=20, random_state=42)
```

Random Forest with an over-sampled training data set

The F1-score is 0.57, which better than the under-sampling model but still not as good as the baseline Random Forest model.

Results & Discussion

	Neural Network	Random Forest	Random Forest Under Sampling	Random Forest Over Sampling	Random Forest Feature importance
Accuracy	0.18	0.63	0.53	00.53	0.60
Recall	0.18	0.63	0.53	0.53	0.60
Precision	0.99	0.94	0.55	0.63	0.80
F1 Score	0.30	0.76	0.53	0.57	0.68
Running Time	4min24s	9min 50s	1min 12s	7min 27s	2min 46s

Accuracy, Recall, Precision, F1 Score and Running Time of the four models

The prediction using meteorological data is promising (F1-score: 0.76). Precision and recall, however, are unbalanced (Recall: 0.63, Precision: 0.94). In other words, our baseline Random Forest returns very few results, but most of the predicted labels are correct when compared to the tested labels. This is due to the unbalanced data in our training dataset. As a matter of fact, a good balance between precision and Recall is particularly important when it comes to imbalanced datasets (Guang-Hui Fu et al. 2018). The objective is both high precision and high recall, with a trade-off between the two.

We can observe that re-sampling methods provide a better precision-recall trade-off, since they provide a better balance between recall (0.53 and 0.53) and precision (0.55 and 0.63). In spite of this, the overall scores are lower.

Conclusion

In this paper, I examined drought prediction using meteorological data in hot climate counties in the United States. The results obtained with Random Forest are very promising, as the F1-score was 0.76. There was, however, an imbalance between precision and recall. It's because the training dataset is imbalanced. Unbalanced datasets are currently the subject of an extensive literature review (Guo et al., 2019). Resampling of training sets can provide a better trade-off, despite the fact that we obtain lower scores. A further step would be to improve both precision and recall equally and to monitor the PRC (precision-recall curve) on the top of the F1-score.

Appendix

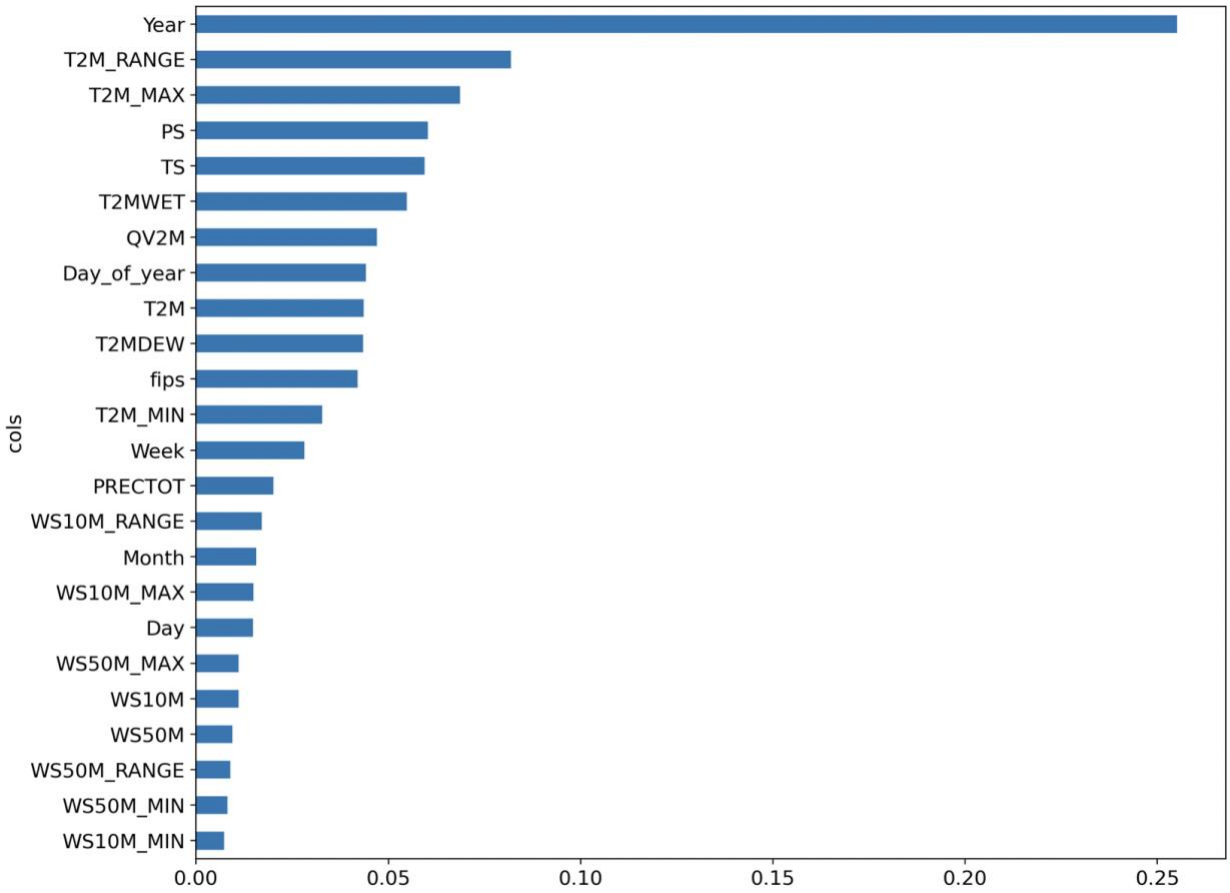
Appendix 1: Categorization of drought level (US Drought Monitor)

Category	Description	Possible Impacts
D0	Abnormally Dry	Going into drought: ▪ short-term dryness slowing planting, growth of crops or pastures Coming out of drought: ▪ some lingering water deficits ▪ pastures or crops not fully recovered
D1	Moderate Drought	▪ Some damage to crops, pastures ▪ Streams, reservoirs, or wells low, some water shortages developing or imminent ▪ Voluntary water-use restrictions requested
D2	Severe Drought	▪ Crop or pasture losses likely ▪ Water shortages common ▪ Water restrictions imposed
D3	Extreme Drought	▪ Major crop/pasture losses ▪ Widespread water shortages or restrictions
D4	Exceptional Drought	▪ Exceptional and widespread crop/pasture losses ▪ Shortages of water in reservoirs, streams, and wells creating water emergencies

Appendix 2 : Variable Description

Indicator	Description
WS10M_MIN	Minimum Wind Speed at 10 Meters (m/s)
QV2M	Specific Humidity at 2 Meters (g/kg)
T2M_RANGE	Temperature Range at 2 Meters (C)
WS10M	Wind Speed at 10 Meters (m/s)
T2M	Temperature at 2 Meters (C)
WS50M_MIN	Minimum Wind Speed at 50 Meters (m/s)
T2M_MAX	Maximum Temperature at 2 Meters (C)
WS50M	Wind Speed at 50 Meters (m/s)
TS	Earth Skin Temperature (C)
WS50M_RANGE	Wind Speed Range at 50 Meters (m/s)
WS50M_MAX	Maximum Wind Speed at 50 Meters (m/s)
WS10M_MAX	Maximum Wind Speed at 10 Meters (m/s)
WS10M_RANGE	Wind Speed Range at 10 Meters (m/s)
PS	Surface Pressure (kPa)
T2MDEW	Dew/Frost Point at 2 Meters (C)
T2M_MIN	Minimum Temperature at 2 Meters (C)
T2MWET	Wet Bulb Temperature at 2 Meters (C)
PRECTOT	Precipitation (mm day-1)

Appendix 3: Most important features for the first Random Forest Model



References

1. National Drought Mitigation Center, 2022, *Drought Basics*
2. Vatter, 2019, *Drought Risk, The Global Thrist for Water in the Era of Climate Crisis.*
3. Prodhan et al., 2022, *A review of machin learning methods for drought hazard monitoring and forecasting: Current research trends, challenges, and future research directions*
4. Guo et al., 2017, Learning from class-imbalanced data: Review of methods and applications
5. Guang-Hui et al., 2018, Tuning model parameters in class-imbalanced learning with precision-recall curve
6. Guang-Hui et al, 2017, Stable variable selection of class-imbalanced data with precision-recall criterion