# Prediction Of Droughts Using Machine Learning Algorithms

By

Maximilian See Tze Jie

2102869A

A Report

Submitted to School of Applied Artificial Intelligence

Temasek Polytechnic IIT

For Diploma In AAI

July 2022

# Content Page

1. Introduction
2. Data Exploration & Pre-processing of data
3. Model Development
4. Methods and Improvement
5. Results & Analysis
6. Conclusion
7. References

# Assumptions

In this report only the soil and weather data are considered. Anomalies that are not in the data set for examples hurricanes and floods are not and will not be considered in this report. Note that the development for the most part follows the CRISP-DM framework. Note that the below EDA are done with the full dataset (2 million + rows), whilst the Machine Learning part mostly consist of the 20000 rows of data. The last part under the three best algorithms the dataset that they are using is the partitioned dataset train\_timeseries-5.csv (267K rows). If a sample is required for testing the Flask application, please refer to the dataset in the ZIP file.

# Abstract

The pandemic, coupled with climate change and the Ukraine-Russian war, possess as an ever-increasing threat to many developed and underdeveloped countries worldwide. To cushion the effects and help with the crisis, I have focused my research on the agricultural sector, where most products such as wheat and grain are grown. I have decided to work on predicting the severity of droughts. This is to better prepare farmers for droughts by allowing them to make more informed decisions based on the data and prediction through weather and soil data collected. Our approach consists of first understanding the data through Exploratory Data Analysis, after which we would pre-process and transform the data to aid in model developments. Finally, we would conclude with the results and analysis. The code is available in the Zip file.

Introduction

From wars to pandemic to climate change, food security is a complex problem where there are many causes with no clear solutions. As such in this paper we are only going to explore one of the main causes of food insecurity, droughts and predict the severity of them.

The solutions today with regards to droughts prediction include complex and expensive equipment. These solutions include satellite imagery and super computers which might be an expensive investment for less developed regions like Africa. Moreover, even if less developed countries could afford such technologies, they might not have the expertise or manpower to operate and maintain it. For example, a typical weather satellite alone cost around 290 million to operate. This does not consider for the launch cost, technological cost, and manpower cost. For context, the GDP of the Democratic Republic of the Congo is $17.87 Billion. The satellite alone would be too expensive for the country to handle. Thus, while the solution might prove to be highly Inaccurate it is too expensive to be implemented.

The goal here or proposed solution in this paper aims to leverage on the increasing technological prowess to better prepare farmers around the world, at a cheaper cost using simple machine learning described in this paper, against droughts. The solution aims to utilise smartphones (Google Collab allows for ML codes to be run on their cloud platform) and consumer computers to run machine learning algorithms to predict the severity of droughts. This solution is especially effective for lesser developed countries as it opens the opportunity for farmers to use machine learning to predict the severity of droughts. The computers could be funded by NGOs and the government itself as it is a cheaper alternative. The solution is also easy to interpret and use, without needing farmers to learn how to operate and interpret it.

Understanding The Dataset Of Choice

The dataset of choice would be agricultural data focused on soil quality and weather conditions. The dataset is available on Kaggle and is provided by Christophe Minixhofer. The data consist of both soil and weather data which is critical in our research and modelling. The essence of our acquired data is mostly quantitative with certain values such as Soil Quality and *Score* (Our Target Variable) being qualitative. The dataset has been finalised about a year ago and consist of 3108 of 3110 counties of the United States.

The description for weather condition is as follows below in the appendix:

|  |  |
| --- | --- |
| **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) |

The description for the soil data could be found below:

|  |  |  |
| --- | --- | --- |
| Soil Qualities | | Soil Characteristics |
| SQ1 | Nutrient availability | Soil texture, soil organic carbon, soil pH, total exchangeable bases |
| SQ2 | Nutrient retention capacity | Soil Organic carbon, Soil texture, base saturation, cation exchange capacity of soil and of clay fraction |
| SQ3 | Rooting conditions | Soil textures, bulk density, coarse fragments, vertic soil properties and soil phases affecting root penetration and soil depth and soil volume |
| SQ4 | Oxygen availability to roots | Soil drainage and soil phases affecting soil drainage |
| SQ5 | Excess salts. | Soil salinity, soil sodicity and soil phases influencing salt conditions |
| SQ6 | Toxicity | Calcium carbonate and gypsum |
| SQ7 | Workability (constraining field management) | Soil texture, effective soil depth/volume, and soil phases constraining soil management (soil depth, rock outcrop, stoniness, gravel/concretions, and hardpans) |

Severity Index/Score

|  |  |  |
| --- | --- | --- |
| Category | Description | Possible Impacts |
| D0 | Abnormally Dry | Going into drought; short-term dryness slowing plating, growth of crops or pastures |
| D1 | Moderate Drought | Some Damage to crops, pastures. Streams, reservoirs, or wells low ,some water shortages developing of imminent . Voluntary water-use restrictions requested |
| D2 | Severe Drought | Crop or pasture losses likely. Water shortages are common  Water restrictions imposed |
| D3 | Extreme Drought | Major crop/pasture losses. Widespread water shortages |
| D4 | Exceptional Drought | Exceptional and widespread crop/pasture losses . Shortages of water in reservoirs streams and wells creating water emergencies |

# Data Exploration & Pre-processing of Data

## I. Merging Datasets

The dataset downloaded was split into Train, Validation, Test, and soil\_data. Since Train has 47% of the total weather data it is chosen. The research focuses on how soil quality and weather conditions affect the severity of droughts, as such the dataset with the soil quality will be needed. To do that we will be combining both dataset together: soil\_data and Train using a pd.merge() command. Doing so will join all data that has the same FIPS (county) code. The new dataset would be named df. We will then drop all irrelevant data that is not directly related to soil quality. The data shape is now (262367,32).

Text

Description automatically generated

## II. Preliminary Exploration

In this section we would be exploring the shape, columns, and number of data missing. This allows us to gain a general feel of the data and think about the ways to explore and present the data. Such commands include:

A picture containing text

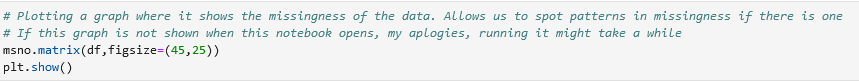
Description automatically generated

After the execution of the code, it tells us that the dataset has 19300680 rows and 21 columns. It also tells us that the dataset consist of a large number of missing values in the column *score* at 2142914 missing values. The .info() tells us that the dataset consist of mostly float types and that all of the data is represented in numerical format of float or int.

## III. Data Cleaning & Handling

Removal Of Null Values

Our Preliminary Exploration has shown that there is many missing values on *score*. Upon further calculation the amount of missing *score* values range at more than 85%. This is not viable as such number of missing values is detrimental to our modelling. Furthermore, since *score* is our target variable, it is not allowed to remove the *score* column. To find the exact reason to the missingness, the MSNO library is summoned. The MSNO library tells and shows us the missingness of the data through a chart.



The MSNO chart has shown a peculiar pattern. The pattern could be inferred from the *score* column. The chart shows that there is an equal interval between the missingness of the data. This means that the missingness is likely MNAR. MNAR means there might be a reason as to the missingness. To find that reason we will infer from the results gathered from Preliminary Exploration. In Preliminary Exploration we have executed the .head() command which allows us to see the first 5 rows of the dataset. We will now increase the value to 21 to observe if there is any patterns to the missingness. As we can see there is an equal interval of missing data. For every 1 present data the next 6 rows would see null values. Upon further research I have found that the *score* is taken only once a week. This explains the missingness of the values and concludes our investigation.

Handling Outliers

To find outliers in the dataset, a boxplot was used to plot the outliers within the respective columns. A function which removes the outliers was then conceived and implemented. After the removal of the outliers the data shape stands at 1942145. Outliers can remarkedly affect our models and can be a valuable source of information. Some outliers, however, lie outside of the IQR range and should be removed as it might affect modelling later on.

Date, Time & Score Handling For Easier Exploration and Deeper Insights

To facilitate for more in-depth research and insight the *date* will be split into three columns, year, month, and day. This increases our flexibility as we could now view and group our data within specific months, days or years and analyse it.

## IV. Exploratory Data Analysis

EDA

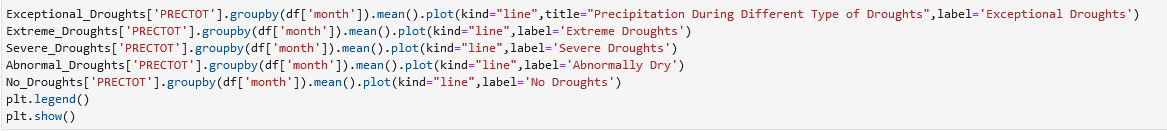
After Cleansing & Handling the data, Exploratory Data Analysis will be conducted. To start, the Weather Conditions will be plotted against the Severity of the Drought. The main goal is to find if there is any difference between the variable of Weather Conditions in different drought severities. This will be assessed by filtering the data into their respective Severity. We will then analyse the mean of Precipitation (PRECTOT), Wind Speed (WS50M, WS10M), Humidity (QV2M), Earth Skin Temperature (TS) and so on during the different types of droughts, group them by their months and plot it via a line graph. We would then observe if there were any significant change between the variables of differing drought severity. For Soil Quality we would be plotting them via a geo plot. This will be done via the *longitude* and *latitude* given in the Soil data csv. This is done so as to understand where the droughts are the most severe. Below are some examples:

Filtering The Dataset into Their Respective Groups

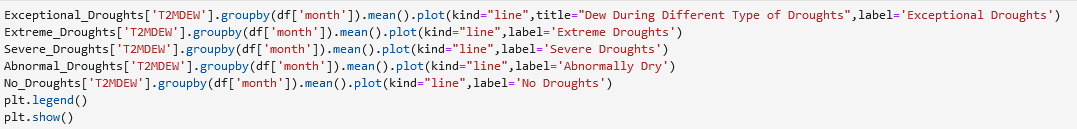
Text

Description automatically generated

Analysis of Exceptional Droughts



Analysis Of Extreme Droughts



Analysis of Drought locations

Text

Description automatically generated with low confidence

## V. Quantitative Analysis

VIF

**Variance inflation factor (VIF)** is a measure of the severity of multicollinearity in a set of multiple regression variables (OLS). VIF provides an index that measure how much the variance of a regression coefficient is increased because of collinearity. A high VIF Value indicate that the independent attribute is highly collinear with other attributes in the model (Potters, 2021). In other words, we are no longer looking at bivariate relationship but multivariate instead which allows us to create an auxiliary regression for each X variable. We will then regress the selected variable for say X1 on the other variables in the dataset, it effectively figures out how X1 is being explained by the other variables. The equation below describes how VIF works

Y = β0 + β1 X1 + β2 X 2 + ... + βk Xk + ε

X1 = β0 + β1 X 2 + ... + βk Xk + ε

We would then find the VIFs using the R-squared from each regression...

Usually, VIF of 10 and above are highly colinear and strategies should be implemented to deal with it. As we can see the VIF is exceptionally high for each variable. However, since all variables have high VIF values, we cannot remove them all or combine them. As such, we will rely on more robust algorithms such as the RFECV and RandomForest’s feature\_importances\_ to find out which variable to keep and which to discard since we are using Random Forest later on. We will also be using XGB to find our which features to keep and which to discard. But first we are going to standardise the data from Spearman Correlation onwards.

Chi Square

**Pearson's chi-squared test** is used to determine whether there is a significant difference between the expected frequencies and the observed frequencies in one or more categories. (Hayes,2022). Pearson's Chi Square test allows us to test the significance of a variable against the target variable. Thus, while the Pearson's correlation and Spearman's rank correlation measure the strength between two categorial variables, the chi-square test measures the significance of association between the two variables. Essentially it tells us whether the relationship we found is likely to exist in the population. (Juhi Ramzai, 2020). In our case we would be testing categorial variables in our dataset towards the *score* variable. If the Null hypothesis cannot be rejected, then the variable will be dropped.

Whilst correlation analysis is typically used. It is mostly and only used to measure the association between quantitative variables. Our Soil Quality column however are categorial variables which are ordinal in nature. To compare these ordinal variables, the data could be summarised into what is called a contingency table which lists the options for one variable as the columns and one as the rows. Chi square will then be calculated by the scipy.stats library which also calculates the P-value.

Pearson’s Correlation

**Pearson’s correlation coefficient** is the test statistics that measures the statistical relationship, or association, between two continuous variables (Laerd Statistics, 2020).  It is known as the best method of measuring the association between variables of interest because it is based on the method of covariance.  It gives information about the magnitude of the association, or correlation, as well as the direction of the relationship.

This dataset consists of both numeric and categorial data. For example, precipitation and humidity lies under numeric features while Soil Quality lies under categorial data. As such, Pearson’s correlation cannot be applied to all variables in the dataset. Thus, instead of comparing the strength of all variables, we would instead be comparing only the numerical variables using Pearson’s correlation.



Spearman Correlation

**Spearman’s Correlation** is a rank-order correlation that is the nonparametric version of Pearson’s correlation. The Spearman correlation between two variables is similar to the Pearson correlation between the rank values of two variables; while Pearson’s correlation assesses linear relationships, Spearman correlation measures monotonic relationships. ([StatisticsHowTo](https://www.facebook.com/statisticshowto), ,n.d.) Spearman’s coefficient is appropriate for both continuous and discrete ordinal variables, thus being suitable for the dataset. In the application below, we are going to first compare the categorial variable of Soil Quality. Then we are going to apply the Spearman’s Correlation to the whole dataset.



VI. Scaling

Standard Scaler

**Standard Scaler** assumes features have a normal distribution and will scale them to have a mean 0 and standard deviation of 1. Unlike Min-Max or Max-Abs scalers, the Standard scaler doesn’t have a predetermined range to scale to. (Jackson,2019). Since our dataset has mostly normal to slightly skewed distributions standard scaler would be used as it works well with data that is normally distributed. Standard Scaler is also a popular option to use, thus it is chosen. Additionally, deep learning algorithms usually go for zero mean and unit variance. Regression type algorithms is a key example where it benefits from normally distributed data. (Hale, 2019). MinMaxScaler which is another popular scaling method was not used as the upper and lower boundaries are not known. It is also prone to outliers. Thus, MinMaxScaler was not used.

Text

Description automatically generated

One Hot Encoding

**One Hot Encoding** is a process whereby categorial variables are converted into a form that could be provided to machine learning algorithm for easier interpretation (by the model). Since our dataset contains categorial variables (SoilQuality) we can create features for each *SQ* column. After the application of One Hot Encoding, we now stand at 60 columns. Applying One Hot Encoding would not affect our curse of dimensionality as our dataset has over 2 million rows of usable data. One Hot Encoding is beneficial as it improves predictions and classification accuracy of a model. (Dey, 2021).

# Model Development

Decision Tree Algorithms Along with KNN & Regression

**Decision Tree Algorithms** would be mainly used for this dataset Decision Trees are supervised learning methods used for classification and regression. Other complimentary algorithms such as KNN (Supervised) and Ensemble Learning Methods are chosen to ensure better representation. Our main goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the given data. Decision trees and KNN consist of a few hyperparameters. The models will first use the default settings for the hyperparameters.

Random Forest

**Random Forest** or random decision forest is an ensemble learning method for classification and regression. Random forest uses multiple decision tree and operates as an ensemble class algorithm. Each individual tree in the random forest will have a prediction and the class with the most votes becomes the model prediction.

The core concept and strength of random forest lies in its ability to draw predictions from the multitude of decision trees created.

KNN

**KNN** is a supervised machine learning algorithm for classification and regression problems. KNN works by finding the distance between a point and all the examples in the data. After which it would select the specified number of example closest to the point. For classification it would find the most frequent label. For regression it would average the label. A simple analogy that describes KNN is: “Show Me Your Friends And I’ll Tell You Who You Are”. (Harrison, 2018)

CatBoost

**Catboost** is an ensemble learning method for classification and regression. In comparison to Random Forest, CatBoost which uses Gradient Boosting, in theory, would perform better than its counterpart. As a start a rough estimate of 6500 iterations would be implemented with a learning rate of 0.1.

XGB

**XGB** or **Extreme Gradient Boosting** is an ensemble learning algorithm which uses Gradient Boosting algorithm. It is built for high scalability and flexibility, which is essential in learning our massive dataset. XGB is also suitable for datasets containing both numerical and categorial features.

LightGBM

**LightGBM** is a gradient boosting framework based on decision tree algorithm used for classification and regression. As it is based on decision tree algorithms it splits the tree leaf wise whereas other algorithms split the tree depth or level wise rather than leaf wise. As the leaf grows in LGBM the leaf-wise algorithm can reduce more loss than the level wise algorithm therefore providing better accuracy.

Results for all unsampled algorithms

|  |  |  |  |
| --- | --- | --- | --- |
| Model Performance | Accuracy | Cohen Kappa | F1 |
| Extreme Gradient Booster | 0.763 | 0.667 | 0.754 |
| Random Forest Classifier | 0.691 | 0.553 | 0.672 |
| K-Nearest Neighbour | 0.634 | 0.479 | 0.619 |
| CatBoost Classifier | 0.745 | 0.641 | 0.736 |
| LightGBM | 0.775 | 0.684 | 0.765 |

# Methods & Improvement

Up Sampling & Down Sampling with SMOTE & KNeighbourhoodCleaner

Our dataset consists of imbalance data with regards to *score*. As observed above in EDA we can conclude that data for No Drought out numbers all other types of droughts. The challenge of working with imbalanced datasets is that most machine learning techniques will "ignore", the minority classes, resulting in lower accuracy or wrong interpretations for and in our predictions. (Brownlee,2020). As such, SMOTE would be used to address the problem by Up sampling the data for the minority columns.

We shall also try Down sample with *KNeighbourhoodCleaner* as we have enough rows and columns to do so. In essence we are reducing the number the number of records of all samples to that of the minority class record count. It is not recommended to Down sample data, but we will continue forward with Down sampling as experimentation.

GridSearchCV

A Grid Search works by going through a range of settings that you want to try for each parameter. It trains and tests the model using every combination of parameters. The combination of parameters that generates the best predictions will be the set of parameters that we will use for our model. As for the metric regarding the selection of hyperparameters, *F1\_Score* would be chosen instead of *accuracy* this is because *accuracy* is not a good metric to use when we have class imbalance. This is because *accuracy* is unable to distinguish between specific type of errors like false positives and false negatives. *F1\_Score* in contrast can not only the number of prediction errors that our model has made but also the type of errors (False positive or false negatives) made. Thus, *F1\_Score* is chosen.

Feature Selection With RFECV

Recursive Feature Elimination with Cross Validation RFECV is a feature selection technique that selects the best subset of features using recursive elimination. The best subsets are selected based on the cross-validation score of the model.

RFECV is a more robust technique of knowing which features are important towards *score*. By knowing which features are of more significance, we are better able to know which features affect the accuracy of the model, thus giving us the ability to correct our techniques for handling the values within the columns that are important. Such use cases involve in improving the accuracy of the model where it is an iterative process. (Verma, 2020)

# Results & Analysis After Remediation

Before SMOTE and GridSearchCV| Before Remediation

Text

Description automatically generatedTable

Description automatically generated with low confidenceTable

Description automatically generatedText

Description automatically generated with medium confidence

Model with no improvements

Model with SMOTE & NCR | After Remediation

SMOTE

|  |  |  |  |
| --- | --- | --- | --- |
| Model Performance | Accuracy | Cohen Kappa | F1 |
| Extreme Gradient Booster | 0.763 | 0.670 | 0.756 |
| Random Forest Classifier | 0.702 | 0.584 | 0.6952 |
| K-Nearest Neighbour | 0.633 | 0.516 | 0.643 |
| CatBoost Classifier | 0.758 | 0.667 | 0.755 |
| LightGBM | 0.768 | 0.678 | 0.763 |

K-NeighbourhoodCleaner

|  |  |  |  |
| --- | --- | --- | --- |
| Model Performance | Accuracy | Cohen Kappa | F1 |
| Extreme Gradient Booster | 0.678 | 0.678 | 0.655 |
| Random Forest Classifier | 0.609 | 0.405 | 0.561 |
| K-Nearest Neighbour | 0.573 | 0.379 | 0.542 |
| CatBoost Classifier | 0.671 | 0.524 | 0.647 |
| LightGBM | 0.688 | 0.667 | 0.550 |

Model with SMOTE & GridSearch | After Remediation

With a lower decrease in overall performance, SMOTE has proven to be the better alternative when sampling the data. As observed, SMOTE produced a better Cohen Kappa score with higher accuracies than the KneighbourghoodCleaner. SMOTE would be used along with GridsearchCV to produce the best result. Existing hyperparameters of that model will be used, referenced, and tweaked to increase overall performance.

Along with LGBM, XGB gained over 90% accuracy with the partitioned dataset (267K) , With over 91.5% accuracy and over 90% score for both Cohen Kappa and F1, XGB & LGBM is a suitable model. It is of surprise that CatBoost did not perform well as it gained mediocre scores across the board. The model with 267K row of data will not be selected as it could not be loaded

Text

Description automatically generatedGraphical user interface, text, application

Description automatically generated

XGB Model with partitioned dataset that is unable to load due to size and memory usage

Since the file could not be loaded the XGB would be run again with the smaller subset data with about 20000 rows. Over here we lose quite a bit of accuracy, F1 and Copen Kappa score. We instead got an 81.25% accuracy for our smaller dataset which is still relatively better (about 5%). Thus, the model with the smaller

dataset would be chosen.

Chart, bar chart

Description automatically generated

Models with SMOTE and GridSearchCV that is unable to load

Table

Description automatically generated with medium confidence

Model that will be chosen XGB (20K)

All but the columns of *date, Severity and fips* will not be included or used in the prediction.

# Model Deployment

The model will be deployed via Flask. The model will consist of a webpage which has all options available for the farmers to place in their data. The following is the look of the website. The 30 columns below are for the inputs.

Graphical user interface, application

Description automatically generated

Webpage Design

# Conclusion

The paper above has presented the steps taken as well as the findings and insights from the dataset. We have also gone into detail on why such steps were taken (e.g., Cleaning of Data) as well as how we have decided to display the data for maximum insights with relation to our target variable score. In this paper we have also explained our rational behind the algorithms we are using. We have explored multiple kinds of algorithms such as the K-nearest neighbour method as well as decision tree ensemble methods against our dataset. Engineering methods for imbalanced dataset SMOTE *Up Sampling* and Neighbourhood Cleaning *Down Sampling* have been implemented and SMOTE prove to be beneficial to our dataset. Hyperparameter optimisation for increased accuracy have been implemented. In comparison to the dataset without Sampling, the dataset with sampling performed no better, however with GridSearchCV and the partitioned dataset (267K) it managed to gain over 90% accuracy. The subset data (20K) was able to gain over 80% accuracy. Lastly, we deployed the model on Flask where the user could fill in the data and get a prediction result.

END

References

A. (2022, May 7). Drought Prediction Using ML Algorithms. Kaggle. <https://www.kaggle.com/code/akshayasrinivasan2/drought-prediction-using-ml-algorithms>

Common parameters. (n.d.). Training Parameters | CatBoost. <https://catboost.ai/en/docs/references/training-parameters/common>

Fraj, M. B. (2018, August 14). In Depth: Parameter tuning for Gradient Boosting - All things AI. Medium. <https://medium.com/all-things-ai/in-depth-parameter-tuning-for-gradient-boosting-3363992e9bae>

Harrison, O. (2019, July 14). Machine Learning Basics with the K-Nearest Neighbors Algorithm. Medium. <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>

Lani, J. (2021, June 9). Pearson’s Correlation Coefficient. Statistics Solutions. <https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/pearsons-correlation-coefficient/>

Lt, Z. (2022, February 25). Essential Things You Need to Know About F1-Score - Towards Data Science. Medium. <https://towardsdatascience.com/essential-things-you-need-to-know-about-f1-score-dbd973bf1a3>

Parameters Tuning — LightGBM 3.3.2.99 documentation. (n.d.). LightBGM. <https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html>

sklearn.neighbors.KNeighborsClassifier. (n.d.). Scikit-Learn. <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

Spearman Rank Correlation (Spearman’s Rho): Definition and How to Calculate it. (2021, July 1). Statistics How To. <https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/spearman-rank-correlation-definition-calculate/>

Which algorithm takes the crown: Light GBM vs XGBOOST? (n.d.). Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/>