**Predicting Lumpy Skin Disease**

**Introduction**

Lumpy skin disease is a viral disease that affects cattle. It is transmitted by blood-feeding insects, such as certain species. of flies and mosquitoes, or ticks. It causes fever, nodules on the skin and can also lead to death, especially in animals that that have not previously been exposed to the virus. Lumpy skin disease (LSD) is an infectious disease in cattle caused by a virus of the family Poxviridae, also known as Neethling virus. Lumpy skin disease is a contagious viral disease that spreads among cattle through mosquitoes, flies, lice and wasps by direct contact, as also through contaminated food and water.

**Dataset Description**

The dataset used for this project includes the following columns:

* Longitude (X-axis spatial coordinates)
* Continent of the outbreak
* Latitude (Y-axis spatial coordinates)
* Monthly Cloud Cover in percent
* Diurnal Temperature Range in degrees Celsius
* Country of outbreak
* Frost Day Frequency in a month
* Potential Evapotranspiration in millimetres per day
* Precipitation in millimetres per month
* Daily Mean Temperature in degrees Celsius
* Temperature in degrees Celsius
* Monthly Average Maximum and Minimum Temperature in degrees Celsius

**Project Flow:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1. Data Collection and Preparation | | |
|  | |  | * Collect the dataset from reliable sources. * Perform data cleaning and preprocessing. |
|  | 2. Exploratory Data Analysis (EDA) | | |
|  | |  | * Analyse the dataset using descriptive statistics and visualizations. * Explore the distribution of variables and identify any patterns or trends. |
|  | 3. Feature Engineering | | |
|  | |  | * Extract relevant features from the dataset. * Handle missing values and outliers, if any. * Transform categorical variables into numerical representations, if required. |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 4. Model Building | | |
|  | |  | * Split the dataset into training and testing sets. * Train various machine learning models on the training set. * Evaluate the performance of each model using appropriate evaluation metrics. * Select the best-performing model for further analysis. |
|  | 5. Model Evaluation | | |
|  | |  | * Evaluate the optimized model on the testing set. * Assess its predictive accuracy and reliability. |
|  | 6. Model Deployment | | |
|  | |  | * Deploy the final model to make predictions on new, unseen data. * Develop a user-friendly interface or API for easy access to the model's predictions. |
|  | 7. Documentation and Reporting | | |
|  | |  | * Prepare a comprehensive project report documenting the entire process. * Present the findings, insights, and conclusions derived from the project. * Provide recommendations for further improvements or future research. |

**Milestone 1: Define Problem / Problem Understanding**

**Activity 1: Specify the Business Problem**

The business problem for the accurate prediction of Lumpy Skin Disease is to develop a machine learning model that can effectively predict the occurrence of Lumpy Skin Disease in cattle. Lumpy Skin Disease is a highly contagious viral disease that affects cattle, causing significant economic losses in the livestock industry. By accurately predicting the disease occurrence, proactive measures can be taken for disease control and prevention, reducing the spread and impact of Lumpy Skin Disease.

**Activity 2: Business Requirements**

To ensure that the Lumpy Skin Disease prediction model meets business requirements and can be deployed effectively, the following rules and requirements need to be considered:

1. Accuracy: The model should demonstrate a high level of accuracy in predicting the occurrence of Lumpy Skin Disease. It should provide reliable and precise predictions to support decision-making processes related to disease control and prevention.

1. Early Detection: The model should be able to detect the presence of Lumpy Skin Disease at an early stage to facilitate timely intervention and minimize the risk of disease spread within cattle populations.
2. Scalability: The model should be scalable to handle large volumes of data and accommodate future growth in the livestock industry. It should be capable of processing data from multiple sources and adapting to evolving disease patterns.
3. Interpretability: The model should be interpretable, meaning that its predictions can be explained and understood by stakeholders. Interpretability is essential for building trust in the model and enabling informed decision-making based on its outputs.
4. Privacy and Security: The model should adhere to privacy and security regulations to protect sensitive data. Measures should be implemented to ensure secure storage, handling, and access to data used for training and prediction purposes.

**Activity 3: Literature Survey**

A literature survey for the accurate prediction of Lumpy Skin Disease would involve researching and reviewing existing studies, articles, and publications related to Lumpy Skin Disease in cattle. The survey aims to gather insights on the following aspects:

1. Disease Characteristics: Understanding the aetiology, epidemiology, and clinical manifestations of Lumpy Skin Disease in cattle. Exploring factors that contribute to disease transmission and spread.
2. Risk Factors: Identifying risk factors associated with Lumpy Skin Disease, such as breed susceptibility, age, geographical location, and environmental conditions.
3. Diagnostic Methods: Reviewing existing diagnostic methods for Lumpy Skin Disease, including clinical observations, laboratory tests, and imaging techniques. Exploring their limitations and potential for improvement.
4. Machine Learning Approaches: Investigating previous studies that have utilized machine learning techniques for disease prediction in cattle. Assessing the performance of different algorithms and feature selection methods.
5. Data Availability: Identifying potential sources of data for training and validating the prediction model. Assessing the quality, completeness, and reliability of available datasets.

The literature survey will help in gaining a comprehensive understanding of Lumpy Skin Disease, its predictive modelling approaches, and the gaps in knowledge that can be addressed through this project.

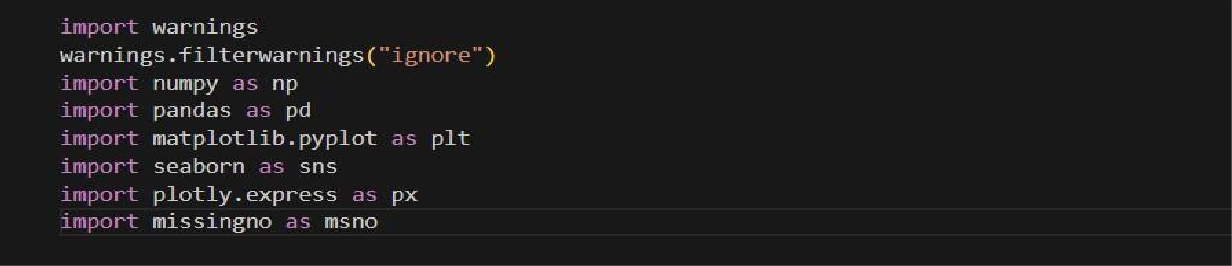
**Milestone 2: Data Collection & Preparation**

**Activity 1: Collect the Dataset**

To develop an accurate prediction model for Lumpy Skin Disease, a comprehensive dataset related to the disease and cattle characteristics needs to be collected. The dataset should include relevant features that can contribute to the prediction of Lumpy Skin Disease occurrence. The following steps should be followed to collect the dataset:

## Activity 1.1: Importing the libraries

|  |  |  |  |
| --- | --- | --- | --- |
| Utilize the necessary software frameworks and dependencies as illustrated in the | | | |
| accompanying visual representation, in order to facilitate the successful implementation of | | | |
| this | machine | learning | endeavour. |

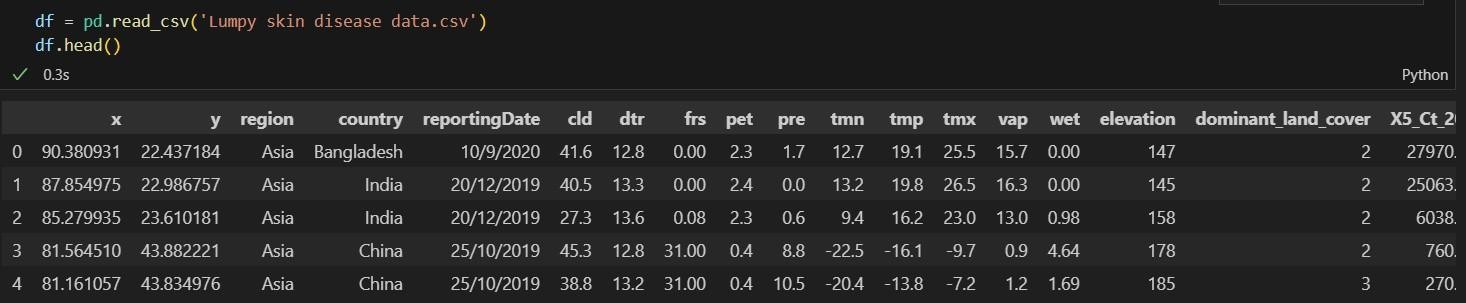


**Activity 1.2: Dataset Reading**

The dataset provided may be in various formats such as .csv, Excel files, .txt, .json, among others. To effectively process the dataset, we will employ the pandas library.

Considering that the dataset is in a CSV file format, we will utilize the pandas function read\_csv() to ingest the dataset. This function requires the directory path to the CSV file as a parameter.

To preview the initial 5 rows of the dataset, we will employ the df.head() function, which displays the desired subset of the data.



## Activity 2: Data Preparation

Data preparation, or data preprocessing, refers to the essential steps of refining, transforming, and organizing raw data prior to its utilization in data analysis or machine learning models.

The outlined activity encompasses the following steps:

* Identification and removal of missing values
* Restoring the missing values.
* Encoding categorical variables.
* Normalizing the data.

Please note that these steps serve as a general guideline for pre-processing data before its application in machine learning training. The specific pre-processing requirements may vary based on the characteristics of the dataset.

## Identification and removal of missing values.

Upon thorough examination, it has come to our attention that there exists a discernible pattern among the missing values observed in three specific variables. However, we have been unable to identify the precise reporting date within our dataset. Consequently, we have made the decision to remove the column pertaining to the reporting date.

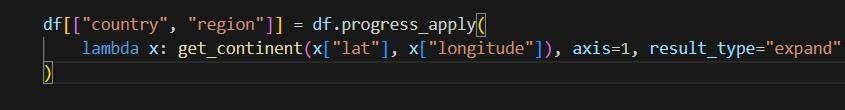
Nonetheless, after careful consideration, we have determined that the continent and countries columns bear significant importance as they play a pivotal role in exploratory data analysis, visualization, and overall model construction. Therefore, we have opted to retain these columns within our dataset, recognizing their value and relevance to our objectives.

## Restoring the missing values.

Remarkably, approximately 80% of the data contained within the continent and country columns has been identified as missing. Fortunately, we possess comprehensive information in the form of longitude and latitude coordinates. Leveraging the capabilities of the Python modules "pycountry" and "geocoder," we can utilize geospatial coordinates to derive and compute the corresponding country and continent for each data point. This approach enables us to bridge the gap in the dataset and successfully determine the missing values for the continent and country variables.



Executing the aforementioned code snippet to implement the proposed solution.



In order to enhance comprehension and facilitate better understanding, we will assign country names based on the existing country codes available in the dataset. By utilizing the

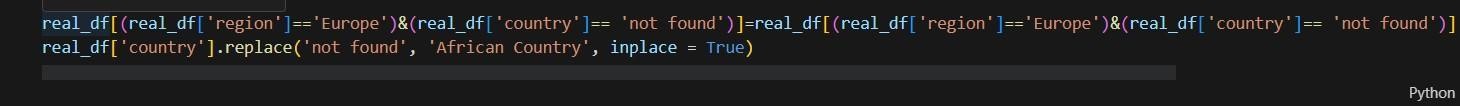
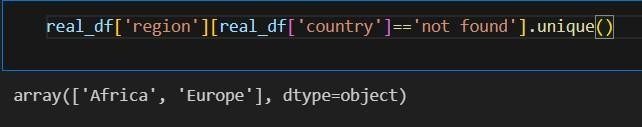
country codes as references, we can replace the country codes with corresponding country names, enabling clearer interpretation of the data.



Upon restoring a significant portion of the missing values in the two columns, a subsequent examination reveals that approximately 14% of the country names remain unresolved.



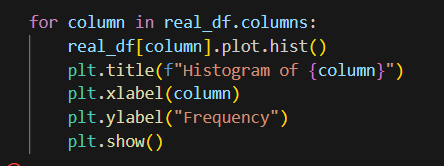
Further scrutiny has confirmed that all of these countries, except for one European country, belong to the African continent. To address this, we shall replace the remaining null values with suitable values, taking into consideration the geographic context and assigning the appropriate country names accordingly.



**Activity 2: Visual analysis**

## Activity 2.1: Univariate analysis

The code snippet presented below facilitates the generation of histograms to visualize the distribution of numerical columns, namely "wet day" and "temperatures." By employing Python's Matplotlib library, these histograms provide a graphical representation of the frequency distribution for each respective column. This aids in gaining a deeper understanding of the data's characteristics and patterns related to "wet day" and "temperatures" variables.



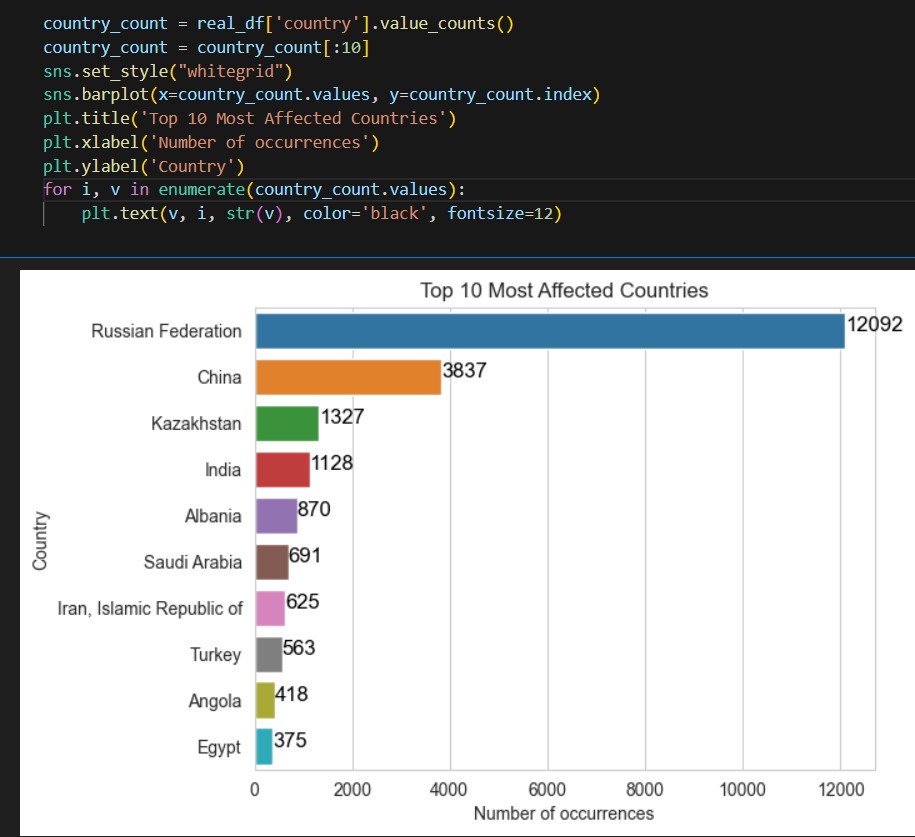
## Activity 2.2: Bivariate analysis

Utilizing the code provided in the accompanying visual representation, we can ascertain the top ten countries that experienced the highest impact from the disease. The code employs a specific methodology to analyze the dataset and extract the relevant information, enabling the identification of the countries that suffered the most significant effects of the disease outbreak.

Confirmation of lumpy skin disease in a new area requires virus isolation and identification. Antigen testing can be done using direct immunofluorescent staining, virus neutralization, or ELISA. Typical capripox (genus) virions can be seen using transmission electron microscopy of biopsy samples or desiccated crusts.

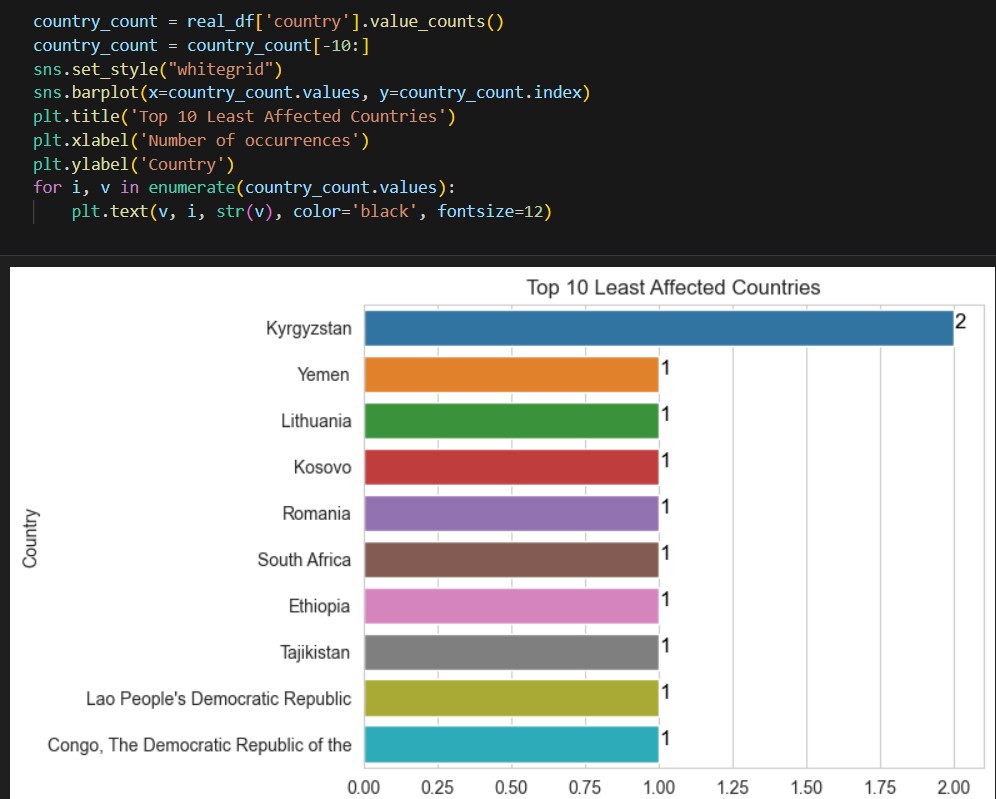
Collect appropriate samples based on the clinical signs. Skin lesions and scabs, nasal, oral and ocular swabs, EDTA blood and serum are preferred samples for laboratory testing. 3. All materials used for sampling skin tissue should either be autoclaved or be disposed off safely.

Serotype: Lumpy Skin Disease Virus (LSDV); Family: Poxviridae; Genus: Capripoxvirus (also Sheep Pox and Goat Pox). Can be recovered from skin nodules kept at −80°C for 10 years and infected tissue culture fluid stored at 4°C for 6 months. Survives for long periods at ambient temperature especially in dried scabs. Lumpy skin disease (LSD) is caused by lumpy skin disease virus (LSDV), a virus from the family Poxviridae, genus Capripoxvirus. Sheeppox virus and Goatpox virus are the two other virus species in this genus.



Similarly, employing the code depicted in the aforementioned visual representation, we can also determine the ten least affected countries. This code utilizes a specific approach to analyze the dataset and extract the pertinent information, enabling the identification of countries that experienced relatively lower impact from the disease outbreak. The incubation period for lumpy skin disease is between 4 and 14 days post-infection. After an initial period of high fever (41°C) and swollen lymph glands, the animal may develop large, firm nodules that are up to 5 cm in diameter in the skin. These can be found all over the body, but particularly on the head. examining the data, we can ascertain the countries that were least affected by the disease.

.



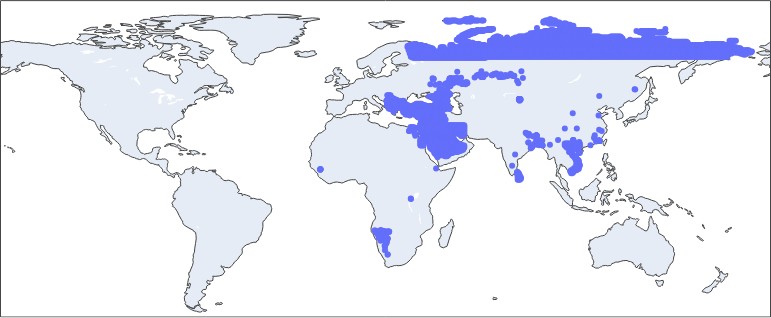
Dairy cattle in peak production are often the most severely affected with a marked decrease in milk production.

Depression, anorexia, rhinitis, conjunctivitis and excess salivation may also be observed.

In severely affected animals, necrotic lesions can also develop in the respiratory and gastrointestinal tract.

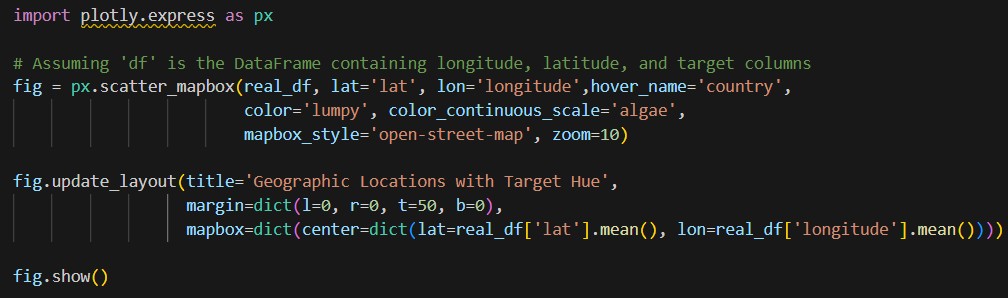
The disease can be subclinical (up to 50% of cases in an outbreak) or may be very severe or even fatal. Morbidity varies between 5 to 45% and mortality rate usually remains below 10% but both rates can be considerably higher when an outbreak occurs in a naïve cattle population.

This is the first study to compare the forecasting abilities of the FTS, NNAR, and ARIMA models across multiple phases of the LSD epidemic. Livestock authorities and decision-makers may incorporate the forecasting techniques demonstrated herein into the LSD surveillance system to enhance its functionality and utility.



## Activity 2.3: Multivariate analysis

Continuing our utilization of the Plotly module, we employ the Scatter mapbox functionality for multivariate analysis. By leveraging the Scatter mapbox function, we can visualize the distinction between locations that were affected by the disease and those that were not. This analysis allows us to observe and discern any discernible patterns, spatial relationships, or differences between diseased and non-diseased locations on a geographical map. The interactive nature of Plotly enables us to explore and gain deeper insights into the spatial dynamics of the disease's impact.



It causes fever, nodules on the skin and can also lead to death, especially in animals that that have not previously been exposed to the virus. Control options include vaccinations and culling of infected animals.

Lumpy skin disease can lead to significant economic losses.

The disease is present in many African countries. In 2012, it spread from the Middle East to south-east Europe, affecting EU Member States (Greece and Bulgaria) and several other countries in the Balkans. A vaccination programme has since halted the epidemic in south-east Europe.

The report says that the more effective the vaccination is in protecting animals against the disease – and the more herds are vaccinated – the shorter the vaccination programme can be. For example, if the vaccination is effective for 80% of vaccinated animals, a two-year programme with coverage of 90% of herds is sufficient.

The probability that LSD will reappear after a vaccination programme is mainly linked to the likelihood of infected animals being introduced from neighbouring affected areas. Other factors examined in the report include the possible persistence of the virus in vectors (such as ticks and insects) or in the environment.

The report also gives an overview of surveillance methods. These include measures for early detection of new cases and how to demonstrate absence of disease.

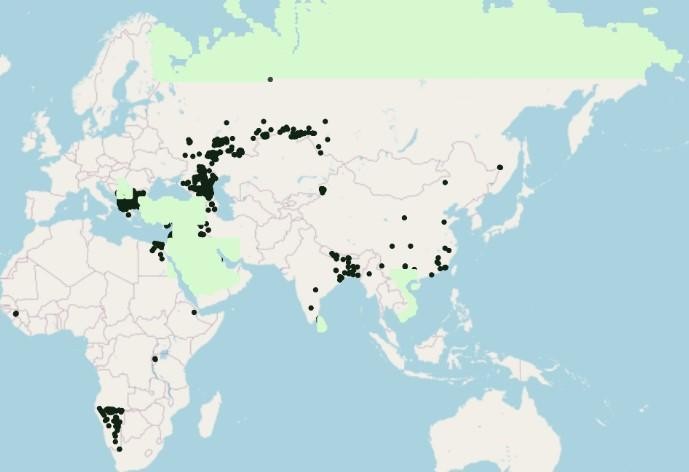
EFSA helps to ensure that food producing animals remain healthy, as part of its mandate to improve EU food safety and animal health and to ensure a high level of consumer protection.

EFSA provides scientific advice on transboundary animal diseases – highly contagious diseases that can spread rapidly across national borders.

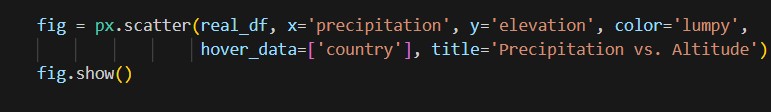
Experts of EFSA’s Animal Health and Welfare Panel assess the latest available knowledge on epidemiology , diagnosis and control of such disease.

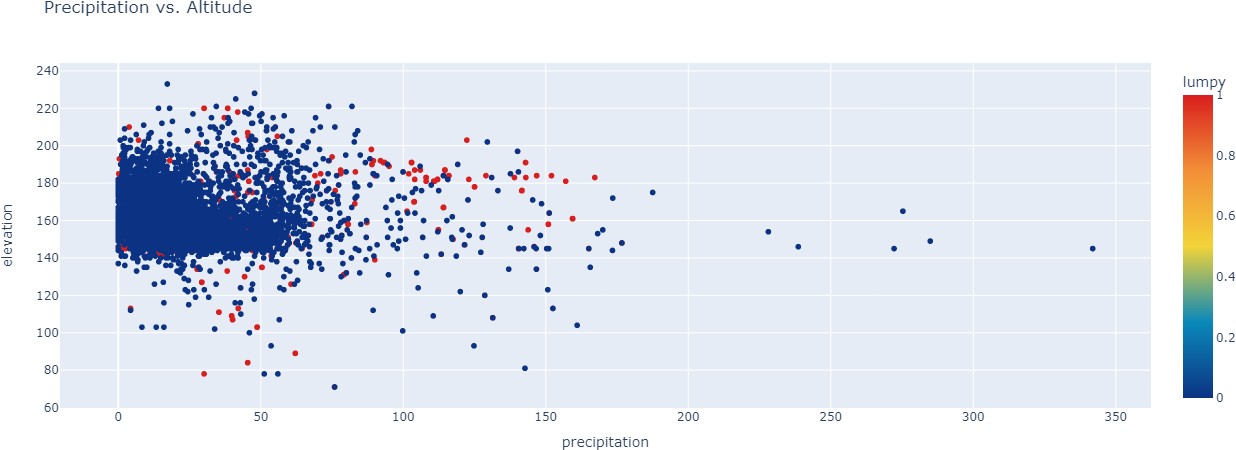
The epidemics of lumpy skin disease in EU has been controlled mainly thanks to the coordinated control measures, taken in the Balkan region, based above all on regional vaccination campaign. Since before the epidemics EFSA has been performing risk assessment on LSD to support the decision‐making process both for EC and the national authorities as well. Periodical meetings with representatives from affected and at‐risk countries have been organised by EFSA in order to commonly agree on type of data to collect and share by the national authorities to serve the risk assessment. Given the fact that some countries have already stopped vaccination and they want to prepare the so called ‘exit strategy’, wever, due to inadequate study design in the reviewed studies, there is no consensus on the magnitude of such effects and on their real consequences on production.



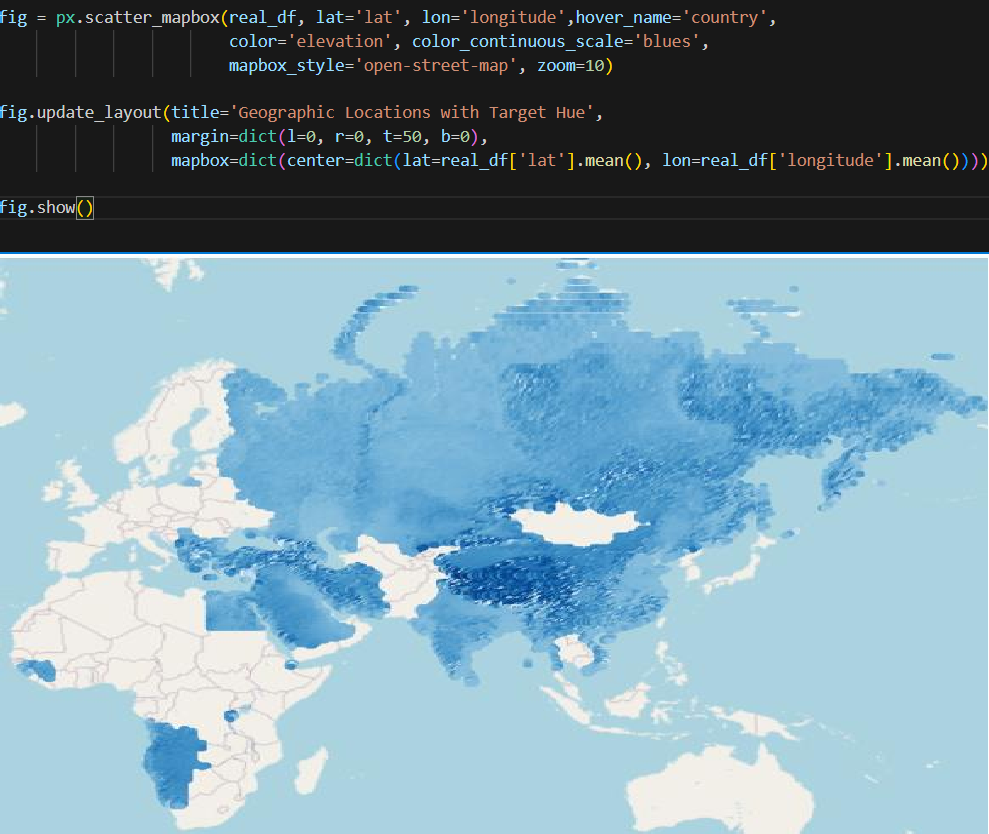


Once again, we harness the power of the Plotly module to explore the relationship between two numerical variables and a categorical column. By employing Plotly's visualization capabilities, we can create interactive charts or graphs that provide insights into the connections, dependencies, or patterns that may exist between these variables. This analysis enables us to better comprehend how the categorical column interacts with and influences the numerical variables, allowing for a more comprehensive understanding of the dataset's underlying dynamics.





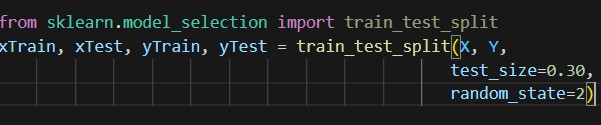
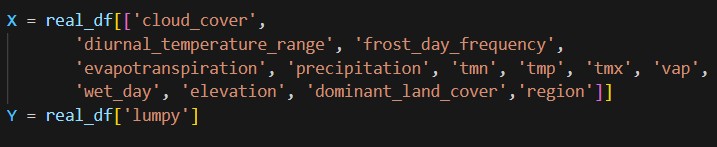
To ascertain the interplay between elevation and geographical coordinates, we employ the following code snippet as part of our analytical methodology.



# Milestone 4: Model Building

## Activity 1: Splitting data into train and test

In order to proceed with the training and evaluation of our model, it is essential to split the dataset into separate train and test sets. This process involves initially dividing the dataset into independent features, denoted as 'X', and the target variable, denoted as 'y'. Subsequently, we perform the actual data split, which partitions the dataset into distinct train and test subsets. This division allows us to utilize the independent features (X) to

predict and assess the accuracy of the target variable (y) in an unbiased manner.

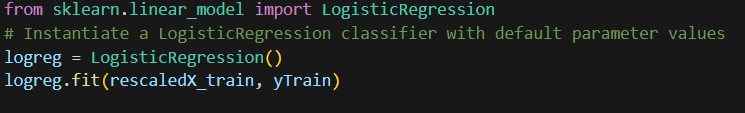
## Activity 2: Training the Model with Multiple Algorithms

With the dataset now cleaned and prepared, we proceed to construct our model. To ensure a comprehensive evaluation, we train our data using multiple algorithms. In this particular project, we have selected four classification algorithms to apply. By employing this ensemble of algorithms, we can leverage their unique strengths and characteristics, enabling us to obtain a more robust and accurate model.

During the training process, we carefully monitor the performance of each algorithm. Based on their respective performance metrics, we identify the best-performing model. This superior model is then saved, ensuring that we retain the optimal solution for subsequent use and further analysis.

# Activity 2.1: Logistic Regression Model.

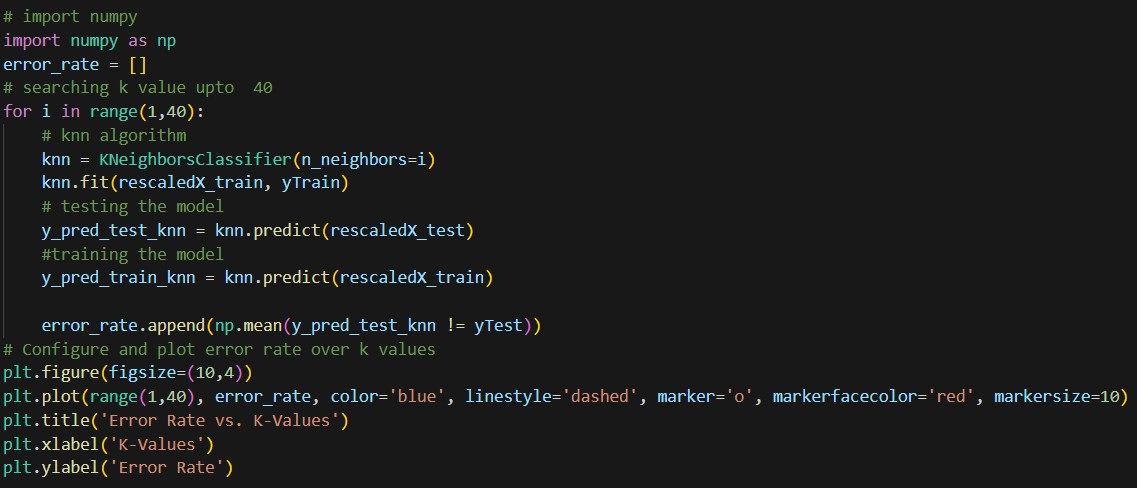
The provided code leverages the scikit-learn library to conduct logistic regression modeling. This process involves utilizing regression algorithms and techniques to analyze the relationship between variables and make predictions based on the linear relationship observed in the data. By employing the scikit-learn library, we can effectively apply the principles of logistic regression to model and interpret the data, aiding in making informed decisions and drawing meaningful insights**.**



## Activity 2.2: Decision Tree Classifier Model.

The presented code demonstrates the implementation of decision tree classification modelling using the scikit-learn library. It involves creating an instance of the DecisionTreeClassifier class, named "dtc", which serves as the decision tree classifier object. The subsequent step involves applying the "fit" method on the training data, denoted as X\_train and y\_train, in order to train the decision tree classifier model. This process allows the model to learn from the provided training data and build a decision tree-based classification model, enabling accurate predictions and classifications of unseen data based on learned patterns and rule.

Lumpy skin disease virus (LSDV) causes an infectious disease in cattle. Due to its direct relationship with the survival of arthropod vectors, geospatial and climatic features play a vital role in the epidemiology of the disease. The objective of this study was to assess the ability of some machine learning algorithms to forecast the occurrence of LSDV infection based on meteorological and geological attributes. Initially, ExtraTreesClassifier algorithm was used to select the important predictive features in forecasting the disease occurrence in unseen (test) data among meteorological, animal population density, dominant land cover, and elevation attributes. Some machine learning techniques revealed high accuracy in predicting the LSDV occurrence in test data (up to 97%). In terms of area under curve (AUC) and F1 performance metric scores, the artificial neural network (ANN) algorithm outperformed other machine learning methods in predicting the occurrence of LSDV infection in unseen data with the corresponding values of 0.97 and 0.94, respectively. Using this algorithm, the model consisted of all predictive features and the one which only included meteorological attributes as important features showed similar predictive performance. According to the findings of this research, ANN can be used to forecast the occurrence of LSDV infection with high precision using geospatial and meteorological parameters. Applying the forecasting power of these methods could be a great help in conducting screening and awareness programs, as well as taking preventive measures like vaccination in areas where the occurrence of LSDV infection is a high risk.

enabling accurate classification of unseen data points.

## Activity 1.1: XGBoost Model.

The code provided illustrates the implementation of XGBoost modeling using the scikit- learn library. It involves creating an instance of the XGClassifier class, denoted as "model," which serves as the XGBoost classifier object. The subsequent step entails invoking the "fit" method on the training data, X\_train and y\_train, in order to train the XGBoost model. By employing the "fit" method, the model learns from the provided training data, optimizing the boosting algorithm to make accurate predictions and classifications.

XGBoost is a powerful gradient boosting framework that enables effective modeling of complex relationships within the data, enhancing prediction accuracy and facilitating robust decision-making.



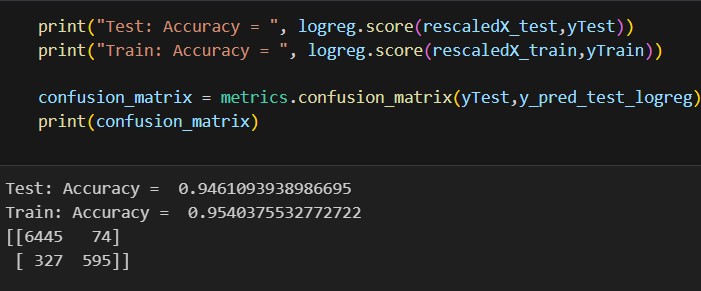
# Milestone 5: Performance Testing

## Activity 1: Evaluating Model Performance on Test and Train Data

To assess the performance of the models on both test and train data, we can employ the "score" method to calculate the discrepancy between the predicted and actual values. By comparing the accuracy scores obtained for the test and train data, we can gain insights into whether the model is exhibiting signs of overfitting or underfitting.

Analyzing the performance on both test and train data is crucial for understanding the model's generalization capabilities and ensuring it is effectively learning from the provided data.

## Activity 1.1: Logistic Regression Model.

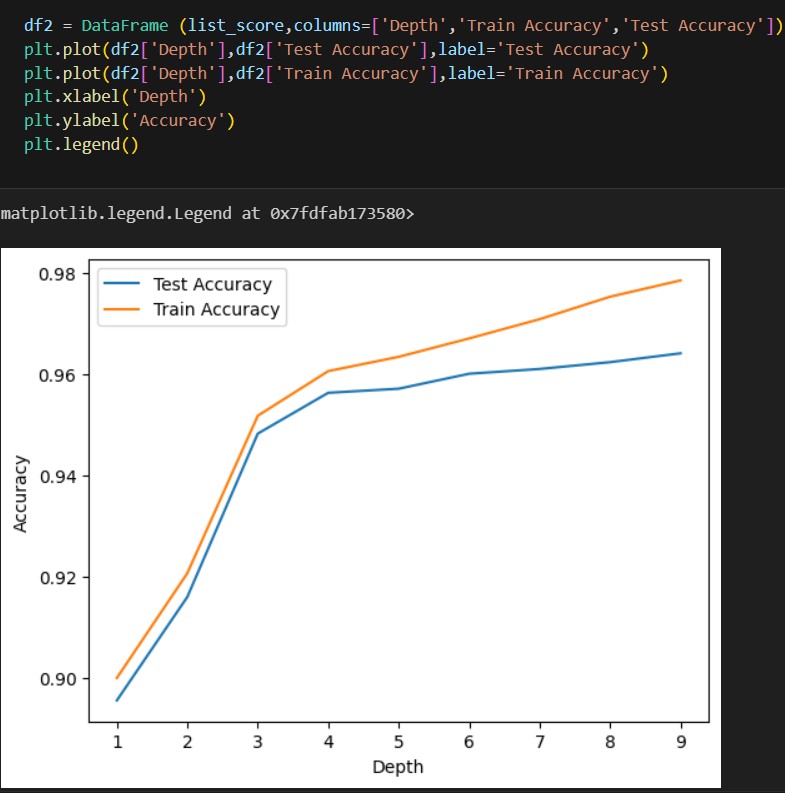


**Activity 1.2: Decision Tree Classifier Mode**

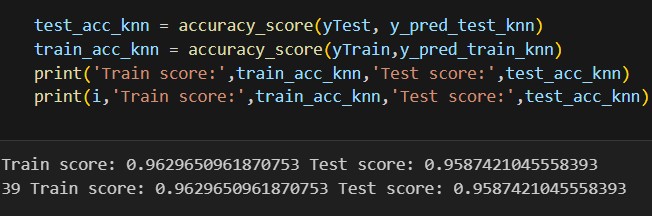
**A decision tree is one of the most powerful tools of supervised learning algorithms used for both classification and regression tasks. It builds a flowchart-like tree structure where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. It is constructed by recursively splitting the training data into subsets based on the values of the attributes until a stopping criterion is met, such as the maximum depth of the tree or the minimum number of samples required to split a node.**

**During training, the Decision Tree algorithm selects the best attribute to split the data based on a metric such as entropy or Gini impurity, which measures the level of impurity or randomness in the subsets. The goal is to find the attribute that maximizes the information gain or the reduction in impurity after the split.**

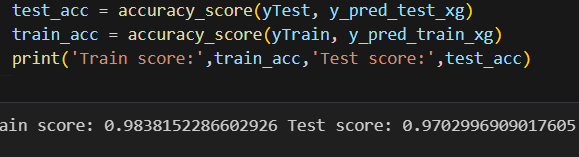
**A decision tree is a flowchart-like tree structure where each internal node denotes the feature, branches denote the rules and the leaf nodes denote the result of the algorithm. It is a versatile supervised machine-learning algorithm, which is used for both classification and regression problems. It is one of the very powerful algorithms. And it is also used in Random Forest to train on different subsets of training data, which makes random forest one of the most powerful algorithms in machine learning.**



## Activity 1.3: K Nearest Neighbours Classifier Model.



**Activity 1.4: XGBoost Model.**



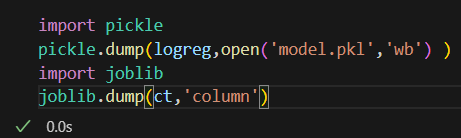
## Activity 2: Comparing models

The provided code snippet generates a Pandas DataFrame called "results," encompassing pertinent information such as the model names, test accuracy scores, and train accuracy scores for both the training and testing data. Specifically, this DataFrame encompasses the evaluation metrics for four regression models: Logistic Regression, Decision Tree Classifier, KNN Classifier, and XGBoost. By utilizing this code, we can systematically compare and analyze the performance of each model based on their respective accuracy scores. This allows for a comprehensive assessment of how well the models perform on both the training and testing datasets. The resulting DataFrame aids in visualizing and interpreting the effectiveness of the regression models, thereby supporting informed decision-making and model selection in complex business scenarios.

# Milestone 6: Model Deployment

## Activity 1: Save the best model

The provided code employs the Python "pickle" library to save the trained Logistic Regression model, named "lr," as a file with the name "model.pkl." The "dump" method from the pickle library is utilized to serialize the model object, allowing it to be stored and reused at a later stage. Notably, the "wb" parameter signifies that the file should be opened

in binary mode for writing data. By utilizing the pickle library and the "dump" method with the specified parameters, the trained Logistic Regression model is persistently stored as a serialized file. This facilitates the convenience of loading and utilizing the model in subsequent sessions, providing the capability for reuse and deployment in various applications without the need for retraining

.

## Activity 2: Integrate with Web Framework

In this section, we will be building a web application that would help us integrate the machine learning model we have built and trained.

A user interface is provided for the users to enter the values for predictions. The entered values are fed into the saved model, and the prediction is displayed on the UI.

The section has following task:

* Building HTML pages
* Building server side script
* Run the web application

# Activity 2.1: Building Html Pages:

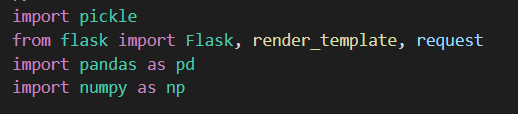
For this project we create three html files:

* first\_page.html
* form\_page.html
* predict\_page.html
* eda\_page.html

and save these html files in the templates folder

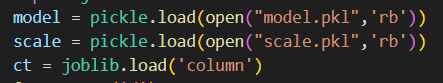
## Activity 2.2: Build Python code:

Importing the libraries



This code first loads the saved Logistic Regression model from the "model.pkl" file using the "pickle.load()" method. The "rb" parameter indicates that the file should be opened in binary mode to read data from it.

After loading the model, the code creates a new Flask web application object named "lumpyapp" using the Flask constructor. The "name" argument tells Flask to use the current module as the name for the application.

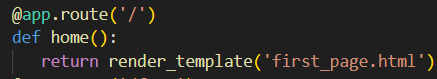


This code sets up a new route for the Flask web application using the "@app.route()" decorator. The route in this case is the root route "/", which is the default route when the website is accessed.

The function "home()" is then associated with this route. When a user accesses the root route of the website, this function is called.

The "render\_template()" method is used to render an HTML template named

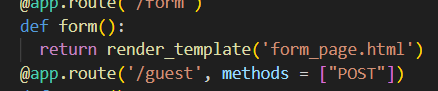
"first\_page.html". The “first\_page.html” is the home page.



The route in this case is "/form". When a user accesses the "/form" route of the website,

this function is “form” called.

The "render\_template()" method is used to render an HTML template named "form.html".

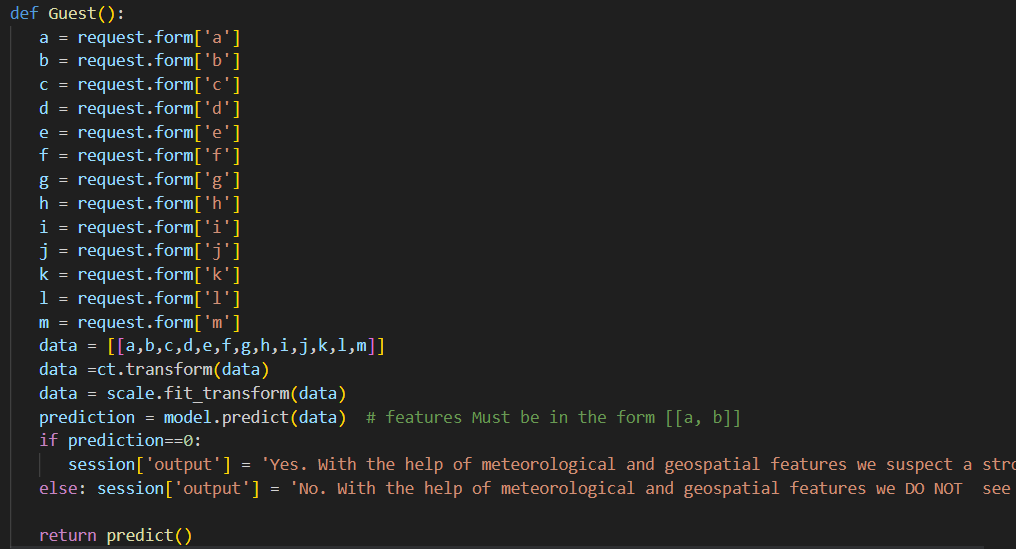


This code sets up another route for the Flask web application using the "@app.route()" decorator. The route in this case is "/guest", and the method is set to GET and POST.

The function "form()" is then associated with this route. This function first loads the previously saved Linear Regression model using "model = pickle.load(open('model.pkl', 'rb'))".

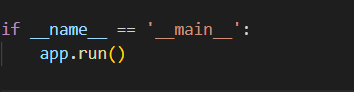
Then, the function receives the user inputs for various geographical variables using "request.form['...']". The function then uses the loaded Logistic Regression model to predict the disease based on the user inputs.

Finally, the predicted body fat percentage is passed to an HTML template, where it is displayed to the user.



Main Function:

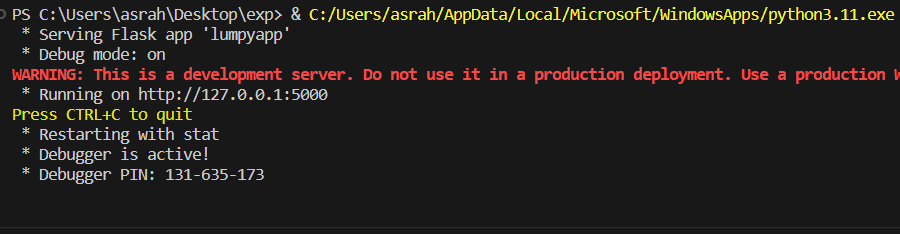
This code sets the entry point of the Flask application. The function "app.run()" is called, which starts the Flask development server.



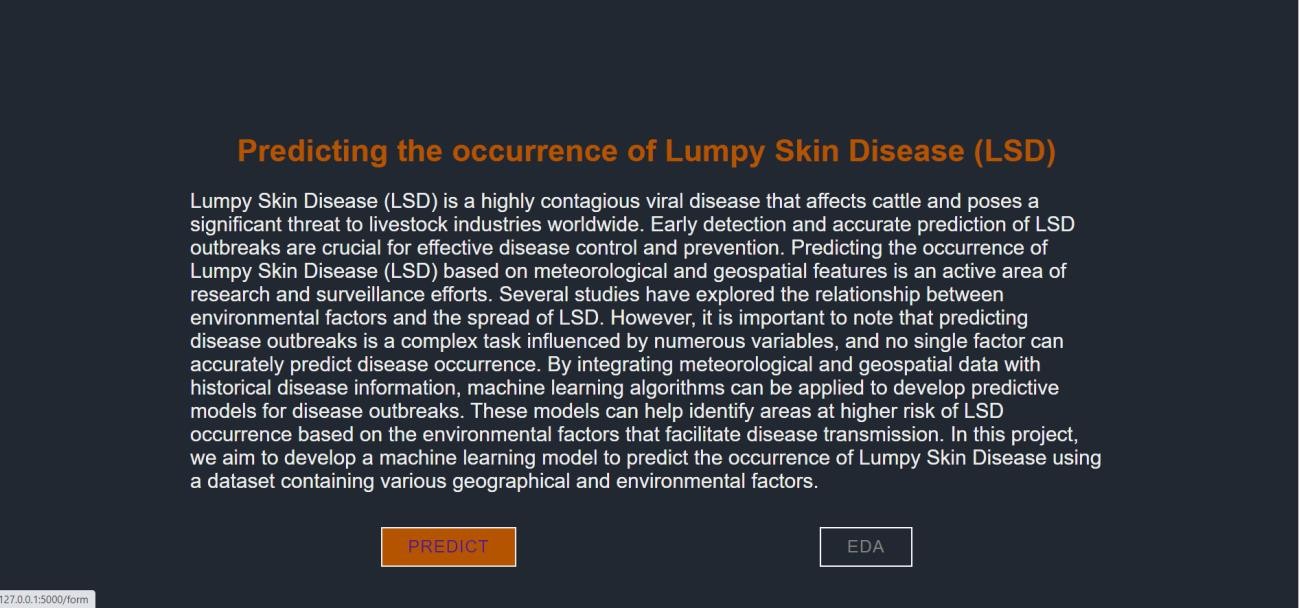
## Activity 2.3: Run the web application

When you run the “main.py” file this window will open in the output terminal. Copy the

**http://127.0.0.1:5000** and paste this link in your browser.

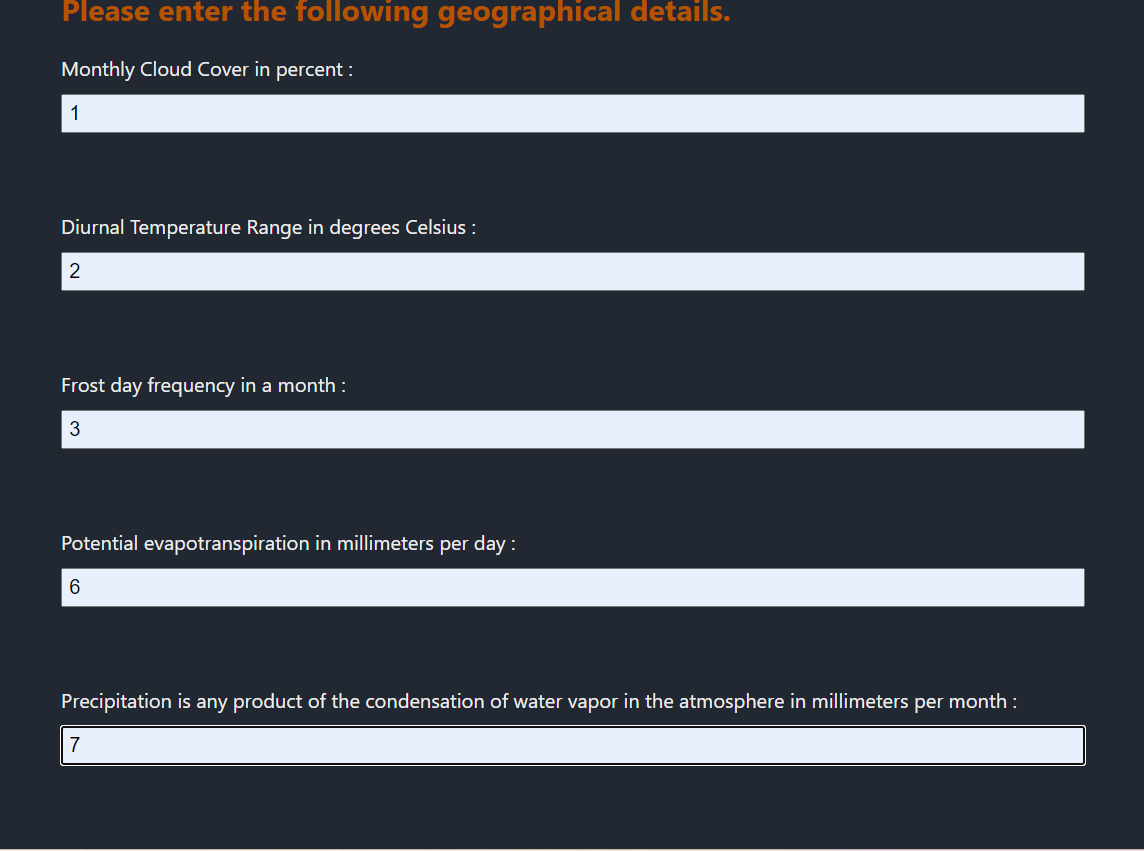
This is the "first\_page.html" file that appears when we paste the URL into the browser. To

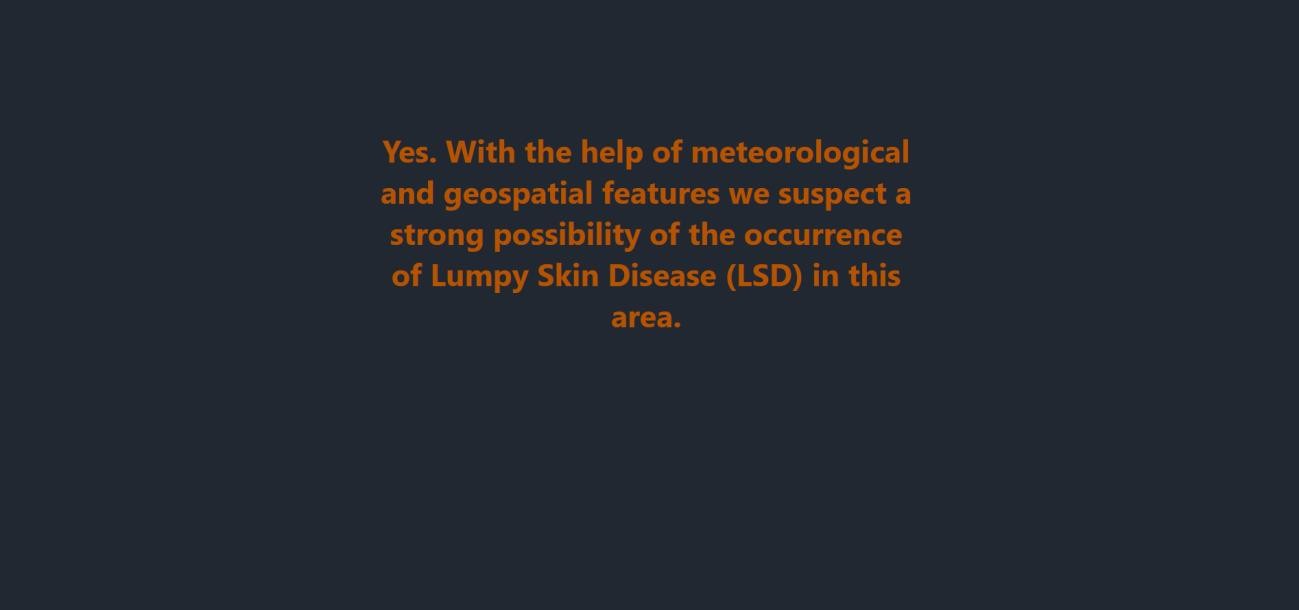
proceed to the next page, click the ‘Predict’ button.



On this page a user will input the following values and then click on “Predict” button to see

the disease prediction.





# Milestone 7: Project Demonstration & Documentation

Below mentioned deliverables to be submitted along with other deliverables

## Activity 1: Project Documentation-Step by step project development procedure.