**Classification Model**

**to determine the presence of mosquitoes**

**Wnv virus in Chicago, USA**

**depending on weather conditions**

**Explanatory note**

**Performed by Oleh Ivanyk**

1. **Information from the Internet on the topic of research.**
   1. Two species of Culex mosquitoes are common throughout much of North America. Culex restuans Theobold is a native species, while Culex pipiens L. is an immigrant from Europe who has lived in North America since the 1600s. “

"Larvae of the East. restuans are numerically dominant in spring and early summer, but Cx. pipiens dominates until mid-summer"

"Cx. pipiens is more likely to transmit West Nile virus to humans. "

"We studied this 31-year continuous record of adult populations

-for the presence of signs of crossing species,

-relationships between the abundance of both species and climatic factors.

- and signs of interspecific competition."

"Pearson's correlations showed that the abundance of both species was related to temperature and precipitation, but Cx. pipiens tended to be positively associated with climatic factors, while Cx. restuans showed a negative correlation."

Source: <https://pubmed.ncbi.nlm.nih.gov/26314047/#:~:text=Culex%20restuans%20Theobold%20is%20a,pipiens%20dominates%20by%20mid%2Dsummer>

* 1. "Cx. restuans, which are important vectors of West Nile virus and are more common than Cx. pipiens in some rural areas in the eastern United States at some times during the transmission season, are probably the most susceptible species studied to population fluctuations due to rising temperatures. In line with these findings, previous studies have shown that Cx. restuans tend to peak in late spring and early summer, followed by decline in the hotter summer months."

Source: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3955846/>

1. **The purpose of the simulation is to develop a forecast program based on a classification model that will assign sets with meteorological data to one of two classes. Class 1 – corresponds to the presence of virus infection in mosquitoes (WnvPresent = 1). Class 2 – corresponds to the absence of virus infection in mosquitoes (WnvPresent = 0).** 
   1. **The train data frame.csv**
      1. According to the modeling goal, we select the following data from this frame:

- Date – The date when the mosquitoes were removed from the trap

- Latitude – the latitude coordinate of the mosquito trap

- Longitude – longitude coordinate of the mosquito trap

- NumMosquitos – the number of mosquitoes caught in the trap

- WnvPresent: = 1 : if trapped mosquitoes are infected with a virus,

= 0 : if trapped mosquitoes are infected with the virus

* + 1. **Types of train data.csv**

|  |  |
| --- | --- |
| **Date** | object |
| **Latitude** | float64 |
| **Longitude** | float64 |
| **NumMosquitos** | int64 |
| **WnvPresent** | int64 |

* 1. **Weather data frame.csv**
     1. Weather Data Types.csv (cls.weather\_types.csv file)

|  |  |
| --- | --- |
| **Station** | int64 |
| **Date** | object |
| **Tmax** | int64 |
| **Tmin** | int64 |
| **Tavg** | object |
| **Depart** | object |
| **DewPoint** | int64 |
| **WetBulb** | object |
| **Heat** | object |
| **Cool** | object |
| **Sunrise** | object |
| **Sunset** | object |
| **CodeSum** | object |
| **Depth** | object |
| **Water1** | object |
| **SnowFall** | object |
| **PrecipTotal** | object |
| **StnPressure** | object |
| **SeaLevel** | object |
| **ResultSpeed** | float64 |
| **ResultDir** | int64 |
| **AvgSpeed** | object |

* + 1. The object data type means that it contains data of type 'str' or data of different types.
    2. Searched and replaced data that do not belong to the 'int', 'float' types in weather.csv (except Date and Station).
       1. Non-numeric values of Tavg are replaced by (Tmax + Tmin)/2;
       2. Non-numeric values of WetBulb are replaced by DewPoint + the average value of the difference between WetBulb and DewPoint (wb\_dpt\_avg);
       3. If a non-numeric data of type str contains a number, then convert this data to a number of type float. Otherwise, set it to np.nan.
       4. After this conversion, we have the following number of nan values for each given (file cls\_data\_nulsum.xlsx):

|  |  |
| --- | --- |
| **Tmax** | 0 |
| **Tmin** | 0 |
| **Tavg** | 0 |
| **Depart** | 1472 |
| **DewPoint** | 0 |
| **WetBulb** | 0 |
| **Heat** | 11 |
| **Cool** | 11 |
| **Sunrise** | 1472 |
| **Sunset** | 1472 |
| **CodeSum** | 2944 |
| **Depth** | 1472 |
| **Water1** | 2944 |
| **SnowFall** | 1484 |
| **PrecipTotal** | 320 |
| **StnPressure** | 4 |
| **SeaLevel** | 9 |
| **ResultSpeed** | 0 |
| **ResultDir** | 0 |
| **AvgSpeed** | 3 |

* + - 1. We remove from consideration the data Depart, Sunrise, Sunset, CodeSum, Depth, Water1, SnowFall, which have more than half of the undefined values.
      2. Replace the nan values of the parameters Heat, Cool, PrecipTotal, StnPressure, SeaLevel, AvgSpeed with the average values of these columns
      3. Change the type of weather data to float
    1. Weather data types.csv after all conversions (cls.weather \_types\_end.csv file)

|  |  |
| --- | --- |
| **Station** | int64 |
| **Date** | object |
| **Tmax** | float64 |
| **Tmin** | float64 |
| **Tavg** | float64 |
| **DewPoint** | float64 |
| **WetBulb** | float64 |
| **Heat** | float64 |
| **Cool** | float64 |
| **PrecipTotal** | float64 |
| **StnPressure** | float64 |
| **SeaLevel** | float64 |
| **ResultSpeed** | float64 |
| **ResultDir** | float64 |
| **AvgSpeed** | float64 |

* + 1. From the weather.csv the following data were selected for modeling:

- Date of meteorological data reading.

- Station is a weather station. There are two weather stations – Station1 and Station2.

- Tmax, Tmin, Tavg are respectively the maximum, minimum, and average temperatures in Fahrenheit F.

- DewPoint is the value of the dew point.

- WetBulb – wet bulb temperature.

- Heat – = the difference between the average temperature Tavg and 65 degrees F if Tavg > 65 and = 0 otherwise.

- Cool – = the difference between 65 degrees F and the average temperature Tavg if Tavg < 65 and = 0 otherwise.

- PrecipTotal - total rainfall in water equivalent

- StnPressure – atmospheric pressure.

- ResultSpeed is the resulting wind speed.

- AvgSpeed is the average wind speed.

- ResultDir is the resulting wind direction in degrees.

- SeaLevel – sea level

* 1. **The file is StationsLocation.txt.**
     1. From this file, we select the following data:

Station1 Latitude is the latitude coordinate of Station1.

Station1 Longitude is the longitude coordinate of the weather station Station1.

Station2 Latitude is the latitude coordinate of the weather station Station2.

Station2 Longitude is the longitude coordinate of Station2.

* 1. **The data preparation program is cls\_weather\_prepare.py. The result of preparing weather.csv and train.csv data is saved in cls\_weather\_prepare.csv and cls\_train\_prepare.csv files.**

1. **Processing of initial data.**
   1. **Inputs: csv\_weather\_prepare.csv, and csv\_train\_prepare.csv.**
   2. **Algorithm.**
      1. In the train frame, each line corresponds to a specific date and coordinates of the trap. In the weather frame, weather data for each date is set for two weather stations, Station1 and Station2. You need to select one weather data from this frame (Station1 or Station2) for each row of the train frame.
      2. For each line of the train frame, you need to find the nearest weather station according to the coordinates specified there. Weather data is selected for the nearest station by date from the train frame line
   3. **The processing program is csl\_data.py.**
      1. The distance(adr) function is used to calculate the distance between the trap and weather stations. 'ADR' – coordinates from the train frame string. The coordinates of Station1 Station2 stations are taken from the DSprojekt1/StationsLocation.txt file.
      2. 4.3.1. The distance(adr) function selects the nearest weather station and returns its index (1 or 2).
      3. Copy the data from the csv\_train\_prepare.csv to the working frame data\_pr. Add columns with weather data and expanded date to it: 'Tmax', 'Tmin', 'Tavg', 'DewPoint', 'WetBulb', 'Heat', 'Cool', 'PrecipTotal', 'SeaLevel', 'StnPressure', 'ResultSpeed', 'ResultDir', 'AvgSpeed'.
      4. In the loop, we select data\_pr nearest weather station from the weather frame (csv\_weather\_prepare.csv) for each line. According to the index of the selected weather station and the date from the data\_pr line, copy the weather data from the weather to the data\_pr frame.
      5. The frame data\_pr saved in the file cls\_dubl\_data.csv The number of rows of data in the frame is 8452.
      6. Remove all duplicate lines except one with the same values of the parameters 'WnvPresent', 'Tmax', 'Tmin', 'Tavg', 'DewPoint', 'WetBulb', 'Heat', 'Cool', 'PrecipTotal', 'StnPressure', 'ResultSpeed', 'ResultDir', 'AvgSpeed', 'SeaLevel'.
      7. Data without duplicate strings is saved in cls\_data.csv. The number of data lines is 205.
2. **Selection of independent parameters**
   1. **Input: cls\_dubl\_data.csv, cls\_data.csv**
   2. **Algorithm.**
      1. A measure of the dependence (relationship) of two random variables is the correlation coefficient between these variables. We calculate the Pearson correlation coefficient to estimate the degree of linear dependence between the meteorological data and the geocoordinates of the trap. We leave those parameters, the correlation coefficients between which do not exceed 0.8.
   3. **Program - cls\_research\_cor.py**
      1. The results of the calculation of the correlation coefficients between the meteorological data and the geocoordinates of the traps for the data from the cls\_dubl\_data.csv file (data with duplicate rows) are in Table 1 (file cls\_dubl\_corr.xlsx)
      2. The results of the calculation of the correlation coefficients between the meteorological data and the geocoordinates of the traps for the data from the cls\_ data.csv file (data without duplicate rows) are in Table 2 (file cls\_corr.xlsx)
      3. Remove the parameters 'Tmin', 'Tavg', 'WetBulb', 'Cool', 'SeaLevel', 'AvgSpeed', which have correlation coefficients > 0.8 with other parameters.
      4. **The data frame for modeling is saved: for data with duplicate rows - in cls\_wnv\_dubl\_mod.csv, for data without duplicate rows - in cls\_wnv\_mod.csv.**
3. **Vibio classification models.**
   1. **Choose the classification method that will have the highest accuracy.**
   2. **Input: cls\_wnv\_mod.csv, cls\_wnv\_dubl\_mod.csv**
   3. **Models using different classification methods were created for both data with duplicate strings (cls\_wnv\_dubl\_mod.csv) and for data without duplicate rows (cls\_wnv\_mod.csv)**
   4. **Modeling Methods**

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Simulation Method Name** | **Short name** |
| **Linear** | **Logistic regression**  **LogisticRegression** | **LR** |
| **Nonlinear** | **k-nearest neighbor method (classification)**  **K-Neighbors Classifier** | **KNN** |
| **Nonlinear** | **Decision Trees**  **Decision Tree Classifier** | **CART** |
| **Nonlinear** | **Naïve Bayesian classifier**  **Naive Bayes Classifier** | **NBC** |
| **Nonlinear** | **Support Vector Method**  **C-Support Vector Classification** | **SVC** |
| **Ensemble Algorithm** | **Random Forest**  **RandomForestClassifier** | **RF** |

* 1. **Simulation without prior data preparation**
  2. **The program cls\_model.py for data without duplicate rows, cls\_dubl\_model.py for data with duplicate rows**
     1. **Educational and training samples were formed by two methods:**

**- using the cross-method of forming KFold training and test samples;**

**- dividing the data into training (80% of the data) and training (20% of the data) samples**

* + 1. **In the K-FOLD crossover method, the sample is broken down into num\_folds folds, of which (num\_folds-1) folds are used for training, and the last one is used for testingThe resulting model is tested on the remaining data.  
       This process is repeated num\_folds times, and a new model is created at each stage.**
    2. **To evaluate the created models, the 'accuracy' indicator was used – the proportion of correctly classified data sets**

**-**

* 1. **Results obtained.**
     1. **Models based on data with duplicate strings (MDR symbol) – cls\_dubl\_model.py program, cls\_dubl\_res.xlsx file**

|  |  |  |  |
| --- | --- | --- | --- |
| **Simulation Method Name** | **Accuracy**  **‘accuracy’** | **Kfold Accuracy** | **Standard deviation** |
| **LR** | **0,94** | **0,94** | **0,04** |
| **KNN** | **0,95** | **0,93** | **0,03** |
| **CART** | **0,93** | **0,92** | **0,02** |
| **NBC** | **0,94** | **0,88** | **0,10** |
| **SVC** | **0,94** | **0,94** | **0,04** |
| **RF** | **0,94** | **0,94** | **0,04** |

* + 1. **Models on data without duplicate rows. (M) program cls\_ model.py, file cls\_ res.xlsx**

|  |  |  |  |
| --- | --- | --- | --- |
| **Simulation Method Name** | **Accuracy '** | **Kfold Accuracy** | **Average**  **Square deviation** |
| **LR** | **0,76** | **0,69** | **0,10** |
| **KNN** | **0,66** | **0,68** | **0,09** |
| **CART** | **0,66** | **0,69** | **0,07** |
| **NBC** | **0,71** | **0,56** | **0,07** |
| **SVC** | **0,76** | **0,69** | **0,09** |
| **RF** | **0,73** | **0,69** | **0,09** |

* 1. **Simulation with preconditioning**
  2. **Programs cls\_pred\_model.py, cls\_pred\_dubl\_model.py**
     1. **The program implements preliminary data preparation by standardization.**
     2. **Standardization of output data is carried out by the StandardScaler function.**
     3. **To automate the operations of preliminary standardization of data and assessments, we use Pipeline**
     4. **The formation of educational and training samples was carried out in the same way as in clause 5.6**
  3. **Results obtained.**
     1. **Models based on standardized data with duplicate strings (MSDR symbol) – cls\_dubl\_pred\_model.py program,**

**cls\_pred\_dubl\_res.xlsx' file**

|  |  |  |  |
| --- | --- | --- | --- |
| **Simulation Method Name** | **Accuracy** | **KFold Accuracy** | **Standard deviation** |
| LR | 0,94 | 0,94 | 0,04 |
| KNN | 0,95 | 0,93 | 0,03 |
| CART | 0,93 | 0,92 | 0,03 |
| NBC | 0,94 | 0,88 | 0,10 |
| SVC | 0,94 | 0,94 | 0,04 |
| RF | 0,94 | 0,94 | 0,04 |

* + 1. **Models based on standardized data without duplicate strings (MC symbol) program cls\_pred\_model.py, file cls\_pred\_res.xlsx**

|  |  |  |  |
| --- | --- | --- | --- |
| **Simulation Method Name** | **Accuracy** | **KFold Accuracy** | **Standard deviation** |
| LR | 0,80 | 0,73 | 0,09 |
| KNN | 0,80 | 0,70 | 0,09 |
| CART | 0,59 | 0,69 | 0,07 |
| NBC | 0,71 | 0,56 | 0,07 |
| SVC | 0,83 | 0,72 | 0,10 |
| RF | 0,78 | 0,70 | 0,09 |

* 1. **Analysis of the results obtained.**
     1. **The value of the RMS accuracy error for MDR models is 0.02 – 0.1, which does not exceed 2 – 11% of the average accuracy value.**

**The value of the RMS accuracy error for M models is 0.07 – 0.1, which does not exceed 12 – 14% of the average accuracy value.**

**The value of the RMS accuracy error for MSBC models is 0.03 – 0.1, which does not exceed 3 – 11% of the average accuracy value.**

**The value of the RMS accuracy error for MC models is 0.07 – 0.1, which does not exceed 1 – 14% of the average accuracy value.**

**All these results indicate that the 'accuracy' estimates of the models are quite accurate.**

* + 1. **For MDR models, MSDR (data with duplicate strings), the highest 'accuracy' is achieved by LR, KNN, SVC methods.**
    2. **For M models (data without duplicate strings), the LR and SVC methods have the highest 'accuracy'.**
    3. **For MC models (data without duplicate strings), the highest 'accuracy' is achieved by LR, KNN, SVC methods.**
    4. **The accuracy of models using the KFold cross-sampling method is lower than that of models without it.**
    5. **The accuracy of models based on data with duplicate MDR lines, MSDR is higher than that of M MS models on data without duplicate lines.**
       1. **The number of rows of datasets with duplicate MDR rows, MSDR is 8452. The number of rows of M MS datasets without duplicate rows is 205. This indicates that in the sets of MDRs, MSDRs there are a very large number of rows with completely identical data on the parameters and the qualifying feature of WnvPresent. This results in a large number of rows with the same data in both the training and test datasets in the simulation. And that's why it gives such high marks. Training and testing models on nearly identical datasets gives unreliable estimates of their accuracy.**
       2. **Therefore, we prefer models based on data without duplicate strings.**
       3. **Finally, we choose the SVC classification model (C-Support Vector Classification) with data standardization (Accuracy 'accuracy' = 0.83)**
  1. **Prediction of mosquito infestation based on meteorological data using classification model**
     1. **Program cls\_prognoz.py**
     2. **Input: Ecel data.xlsx table with data:**

**Tmax, DewPoint, Heat, PrecipTotal, StnPressure, ResultSpeed, ResultDir, Latitude, Longitude.**

**Table placement: D:\Data\data.xlsx**

* + 1. **The result of the program is displayed in the form of an Ecel table**

**D:\Data\data.xlsx**

* + 1. **The test results are shown in Tables 3 and 4. According to the results of testing accurate forecasts, 5 out of 5.**
  1. **Prediction of mosquito infestation based on meteorological data using a neural network**
     1. **Program cls\_neural.py**
     2. **Input: Ecel data.xlsx table with data:**

**Tmax, DewPoint, Heat, PrecipTotal, StnPressure, ResultSpeed, ResultDir, Latitude, Longitude.**

**Table placement: D:\Data\data.xlsx**

* + 1. **The classification model is based on a neural network with a multilayer perceptron. We use the MLPClasifier class, which implements a multilayer perceptron (MLP) algorithm that is trained using backpropagation.**
    2. **The result of the program is displayed in the form of an Ecel table**

**D:\Data\cls\_res\_neuro.xlsx**

* + 1. **Testing was carried out according to the same data as in paragraph 5.12.4. The test result is shown in Table 5. According to the results of testing accurate forecasts, 3 out of 5.**

**Table 1. Correlation coefficients between parameters for data with duplicate rows (8452 rows of data)**

****

**Table 2 Correlation coefficients between parameters. Data without duplicate rows (205 rows)**

****

**Table 3. A set of inputs for forecast testing.**

****

**Table 4. Forecast results using a classification model. The prediction is correct: 5 out of 5 correct answers**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tmax** | **DewPoint** | **Heat** | **PrecipTotal** | **StnPressure** | **ResultSpeed** | **ResultDir** | **Latitude** | **Longitude** | **Is the mosquito infected with the virus?** |
| **63** | **47** | **8** | **0,27** | **29,16** | **6,2** | **3** | **41,9216** | **-87,666455** | **No** |
| **85** | **69** | **0** | **0,92** | **29,18** | **10,3** | **24** | **41,686398** | **-87,531635** | **Yes** |
| **83** | **67** | **0** | **0,04** | **29,16** | **1,2** | **36** | **41,9216** | **-87,666455** | **No** |
| **92** | **62** | **0** | **0** | **29,29** | **3,5** | **9** | **41,95469** | **-87,800991** | **No** |
| **91** | **63** | **0** | **0** | **29,34** | **2,1** | **13** | **41,688324** | **-87,676709** | **Yes** |
|  |  |  |  |  |  |  |  |  |  |

**Table 5. Forecast results using a neural network. 3 out of 5 correct answers**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tmax** | **DewPoint** | **Heat** | **PrecipTotal** | **StnPressure** | **ResultSpeed** | **ResultDir** | **Latitude** | **Longitude** | **Is the mosquito infected with the virus?** |
| 63 | 47 | 8 | 0,27 | 29,16 | 6,2 | 3 | 41,9216 | -87,666455 | No |
| 85 | 69 | 0 | 0,92 | 29,18 | 10,3 | 24 | 41,686398 | -87,531635 | No |
| 83 | 67 | 0 | 0,04 | 29,16 | 1,2 | 36 | 41,9216 | -87,666455 | Yes |
| 92 | 62 | 0 | 0 | 29,29 | 3,5 | 9 | 41,95469 | -87,800991 | No |
| 91 | 63 | 0 | 0 | 29,34 | 2,1 | 13 | 41,688324 | -87,676709 | Yes |