Name: COURTIOL Pierre

Location: Paris

Email: [pierre.courtiol@free.fr](mailto:pierre.courtiol@free.fr)

Competition: West Nile Virus Prediction

Index

[1. Summary 2](#_Toc423710291)

[2. Features extraction 2](#_Toc423710292)

[a. Basic features 2](#_Toc423710293)

[b. Additional features for all traps 2](#_Toc423710294)

[c. Additional features for the 60 closest traps 3](#_Toc423710295)

[3. Modeling techniques and Training 4](#_Toc423710296)

[4. Leaderboard feedback 4](#_Toc423710297)

[a. Date multipliers 4](#_Toc423710298)

[b. Species multipliers 4](#_Toc423710299)

[5. File description 5](#_Toc423710300)

[a. train.py and predict.py scripts 5](#_Toc423710301)

[b. preprocess.py script 5](#_Toc423710302)

[c. train.m and predict.m scripts 5](#_Toc423710303)

[d. adjustProba.m 5](#_Toc423710304)

[6. Dependencies 5](#_Toc423710305)

[a. Octave dependencies 5](#_Toc423710306)

[b. Python dependencies 5](#_Toc423710307)

[7. How to generate the solution 6](#_Toc423710308)

[a. Training step 6](#_Toc423710309)

[b. Prediction step 6](#_Toc423710310)

[c. Set the path 6](#_Toc423710311)

# Summary

The general approach is mainly based on the duplicate rows in the dataset, from which the approximate number of mosquitos can be inferred. The number of mosquitos is a very useful piece of information to predict if West Nile virus is present or not.

# Features extraction

## Basic features

The following simple features were extracted from the dataset:

F1 => day of year

F2 => longitude

F3 => latitude

F4 => week of year

F5 => count duplicates

F6:13 => one-hot encoding for species

## Additional features for all traps

For all traps:

F14 => number of rows found twice from the beginning of the year to the current date

F15 => number of rows found three times from the beginning of the year to the current date

F16 => number of rows found four times from the beginning of the year to the current date

F17 => number of rows found five times from the beginning of the year to the current date

F18 => number of rows found six times rows from the beginning of the year to the current date

F19 => sum(F14:F18)

F20 => number of rows found at least twice from the beginning of the year to the current date

F21 => number of rows found at least twice for the current date

An example might be of some help here:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Basic features | | | | | | | | | Additional features | | | | | | | |
| F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | … | F14 | F15 | F16 | F17 | F18 | F19 | F20 | F21 |
| 149 | -87 | 41 | 22 | 1 | 0 | 0 | 1 |  | 2 | 0 | 1 | 0 | 0 | 3 | 8 | 8 |
| 149 | -87 | 41 | 22 | 1 | 0 | 1 | 0 |  | 2 | 0 | 1 | 0 | 0 | 3 | 8 | 8 |
| 149 | -87 | 40 | 22 | 2 | 0 | 0 | 1 |  | 2 | 0 | 1 | 0 | 0 | 3 | 8 | 8 |
| 149 | -87 | 40 | 22 | 2 | 0 | 0 | 1 |  | 2 | 0 | 1 | 0 | 0 | 3 | 8 | 8 |
| 149 | -87 | 39 | 22 | 4 | 1 | 0 | 0 |  | 2 | 0 | 1 | 0 | 0 | 3 | 8 | 8 |
| 149 | -87 | 39 | 22 | 4 | 1 | 0 | 0 |  | 2 | 0 | 1 | 0 | 0 | 3 | 8 | 8 |
| 149 | -87 | 39 | 22 | 4 | 1 | 0 | 0 |  | 2 | 0 | 1 | 0 | 0 | 3 | 8 | 8 |
| 149 | -87 | 39 | 22 | 4 | 1 | 0 | 0 |  | 2 | 0 | 1 | 0 | 0 | 3 | 8 | 8 |
| 149 | -87 | 39 | 22 | 1 | 0 | 1 | 0 |  | 2 | 0 | 1 | 0 | 0 | 3 | 8 | 8 |
| 149 | -87 | 38 | 22 | 2 | 1 | 0 | 0 |  | 2 | 0 | 1 | 0 | 0 | 3 | 8 | 8 |
| 149 | -87 | 38 | 22 | 2 | 1 | 0 | 0 |  | 2 | 0 | 1 | 0 | 0 | 3 | 8 | 8 |
| 156 | -87 | 38 | 23 | 2 | 1 | 0 | 0 |  | 4 | 0 | 1 | 1 | 0 | 6 | 17 | 9 |
| 156 | -87 | 38 | 23 | 2 | 1 | 0 | 0 |  | 4 | 0 | 1 | 1 | 0 | 6 | 17 | 9 |
| 156 | -87 | 38 | 23 | 1 | 0 | 0 | 1 |  | 4 | 0 | 1 | 1 | 0 | 6 | 17 | 9 |
| 156 | -87 | 40 | 23 | 5 | 0 | 1 | 0 |  | 4 | 0 | 1 | 1 | 0 | 6 | 17 | 9 |
| 156 | -87 | 40 | 23 | 5 | 0 | 1 | 0 |  | 4 | 0 | 1 | 1 | 0 | 6 | 17 | 9 |
| 156 | -87 | 40 | 23 | 5 | 0 | 1 | 0 |  | 4 | 0 | 1 | 1 | 0 | 6 | 17 | 9 |
| 156 | -87 | 40 | 23 | 5 | 0 | 1 | 0 |  | 4 | 0 | 1 | 1 | 0 | 6 | 17 | 9 |
| 156 | -87 | 40 | 23 | 5 | 0 | 1 | 0 |  | 4 | 0 | 1 | 1 | 0 | 6 | 17 | 9 |
| 156 | -87 | 40 | 23 | 1 | 1 | 0 | 0 |  | 4 | 0 | 1 | 1 | 0 | 6 | 17 | 9 |
| 156 | -87 | 41 | 23 | 2 | 1 | 0 | 0 |  | 4 | 0 | 1 | 1 | 0 | 6 | 17 | 9 |
| 156 | -87 | 41 | 23 | 2 | 1 | 0 | 0 |  | 4 | 0 | 1 | 1 | 0 | 6 | 17 | 9 |
| 177 | -87 | 41 | 26 | 3 | 1 | 0 | 0 |  | 5 | 1 | 1 | 1 | 0 | 8 | 22 | 5 |
| 177 | -87 | 41 | 26 | 3 | 1 | 0 | 0 |  | 5 | 1 | 1 | 1 | 0 | 8 | 22 | 5 |
| 177 | -87 | 41 | 26 | 3 | 1 | 0 | 0 |  | 5 | 1 | 1 | 1 | 0 | 8 | 22 | 5 |
| 177 | -87 | 41 | 26 | 1 | 0 | 1 | 0 |  | 5 | 1 | 1 | 1 | 0 | 8 | 22 | 5 |
| 177 | -87 | 41 | 26 | 2 | 0 | 0 | 1 |  | 5 | 1 | 1 | 1 | 0 | 8 | 22 | 5 |
| 177 | -87 | 41 | 26 | 2 | 0 | 0 | 1 |  | 5 | 1 | 1 | 1 | 0 | 8 | 22 | 5 |

The counters are reset between years.

## Additional features for the 60 closest traps

As in the previous section, 8 features are added. The approach is the same but only the 60 closest traps are taken into consideration.

# Modeling techniques and Training

All features are normalized to have zero mean, stddev 1.

The ensemble is composed of 24 random neural networks' architectures:

* 15 x 50+rand(50) x 2
* 15 x 50+rand(50) x 25+rand(25) x2

15 attributes are randomly selected (attribute bagging) for each NN among the 29 attributes (see section 2).

The weight of each record labeled as “0” is divided by 20 (see nnff.m line 52).

Each network is trained with cross-entropy loss and backpropagation, hidden layers have sigmoid activation and output layers have softmax activation.

# Leaderboard feedback

Additional multipliers are added based on the leaderboard feedback as described in the following tables.

## Date multipliers

|  |  |
| --- | --- |
| **Filter** | **Multiplier** |
| If WeekOfYear >= 39 | 0.2 |
| Else If Year == 2014 && WeekOfYear >= 35 && WeekOfYear <= 36 | 0.6 |
| Else If Year == 2012 && WeekOfYear == 26 | 0.8 |
| Else if Year == 2012 && WeekOfYear >= 28 && WeekOfYear <= 33 | 0.6 |
| Else | 0.4 |

## Species multipliers

|  |  |
| --- | --- |
| **Filter** | **Multiplier** |
| If Species == “CULEX PIPIENS” | 0.8 |
| Else If Species == “CULEX PIPIENS/RESTUANS” | 1.1 |

# File description

## train.py and predict.py scripts

These scripts are based on the “simple-lasagne-nn” Kaggle script available here: <https://www.kaggle.com/molozhenko/predict-west-nile-virus/simple-lasagne-nnb/run/15548>

The script train.py produces a model file “modellasagne.dat” used in the “predict.py” script.

The submission file “simplelasagna.tmp” generated with “predict.py” is used to create the final ensemble.

## preprocess.py script

This script produces a preprocessed file with new features (as described in section 2):

* A file preprocesstrain.csv during the training phase
* A file preprocesstest.csv during the predicting phase

## train.m and predict.m scripts

The script train.m produces a model file “modelNNOctave.dat” (as described in section 3) from the file processedtrain.csv. This model file is loaded in the “predict.m” script.

The submission file generated with “predict.m” is used to create the final ensemble.

## adjustProba.m

This script adds multipliers as described in section 4. This script also merges submissions files produced by “predict.py” and “predict.m” as following:

Modellasagne => 0.790 on private LB

modelNNOctave => 0.839 on private LB

modelNNOctave + additional multipliers => 0.874 on private LB

Final ensemble = (modelNNOctave + 0.2\*modellasagne) / 1.2 => 0.876 on private LB

# Dependencies

## Octave dependencies

* Octave 4.0.0 : <http://www.gnu.org/software/octave/download.html>
* DeepLearnToolbox : <https://github.com/rasmusbergpalm/DeepLearnToolbox>

## Python dependencies

The specific python dependencies are the ones used by the “simple-lasagne-nn” Kaggle script to improve the final ensemble score by approximately 0.002:

* Numpy
* Scikit-learn
* Lasagne
* Theano

# How to generate the solution

## Training step

The script train.bat takes 3 arguments:

* The training file
* The weather file
* The model filename

Example: C:\WestnileVirus>train.bat ./input/train.csv ./input/weather.csv model.dat

## Prediction step

The script predict.bat also takes 3 arguments:

* The test file
* The weather file
* The model filename

Example: C:\WestnileVirus>predict.bat ./input/test.csv ./input/weather.csv model.dat

A submission.csv file is created at the end of the process.

## Set the path

Octave and Python must be set in the path.

Example: set PATH=%PATH%;C:\Users\Psychedelic\Anaconda;C:\Octave\Octave-4.0.0\bin\