Location-Based Species Recommendation GeoLifeCLEF 2019 Challenge

Costel-Sergiu Atodiresei, Adrian Iftene

UAIC: Faculty of Computer Science, “Alexandru Ioan Cuza” University, Romania
sergiu.atodiresei@gmail.com, adiftene@info.uaic.ro

Abstract. In this paper, it is presented a system built with the aim to predict plant species given their location in the context of the GeoLifeCLEF 2019 challenge. There are developed several prediction models on the basis of a representation in the environmental space (a feature vector composed of climatic variables and other variables such as soil type, land cover, distance to water) and using the spatial occurrences of species (latitude and longitude): Random Forest, K-Nearest Neighbors, Artificial Neural Networks and XGBoost.

Keywords: GeoLifeCLEF, Random Forest, K-Nearest Neighbors, Artificial Neural Networks, XGBoost.

1 Introduction

Automatically predicting the list of species that are the most likely to be observed at a given location is useful for many scenarios in biodiversity informatics. First of all, it could improve species identification processes and tools by reducing the list of candidate species that are observable at a given location (be they automated, semi-automated or based on classical field guides or flora). More generally, it could facilitate biodiversity inventories through the development of location-based recommendation services (typically on mobile phones) as well as the involvement of non-expert nature observers. Last but not least, it might serve educational purposes thanks to biodiversity discovery applications providing functionalities such as contextualized educational pathways [1].

The aim of the challenge is to predict the list of species that are the most likely to be observed at a given location. In the previous edition, in the context of the GeoLifeCLEF 2018 challenge[1], several approaches for plant predictions given their location were used. The winners of the challenge 2018 have developed three kinds of prediction models, one convolutional neural network on environmental data (CNN), one neural network on co-occurrences data and two other models only based on the spatial

[1] https://www.imageclef.org/GeoLifeCLEF2019

occurrences of species. Results show the effectiveness of the CNN which obtained the best prediction score of the whole GeoLifeCLEF challenge. The fusion of this model with the spatial ones only provides slight improvements suggesting that the CNN already captured most of the spatial information in addition to the environmental preferences of the plants [2]. Location-based species prediction is very similar with the problem Species distribution modelling (SDM), also known as environmental (or ecological) niche modelling (ENM), habitat modelling, or predictive habitat distribution modelling, which uses computer algorithms to predict the distribution of a species across geographic space and time using environmental data. The environmental data are most often climate data (e.g. temperature, precipitation), but can include other variables such as soil type, water depth, and land cover. SDMs are used in several research areas in conservation biology, ecology and evolution. Predictions of current and/or future habitat suitability can be useful for management applications (e.g. reintroduction or translocation of vulnerable species, reserve placement in anticipation of climate change) [3].

In this paper are presented 4 machine learning models for predicting plant species in the context of the GeoLifeCLEF 2019 challenge [4]: (1) K-Nearest Neighbors: (i) Based on the environmental data (like temperature, precipitation, soil type, water depth, land cover etc.); (ii) Based on the spatial occurrences of species (a spatial model), (2) Random Forest: (i) Based on the environmental data; (ii) Based on the spatial occurrences of species, (3) Artificial Neural Networks based on the environmental data, (4) XGBoost: (i) Based on the environmental data; (ii) Based on the spatial occurrences of species. After splitting the train dataset in 90% (for training) and 10% (for testing), Random Forest and K-NN captured most of the environmental information.

2 Data and Evaluation Methodology

GeoLifeCLEF provides a large training set of species occurrences, each occurrence being associated to a multi-channel image characterizing the local environment. Indeed, it is usually not possible to learn a species distribution model directly from spatial positions because of the limited number of occurrences and the sampling bias.

The train dataset includes 280,945 train and test georeferenced occurrences of plant species from last year (file GLC_2018.csv). Plus, 2,367,145 plant species occurrences with uncertain identifications are added (file PL_complete.csv). They come from automatic species identification of pictures produced in 2017-2018 by the smartphone application Pl@ntNet, where users are mainly amateur botanists [5]. A trusted extraction of this dataset is also provided (file PL_trusted.csv), insuring a reasonable level of identification certainty. Finally, 10,618,839 species occurrences from other kingdoms (as mammals, birds, amphibians, insects, fungi's etc.) were selected from the GBIF database (file noPlant.csv). 33 environmental raster's (constructed from various open datasets) covering the French territory are made available, so that each occurrence may be linked to an environmental tensor via a participant customizable Python code [6]. The test occurrences data come from independents datasets of the French National Botanical Conservatories. This TestSet includes 844 plant species. It is a subset of those found in the train set. The main evaluation criteria will be the accuracy based on the 30 first answers, also called Top 30. It is the mean of the function scoring 1 when the good species is in the 30 first answers, and 0 otherwise, over all test set occurrences.

3 Proposed Solution

3.1 General Architecture of the System

The general architecture of the system is presented in **Fig. 1**. There is a **Processing Training Data Module**, which extracts and augments (with climatic variables and other variables such as soil type, land cover, distance to water, etc.) plant species occurrences with certain and uncertain identifications.

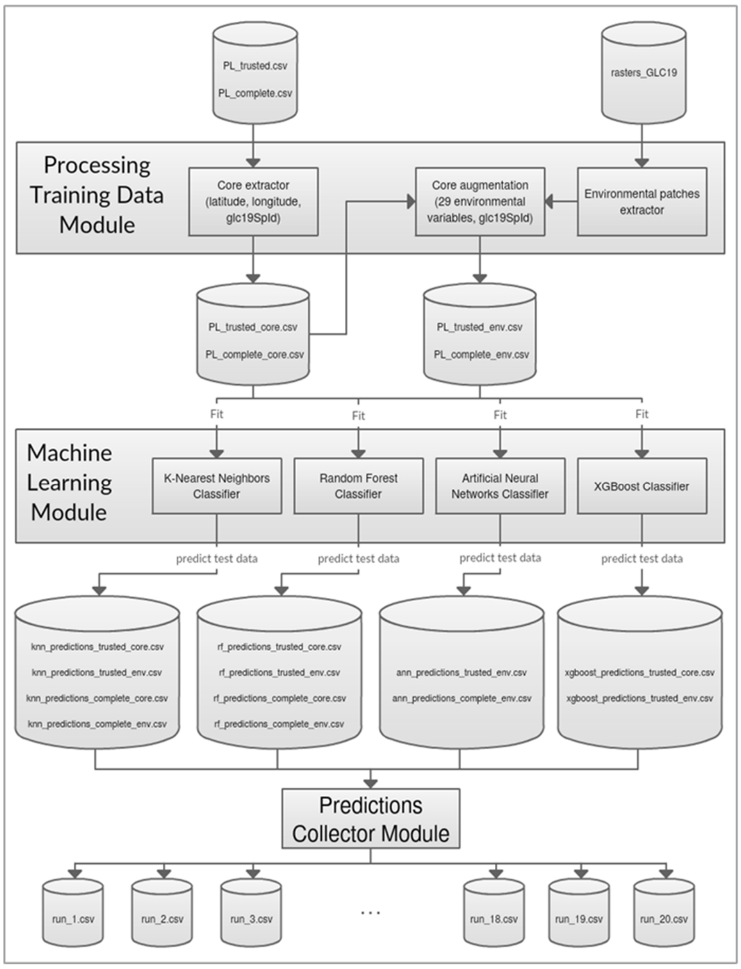
PL_trusted.csv
PL_complete.csv
rasters_GLC19
Processing Training Data Module
Core extractor (latitude, longitude, glc19SpId)
Core augmentation (29 environmental variables, glc19SpId)
Environmental patches extractor
PL_trusted_core.csv
PL_complete_core.csv
PL_trusted_env.csv
PL_complete_env.csv
Fit
Fit
Fit
Fit
Machine Learning Module
K-Nearest Neighbors Classifier
Random Forest Classifier
Artificial Neural Networks Classifier
XGBoost Classifier
predict test data
predict test data
predict test data
predict test data
knn_predictions_trusted_core.csv
knn_predictions_trusted_env.csv
knn_predictions_complete_core.csv
knn_predictions_complete_env.csv
rf_predictions_trusted_core.csv
rf_predictions_trusted_env.csv
rf_predictions_complete_core.csv
rf_predictions_complete_env.csv
ann_predictions_trusted_env.csv
ann_predictions_complete_env.csv
xgboost_predictions_trusted_core.csv
xgboost_predictions_trusted_env.csv
Predictions Collector Module
run_1.csv
run_2.csv
run_3.csv
...
run_18.csv
run_19.csv
run_20.csv

Fig. 1. System architecture

The output of this module consists of 2 .csv files. The first one is made of spatial occurrences of species (latitude and longitude) and the second one is composed of 29 environmental variables. This files are the input for the **Machine Learning Module** where our 4 machine learning models are used to predict 25,000 plant species occurences from the test data. Finally, the **Predictions Collector Module** generates 20 run files based on the output (32 .csv files) from the previous module, each run file containing the prediction of a particular model, or mixed predictions from several/all models. These runs has been submitted to be evaluated within the GeoLifeCLEF 2019 challenge.

3.2 Implementation Details

The system is based of 3 modules, developed in Python and using several libraries like Pandas (providing high-performance, easy-to-use data structures and data analysis tools) [7], NumPy (for scientific computing) [8], Scikit-learn (contains machine learning algorithms including K-Nearest Neighbors, Random Forest and also preprocessing) [9] similar to work from [10, 11], Keras (high-level neural networks API) [12], XGBoost (implements machine learning algorithms under the Gradient Boosting framework) [13].

3.3 Main Modules

I. Processing Training Data Module

This module processes 2 datasets: (1) PL_complete.csv with 2,367,145 plant species occurrences with uncertain identifications, (2) PL_trusted.csv - a trusted extraction of the complete dataset (approx. 10%) insuring a reasonable level of identification certainty.

The output consists in 4 files: (1, 2) PL_complete_core.csv, PL_trusted_core.csv - based on the spatial occurrences of species and have 3 columns containing in this order: latitude, longitude, glc19SpId. The values of latitude and longitude are rounded to 4 decimals to avoid overfitting as much as possible in the machine learning models, (3, 4) PL_complete_env.csv, PL_trusted_env.core - based on the environmental variables and have 30 columns. Each plant occurrence was linked to environmental tensors via a customizable Python code provided in the context of the challenge and **29** climatic variables and other variables were extracted: chbio_1, chbio_3, chbio_4, chbio_5, chbio_6, chbio_7, chbio_8, chbio_9, chbio_10, chbio_11, chbio_12, chbio_13, chbio_14, chbio_15, chbio_16, chbio_17, chbio_18, chbio_19, etp, alti, awc_top, bs_top, cec_top, crusting, dgh, dimp, erodi, oc_top, pd_top. More variables were available, but those didn't provide a clear added value. Each occurrence ID in the submitted run must exist in the testSet file (glc19TestOccId). Each predicted plant species (glc19SpId) must be one of the species marked as TRUE in the column "test" of the Table of species Ids and names and identification of test set species [14]. As a consequence, to predict the test dataset, the .csv files are filtered and contain only the occurrences of species marked as TRUE in the test set species.

II. Machine Learning Module

A. K-Nearest Neighbors Classifier

One of the machine learning models used is K-Nearest Neighbors Classifier, where the output is the class membership (e.g. plant species id - glc19SpId) most common among its k nearest neighbors [15]. The distance between neighbors is determined with the Euclidean metric.

Multiple K-NN models were developed with 1, 3, and 5 neighbors. No more than 5 neighbors were used because, as can be seen in the section 4, the accuracy decreases as many trees are used. Finally, the predictions from several/all models were merged, as stated in section 3.3.III.

This classifier generates 12 different predictions (.csv files), depending to the number of the nearest neighbors, if the training dataset has certain and uncertain identifications and based on the spatial occurrences of species or based on the environmental data. Considering that, the K-NN predicts the following lists of species: (1) knn_spatial_[1|3|5]nn_complete.csv - 1/3/5 nearest neighbor(s), the complete training dataset is used and is based on the spatial occurrences of species (longitude, latitude); (2) knn_spatial_[1|3|5]nn_trusted.csv - 1/3/5 nearest neighbor(s), the trusted training dataset is used and is based on the spatial occurrences of species (longitude, latitude); (3) knn_env_[1|3|5]nn_complete.csv - 1/3/5 nearest neighbor(s), the complete training dataset is used and is based on the environmental data; (4) knn_env_[1|3|5]nn_trusted.csv - 1/3/5 nearest neighbor(s), the trusted training dataset is used and is based on the environmental data.

B. Random Forest Classifier

Another machine learning algorithm used is Random Forest Classifier that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode (the value that appears most often) of the classes [16]. First, feature scaling is applied to the train and test data by removing the mean and scaling to unit variance. Then the Random Forest Classifier is fitted to the training set. The number of trees in the forest is 10 and the function to measure the quality of a split is "entropy" criteria for the information gain.

This classifier generates 12 different predictions (.csv files), depending to random state (the seed used by the random number generator), if the training dataset has certain and uncertain identifications and based on the spatial occurrences of species or based on the environmental data. Considering that, the K-NN model predicts the following lists of species: (1) random_forest_spatial_[1|2|3]_complete.csv - random state = 1/2/3, the complete training dataset is used and is based on the spatial occurrences of species (longitude, latitude); (2) random_forest_spatial_[1|2|3]_trusted.csv - random state = 1/2/3, the trusted training dataset is used and is based on the spatial occurrences of species (longitude, latitude); (3) random_forest_env_[1|2|3]_complete.csv - random state = 1/2/3, the complete training dataset is used and is based on the environmental data; (4) random_forest_env_[1|2|3]_trusted.csv - random state = 1/2/3, the trusted training dataset is used and is based on the environmental data.

A Random Forest Classifier with 8 number or trees was developed, but as stated in the section 4, the accuracy was smaller than the model with 10 trees.

C. *Artificial Neural Networks*

An artificial neural network is an interconnected group of nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it [17]. First, label binarizer is applied to the output variable (glc19SpId) to convert multi-class labels to binary labels (belong or does not belong to the class). Then feature scaling is applied to the train and test data.

A Neural Network model is based on the Sequential model which is a linear stack of layers. The model needs to know what input shape it should expect. In consequence, the input layer in a Sequential model (and only the first, because following layers can do automatic shape inference) is a Dense layer where the input shape is specified via the argument input_dim. In our case the input shape is equal to a feature vector shape (composed of 29 climatic variables and other variables such as soil type, land cover, distance to water) used to train this machine learning model. The Rectified Linear Unit function (relu) is passed to the input and hidden layers as the activation argument. For the output layer, the Softmax activation function is used. The system makes use of 3 Sequential models with a different number of hidden layers based on the geometric pyramid rule (the Masters rule):

a) for one hidden layer the number of neurons in the hidden layer is equal to:

$$nrHidden = \sqrt{nrInput \times nrOutput}$$

where *nrHidden* – the number of neurons in the hidden layer; *nrInput* – the number of neurons in the input layer; *nrOutput* – the number of neurons in the output layer.

b) for two hidden layers:

$$r = \sqrt[3]{nrInput \times nrOutput}$$
$$nrHidden1 = nrOutput \times \mathrm{r}^2$$
$$nrHidden2 = nrOutput \times r$$

where *nrHidden1* – the number of neurons in the first hidden layer; *nrHidden2* – the number of neurons in the second hidden layer.

c) for three hidden layers:

$$r = \sqrt[4]{nrInput \times nrOutput}$$
$$nrHidden1 = nrOutput \times \mathrm{r}^3$$
$$nrHidden2 = nrOutput \times \mathrm{r}^2$$
$$nrHidden3 = nrOutput \times r$$

where *nrHidden1* – the number of neurons in the first hidden layer; *nrHidden2* – the number of neurons in the second hidden layer; *nrHidden3* – the number of neurons in the third hidden layer.

In our case, we consider 3 models: (1) **the first model** - trained with observations from the trusted dataset has 29 neurons in the input layer, one for each environmental and climatic feature, 197 neurons in the hidden layer, and 1348 neurons in the output layer, one for each plant species id, then trained with observations from the complete dataset has 29 neurons in the input layer, one for each environmental and climatic feature, 336 neurons in the hidden layer, and 3096 neurons in the output layer, one for each plant species id, (2) **the second model -** trained with observations from the trusted dataset has 29 neurons in the input layer, 1089 neurons in the first hidden layer, 33 neurons in the second hidden layer, and 1348 neurons in the output layer, and then trained with observations from the complete dataset has 29 neurons in the input layer, 2304 neurons in the first hidden layer, 48 neurons in the second hidden layer, and 3096 neurons in the output layer, and (3) **the third model** - trained with observations from the trusted dataset has 29 neurons in the input layer, 2744 neurons in the first hidden layer, 196 neurons in the second hidden layer, 14 neurons in the third hidden layer, and 1348 neurons in the output layer, and then trained with observations from the complete dataset has 29 neurons in the input layer, 5832 neurons in the first hidden layer, 324 neurons in the second hidden layer, 18 neurons in the third hidden layer, and 3096 neurons in the output layer.

This classifier generates 6 different predictions (.csv files), depending to the model used and if the training dataset has certain and uncertain identification. Considering that, the ANN predicts the following lists of species: (1) ann_[1|2|3]_complete_env.csv - the 1st/2nd/3rd model is used and is trained with the complete dataset; (2) ann_[1|2|3]_trusted_env.csv - the 1st/2nd/3rd model is used and is trained with the trusted dataset.

D. XGBoost

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework [18].

First, feature scaling is applied to the train and test data by removing the mean and scaling to unit variance. Then the XGBClassifer (implementation of the scikit-learn API for XGBoost classification.) is fitted to the training set. The number of trees to fit is 100.

This classifier generates 2 different predictions (.csv files), based on the spatial occurrences of species or based on the environmental data:

- xgboost_env_trusted.csv - the trusted training dataset is used and is based on the environmental data;
- xgboost_spatial_trusted.csv - the trusted training dataset is used and is based on the spatial occurrences of species (longitude, latitude);

III. Predictions Collector Module

All the predictions lists (.csv files) are collected and processed to generate 20 run files, each run file containing the predictions of a particular model, or mixed predictions from several/all models. A run is a .csv file with 4 columns separated by ";" and containing in this order: glc19TestOccId ; glc19SpId ; Rank ; Probability. The models who predict the same plant species (glc19SpId) are combined in a single prediction (row in the .csv) with the probability being equal with the sum of probabilities associated with each model. These runs (including the algorithms who produced the predictions list) has been submitted to be evaluated within the GeoLifeCLEF 2019 challenge:

1) *run_1* (Visual retrieval type): K-Nearest Neighbors algorithm (1-NN) - Based on the environmental data (trusted dataset).
2) *run_2* (Visual retrieval type): K-Nearest Neighbors algorithm (1-NN) - Based on the environmental data (complete dataset).
3) *run_3* (Visual retrieval type): K-Nearest Neighbors algorithm (3-NN and 5-NN) - Based on the environmental data (trusted and complete datasets).
4) *run_4* (Textual retrieval type): K-Nearest Neighbors algorithm (1-NN) - Based on the spatial occurrences of species (trusted dataset).
5) *run_5* (Textual retrieval type): K-Nearest Neighbors algorithm (1-NN) - Based on the spatial occurrences of species (complete dataset).
6) *run_6* (Textual retrieval type): K-Nearest Neighbors algorithm (3-NN and 5-NN) - Based on the spatial occurrences of species (trusted and complete datasets). The probabilities associated with each model are the same as for *run_3*.
7) *run_7* (Mixed retrieval type): K-Nearest Neighbors algorithm (1-NN) - Based on the spatial occurrences of species and the environmental data (trusted and complete datasets).
8) *run_8* (Visual retrieval type): Random Forest algorithm - Based on the environmental data (trusted dataset).
9) *run_9* (Visual retrieval type): Random Forest algorithm - Based on the environmental data (complete dataset).
10) *run_10* (Visual retrieval type): Random Forest algorithm - Based on the environmental data (trusted and complete datasets).
11) *run_11* (Textual retrieval type): Random Forest algorithm - Based on the spatial occurrences of species (trusted dataset).
12) *run_12* (Textual retrieval type): Random Forest algorithm - Based on the spatial occurrences of species (complete dataset).
13) *run_13* (Textual retrieval type): Random Forest algorithm - Based on the spatial occurrences of species (trusted and complete datasets). The probabilities associated with each model are the same as for *run_10*.
14) *run_14* (Mixed retrieval type): Random Forest algorithm - Based on the spatial occurrences of species and the environmental data (trusted and complete datasets).
15) *run_15* (Mixed retrieval type): K-Nearest Neighbors algorithm (1-NN) and Random Forest algorithm - Based on the spatial occurrences of species and the environmental data (trusted dataset). The probability associated with each model is 0.5.
16) *run_16* (Mixed retrieval type): K-Nearest Neighbors algorithm (1-NN) and Random Forest algorithm - Based on the spatial occurrences of species and the environmental data (complete dataset). The probability associated with each model is 0.5.
17) *run_17* (Mixed retrieval type): K-Nearest Neighbors algorithm (1-NN) and Random Forest algorithm - Based on the spatial occurrences of species and the environmental data (trusted and complete datasets).
18) *run_18* (Visual retrieval type): Artificial neural networks (ANN) - Based on the environmental data (trusted and complete datasets). The ANN model with 1 and 2 hidden layers trained with complete dataset aren't used in this run because the

accuracy obtained in the validation experiments is lower than the model with 3 hidden layers.

19) *run_19* (Mixed retrieval type): XGBoost - Based on the spatial occurrences of species and the environmental data (trusted dataset). The probability associated with each model is 0.5.
20) *run_20* (Visual retrieval type): K-Nearest Neighbors algorithm (1-NN), Random Forest, Artificial neural networks (ANN), XGBoost - Based on the environmental data (trusted dataset).

4 Experiments and evaluation

4.1 Experimental results

To estimate the skill of a machine learning model the k-Fold Cross-Validation is used. Cross-validation is a statistical method used to evaluate machine learning models. It has a single parameter called k that indicates the number of groups that a given data sample is going to be split. Because of that, the method is called k-Fold Cross-Validation. The *KFold()* scikit-learn class [19] is used to split a given dataset into 10 consecutive folds (k=10). The *cross_val_score* scikit-learn function [20] is called on the classifier and the fold. The results of a k-fold cross-validation run are summarized with the mean of the model skill scores and the standard deviation obtained from the *cross_val_score* function [21].

For the **K-Nearest Neighbors** models the following results were obtained:

Dataset	**Number of neighbors**	**Mean**	**Standard deviation**
Based on the spatial occurrences of species (trusted dataset)	1	37.85%	0.27%
	3	22.86%	0.22%
	5	17.55%	0.21%
Based on the spatial occurrences of species (complete dataset)	1	26.58%	0.06%
	3	17.90%	0.06%
	5	14.48%	0.07%
Based on the environmental data (trusted dataset)	1	31.76%	0.26%
	3	20.84%	0.22%
	5	16.70%	0.28%
Based on the environmental data (complete dataset)	1	15.46%	0.05%
	3	12.11%	0.05%
	5	10.86%	0.03%

The best model is that with 1 nearest neighbor representing *run_4*. The differences between the model based on the environmental data and the one based on the spatial occurrences are quite significant. If the test set contains occurrences in areas without unnoticed species, the first model (trained with a feature vector composed of climatic variables and other variables such as soil type, distance to water) will do a better job in predicting the actual plant species. The complete dataset has three times more plant species than the trusted dataset so differences between the models trained with these datasets is normal considering that some species are present more often than others.

For the **Random Forest** models the following results were obtained:

Dataset	**Mean**	**Standard deviation**
Based on the spatial occurrences of species (trusted dataset)	36.88%	0.26%
Based on the spatial occurrences of species (complete dataset)	25.22%	0.08%
Based on the environmental data (trusted dataset)	30.20%	0.29%
Based on the environmental data (complete dataset)	13.89%	0.06%

The best model is based on the spatial occurrences of species (trusted dataset) representing *run_11*. The conclusion is the same as for the K-NN model. The differences between the model based on the environmental data and the one based on the spatial occurrences are quite considerable.

For the **Artificial Neural Network** models the following results were obtained:

Dataset	**Number of hidden layers**	**Mean**	**Standard deviation**
Based on the environmental data (trusted dataset)	1	6.46%	0.22%
	2	6.59%	0.27%
	3	6.71%	0.24%
Based on the environmental data (complete dataset)	1	6.28%	0.05%
	2	6.41%	0.03%
	3	6.65%	0.05%

The differences between models are negligible. The models built with 3 hidden layers have achieved slightly better results (*run_18*).

For the **XGBoost** models (*run_19*) the following results were obtained:

Dataset	**Mean**	**Standard deviation**
Based on the spatial occurrences of species (trusted dataset)	9.38%	0.28%
Based on the environmental data (trusted dataset)	8.24%	0.06%

4.2 Challenge Results

This year two criteria's where used to evaluate the runs:

- The accuracy based on the 30 first answers, also called Top30. It is the mean of the function scoring 1 when the good species is in the 30 first answers, and 0 otherwise, over all test set occurrences.

- The Mean Reciprocal Rank was chosen as secondary metric for enabling comparison with the 2018 edition.

The results of the submitted runs to GeoLifeCLEF 2019 are:

runId	**top30**	**runName**
26968	0.0205	run_14
26964	0.0191	run_10
26961	0.0190	run_7
26971	0.0184	run_17
26967	0.0180	run_13
26960	0.0168	run_6
27062	0.0159	run_20
26958	0.0146	run_3
26970	0.0102	run_16
26969	0.0099	run_15
26972	0.0089	run_18
26963	0.0079	run_9
26965	0.0068	run_11
26926	0.0067	run_4
26973	0.0064	run_19
26959	0.0063	run_5
26962	0.0062	run_8
26957	0.0061	run_2
26966	0.0058	run_12
26956	0.0058	run_1

The final results presented by the organizers show that the *run_14*, which is a list predicted by a Random Forest model based on the spatial occurrences of species and the environmental data (trusted and complete datasets), has achieved the best accuracy of our 20 submitted runs. The next one is the *run_10*, also a list predicted by a Random Forest model, this time trained with environmental data (trusted and complete datasets). The third one is the *run_7*, generated by the 1-NN model based on the spatial occurrences of species and the environmental data (trusted and complete datasets).

On the opposite side, the last places are occupied by *run_1*, *run_12* and *run_2*. These are produced by 1-NN model (based on the environmental data - trusted dataset), Random Forest model (based on the spatial occurrences of species - complete dataset), and again 1-NN model based on the environmental data, but this time using the complete dataset. The predicted plant species from *run_18 (*ANN) and *run_19* (XGBoost) have approximately half and one third, respectively, of the accuracy of the predicted list from *run_14*.

It appears that Random Forest and 1-Neareast Neighbors predict the plant species with the best accuracy, like in experimental results, only if a run contains a mixed predictions from several models.

5 Challenge Results Analysis

We cannot rely solely on the spatial appearances because of environmental differences (e.g. two species at 10 km can live in different environments, plain vs. sea coast). Another issue is that some species are present more often than others (e.g. plants that have been observed several times, others thousands of times), so we do not have uniform observations, especially in the complete training dataset as can be seen in the previous section (4). Last difficulty is that an occurrence means a punctual presence of that species, and if it is missing, it does not mean that the species does not exist. Unfortunately this leads to uncertainty in areas with unnoticed species.

An improvement would be to use more than 10 trees for the Random Forest algorithm to provide a prediction list of 30 species.

6 Conclusions

This paper details the system developed with the aim to predict plant species given their location for the 2019 edition of GeoLifeCLEF's challenge. We presented and compared four kinds of prediction models trained with the spatial occurrences of species and with the environmental data: Random Forest, K-Nearest Neighbors, Artificial Neural Networks and XGBoost. The results obtained using the Ranking metric and the Mean Reciprocal Rank show that the fusion of mixed predictions from several Random Forest models has the best prediction score of our 20 submitted runs, followed closely by mixed predictions from several K-Nearest Neighbors models. The Random Forest models captured most of the environmental information. For further development of the system the noPlant.csv dataset can be used to extract interesting correlations between plants and animals. This could lead to a more accurate prediction of plant species because some animals live only where certain plants are present.

Acknowledgement. This work is partially supported by POC-A1-A1.2.3-G-2015 program, as part of the PrivateSky project (P 40 371/13/01.09.2016).

References

1. GeoLifeCLEF 2019 (Usage scenario section): https://www.imageclef.org/GeoLifeCLEF2019 accessed last time on April, 2019.
2. Deneu, B., Servajean, M., Botella, C., Joly, A.: Location-based species recommendation using co-occurrences and environment - GeoLifeCLEF 2018 challenge: https://www.researchgate.net/publication/327542744_Location-based_species_recommendation_using_co-occurrences_and_environment-GeoLifeCLEF_2018_challenge accessed last time on April, 2019.
3. Elith, J., Leathwick, J. R.: Species Distribution Models: Ecological Explanation and Prediction Across Space and Time. In Annual Review of Ecology, Evolution, and Systematics, vol. 40 (1): pp. 677–697, (1009)

4. Botella, C., Servajean, M., Bonnet, P., Joly, A.: Overview of GeoLifeCLEF 2019: plant species prediction using environment and animal occurrences. In CLEF working notes 2019, (2019)
5. Affouard, A., Goeau, H., Bonnet, P., Lombardo, J.-C., Joly, A.: Pl@ntnet app in the era of deep learning. ICLR 2017-Workshop Track-5th International Conference on Learning Representations, pp. 1-6, (2017)
6. LifeCLEF 2019 Geo (Challenge description, Data and Evaluation sections): https://www.crowdai.org/challenges/lifeclef-2019-geo accessed last time on May, 2019.
7. Pandas Data Analysis Library: https://pandas.pydata.org accessed last time on May, 2019.
8. NumPy Package: https://www.numpy.org accessed last time on May, 2019.
9. Scikit-learn Machine Learning Library: https://scikit-learn.org/stable accessed last time on May, 2019.
10. Cușmaliuc, C. G., Coca, L. G., Iftene, A.: Identifying Fake News on Twitter using Naive Bayes, SVM and Random Forest Distributed Algorithms. Proceedings of The 13th Edition of the International Conference on Linguistic Resources and Tools for Processing Romanian Language (ConsILR-2018). ISSN: 1843-911X, pp. 177-188, (2018)
11. Șerban, C., Sirițeanu, A., Gheorghiu, C., Iftene, A., Alboaie, L., Breabăn, M.: Combining image retrieval, metadata processing and naive Bayes classification at Plant Identification 2013. Notebook Paper for the CLEF 2013 LABs Workshop - ImageCLEF - Plant Identification, 23-26 September, Valencia, Spain (2013)
12. Keras Deep Learning Library: https://keras.io accessed last time on May, 2019.
13. XGBoost Gradient Boosting Framework: https://keras.io accessed last time on May, 2019.
14. LifeCLEF 2019 Geo (Submission section): https://www.crowdai.org/challenges/lifeclef-2019-geo accessed last time on May, 2019.
15. Altman, N. S.: An introduction to kernel and nearest-neighbor nonparametric regression. In The American Statistician, vol. 46 (3), pp. 175–185, (1992)
16. Ho, T. K.: Random Decision Forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282, (1995)
17. Scherer, D., Müller, A. C., Behnke, S.: Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In 20th International Conference Artificial Neural Networks (ICANN), pp. 92–101, (2010)
18. Chen, T., Guestrin, C.: XGBoost: A Scalable Tree Boosting System. In Krishnapuram, Balaji; Shah, Mohak; Smola, Alexander J.; Aggarwal, Charu C.; Shen, Dou; Rastogi, Rajeev (eds.). Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. ACM. pp. 785–794, (2016)
19. McLachlan, G. J., Do, K. A., Ambroise, C.: Analyzing microarray gene expression data. Wiley, (2004)
20. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. San Mateo, CA: Morgan Kaufmann, vol. 2 (12), pp. 1137-1143, (1995)
21. Joly, A., Goëau, H., Botella, C., Kahl, S., Servajean, M., Glotin, H., Bonnet, P., Vellinga, W. P., Planqué, R., Stöter, F. R., Müller, H.: Overview of LifeCLEF 2019: Identification of Amazonian Plants, South \& North American Birds, and Niche Prediction. In Proceedings of CLEF 2019, (2019)