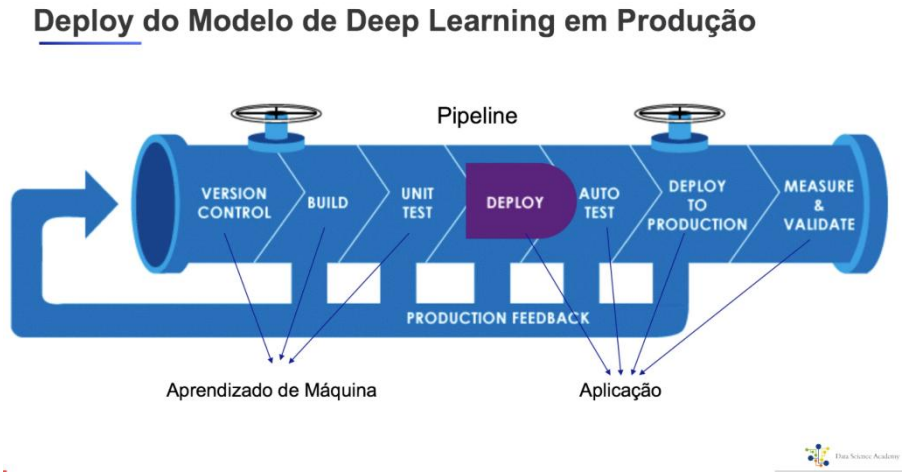
Text classification with Deep Learning

Occurrence forecasting models

Guttenberg Ferreira Passos

This article aims to classify texts and predict the categories of occurrences, through the study of Artificial Intelligence models, using Machine Learning and Deep Learning for the classification of texts and analysis of predictions, suggesting the best option with the smallest error.

The solution was designed to be implemented in two stages: Machine Learning and Application, according to the diagram below from the Data Science Academy.



Source: Data Science Academy

The focus of this article is the Machine Learning stage, the application development is out of scope and may be the object of future work.

The solution was applied to an agency in the State of Minas Gerais with the collection of three data sets containing 5,000, 100,000 and 1,740,000 occurrences respectively.

The project of elaboration of the algorithms of the Machine Learning stage was divided into four phases:

1) Development of a prototype for customer approval, with the Orange tool, for training a sample of 5,000 occurrences and forecasting 300 occurrences. In this step, Machine Learning algorithms were used.
2) Development of a Python program for training a sample of 100,000 occurrences and forecasting 300 occurrences. In this step, Deep Learning algorithms were used.
3) Training of a sample of 1,700,000 occurrences and prediction of 300 occurrences, using the same environment.
4) Training of a sample of 1,700,000 occurrences and forecast of 60,000 occurrences, using the same environment.

All models were adapted from the website https://orangedatamining.com/, of the videos: https://www.youtube.com/watch?v=HXjnDIgGDuI&t=10s&ab_channel=OrangeDataMining and the Data Science Academy Deep Learning II course classes: https://www.datascienceacademy.com.br

Machine Learning Algorithms Used:

- AdaBoost
- kNN
- Logistic Regression
- Naive Bayes
- Random Forest

Deep Learning Algorithms Used:

- LSTM - Long short-term memory
- GRU - Gated Recurrent Unit
- CNN - Convolutional Neural Networks

AdaBoost

It is a machine learning algorithm, derived from Adaptive Boosting. AdaBoost is adaptive in the sense that subsequent ratings made are adjusted in favor of instances negatively rated by previous ratings.

AdaBoost is sensitive to noise in data and isolated cases. However, for some problems it is less susceptible to loss of generalization ability after learning many training patterns (overfitting) than most machine learning algorithms.

kNN

It is a machine learning algorithm, the kNN algorithm looks for the nearest k training examples in the feature space and uses their mean as a prediction.

Logistic Regression

The logistic regression classification algorithm with LASSO regularization (L1) or crest (L2). Logistic regression learns a logistic regression model from the data. It only works for sorting tasks.

Naive Bayes

A fast and simple probabilistic classifier based on Bayes' theorem with the assumption of feature independence. It only works for sorting tasks.

Random Forest

Random Forest builds a set of decision trees. Each tree is developed from a bootstrap sample of training data. When developing individual trees, an arbitrary subset of attributes is drawn (hence the term “Random”), from which the best attribute for splitting is selected. The final model is based on the majority of votes from individually grown trees in the forest..

Source of Machine Learning Algorithms: Wikipedia and https://orange3.readthedocs.io/en/latest

LSTM

The Long Short Term Memory - LSTM network is a recurrent neural network, which is used in several Natural Language Processing scenarios. LSTM is a recurrent neural network (RNN) architecture that “remembers” values at arbitrary intervals. LSTM is well suited for classifying, processing, and predicting time series with time intervals of unknown duration. The relative gap length insensitivity gives LSTM an advantage over traditional RNNs (also called “vanilla”), Hidden Markov Models (MOM) and other sequence learning methods.

GRU

The Gated Recurrent Unit - GRU network aims to solve the problem of gradient dissipation that is common in a standard recurrent neural network. The GRU can also be considered a variation of the LSTM because both are similarly designed and in some cases produce equally excellent results.

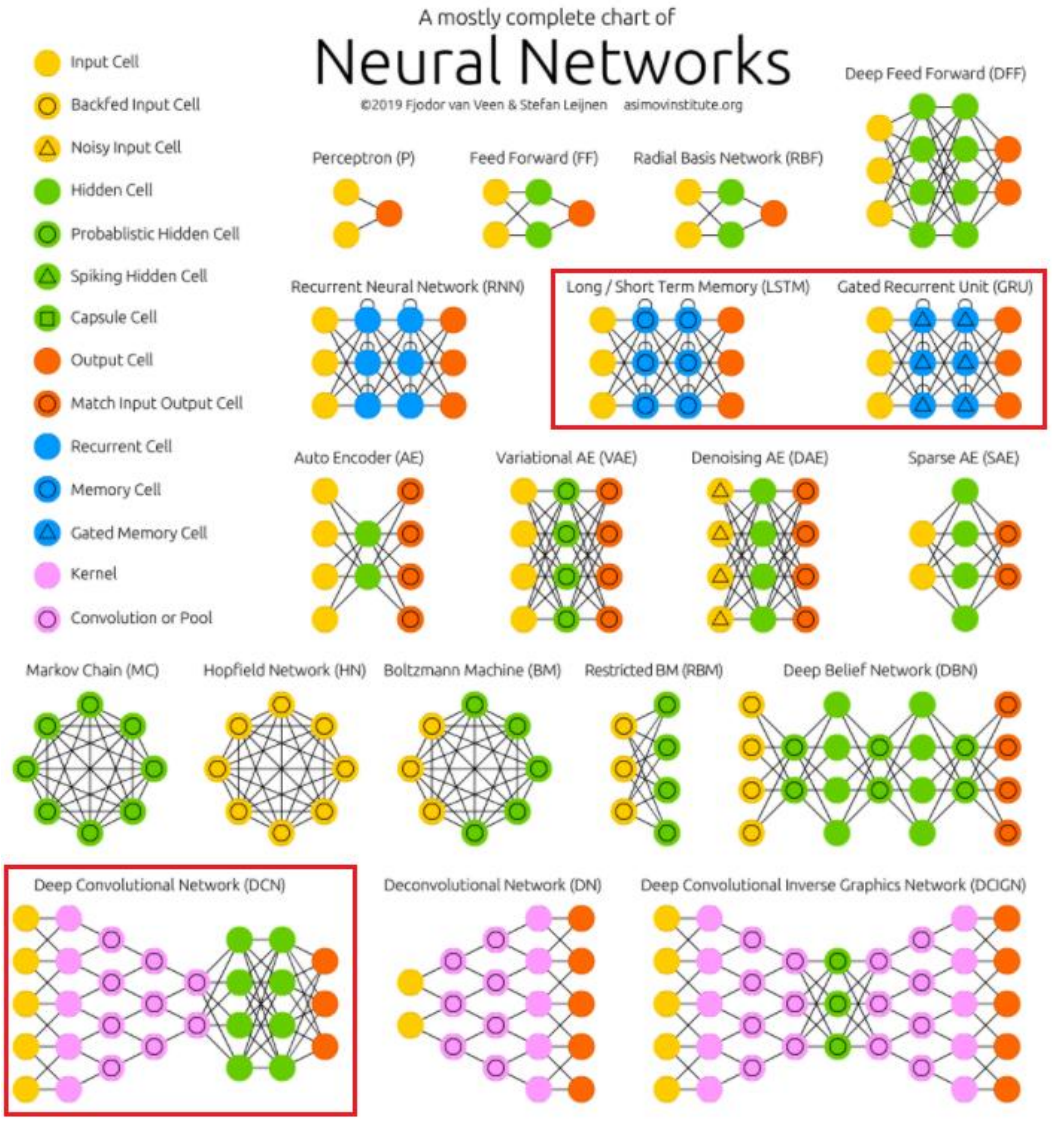
CNN

The Convolutional Neural Network - CNN is a Deep Learning algorithm that can capture an input image, assign importance (learned weights and biases) to various aspects/objects of the image and be able to differentiate one from the other. The pre-processing required in a CNN is much less compared to other classification algorithms. While in primitive methods filters are made by hand, with enough training, CNNs have the ability to learn these filters/features..

Source of Deep Learning Algorithms: https://www.deeplearningbook.com.br

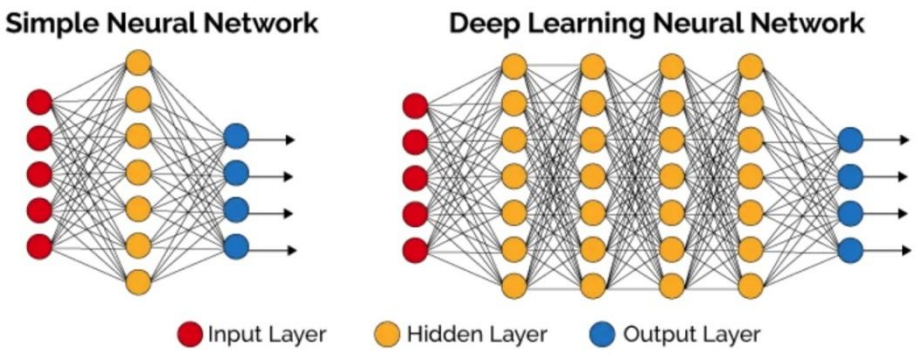
Neural networks are computing systems with interconnected nodes that function like neurons in the human brain. Using algorithms, they can recognize hidden patterns and correlations in raw data, group and classify them, and over time continually learn and improve.

The Asimov Institute https://www.asimovinstitute.org/neural-network-zoo/ published a cheat sheet containing various neural network architectures, we will focus on the architectures highlighted in red LSTM, GRU and CNN.



Source: THE ASIMOV INSTITUTE

Deep Learning is one of the foundations of Artificial Intelligence (AI), a type of machine learning (Machine Learning) that trains computers to perform tasks like humans, which includes speech recognition, image identification and predictions, learning over time. We can say that it is a Neural Network with several hidden layers:



Phase 1

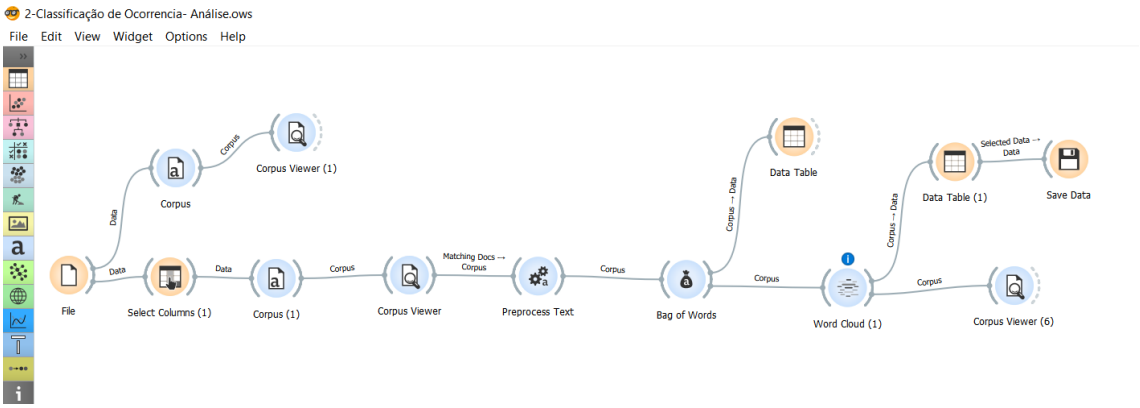
Phase 1 of the project is the development of a prototype for the presentation of the solution and its first approval by the customer. The tool that was chosen for this phase is Orange Canvas, because it is a more user-friendly graphical environment. In this environment, elements are dragged to the canvas without having to type lines of code.

The work begins with Exploratory Data Analysis. Initially, it was found that the first sample of 5,000 occurrences was unbalanced, as shown below. It was decided to discard the occurrences of the categories with the lowest volume of data.

The first phase was structured in three steps: pre-processing and data analysis, training the models and forecasting the categories of occurrences. The stages were planned to facilitate the development and implementation of the project as they are independent of each other and their processing is completed at each stage, not needing to be repeated in the later stage.

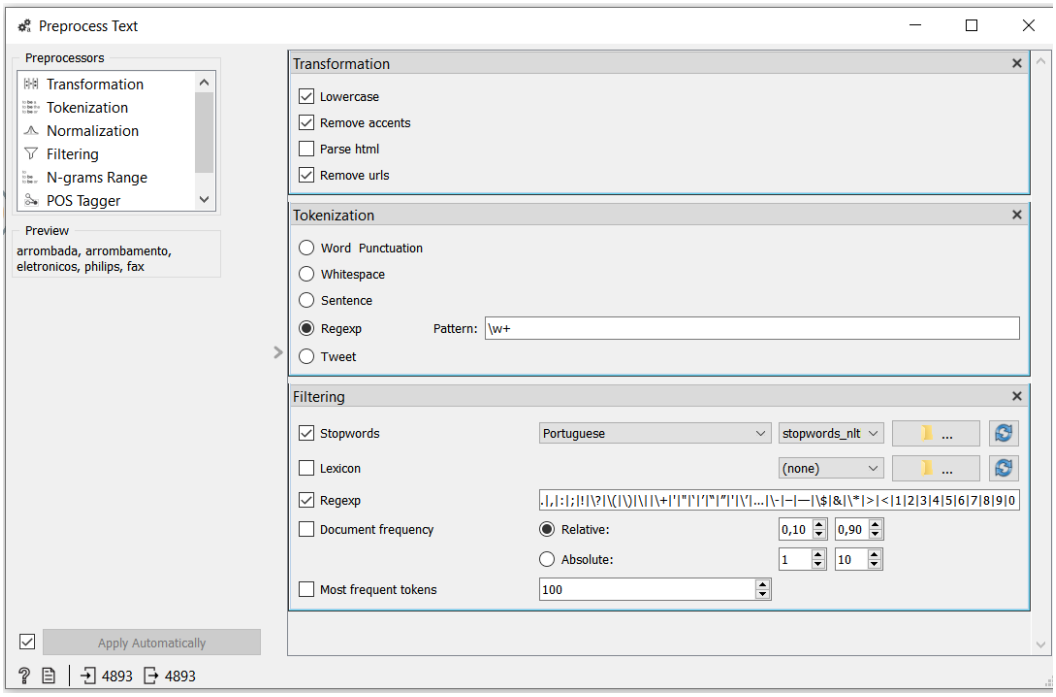
Phase 1 - Step 1: Pre-processing and data analysis

In the first stage, data collection, pre-processing and analysis are carried out, as shown in the figure below in the Orange tool.



Samples of 5,000 occurrences were collected to train the model and 300 occurrences to make the prediction, simulating a production environment.

After collection, the data are organized into a Corpus to carry out the pre-processing, performing the Transformation, Tokenization and Filtering actions of the data.



Words are arranged in a Bag of Words format, a simplified representation used in Natural Language Processing - NLP. In this model, a text (such as a sentence or a document) is represented as the bag of its words, disregarding grammar and even word order, but maintaining multiplicity.

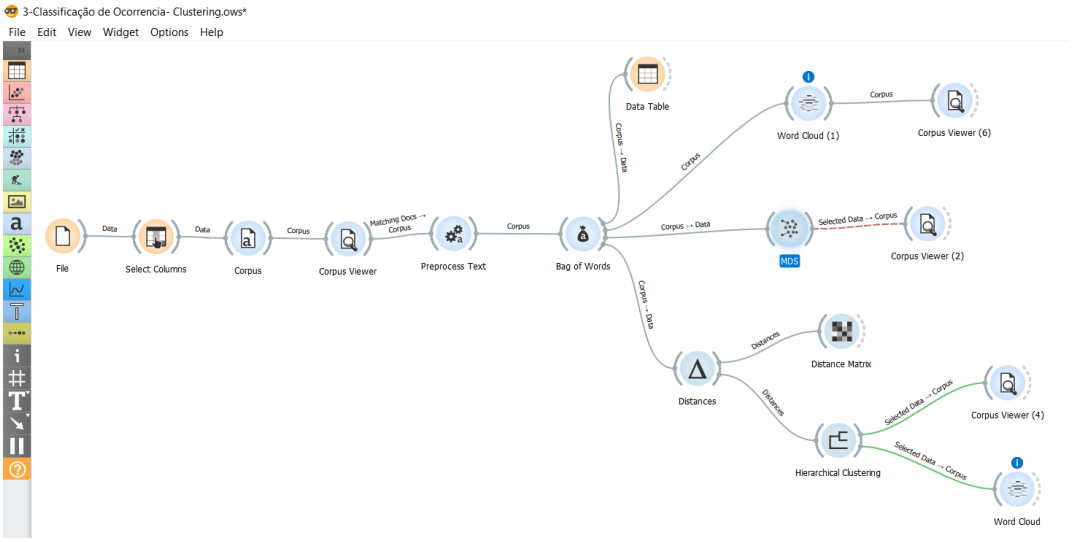
Machine learning is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. Machine learning explores the study and construction of algorithms that can learn from their errors and make predictions about data.

Machine learning can be classified into two categories:

Supervised learning: The computer is presented with examples of desired inputs and outputs.

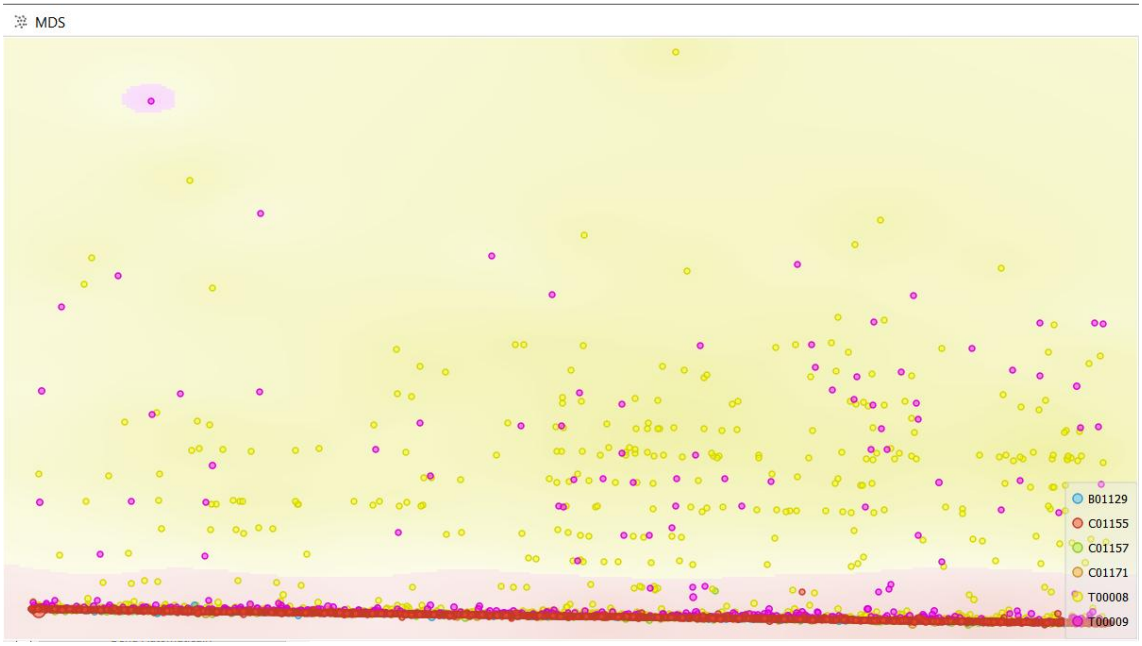
Unsupervised Learning: No tags are given to the learning algorithm, leaving it alone to find patterns in the given inputs.

Through unsupervised learning it is possible to identify the Clusters and their hierarchy.



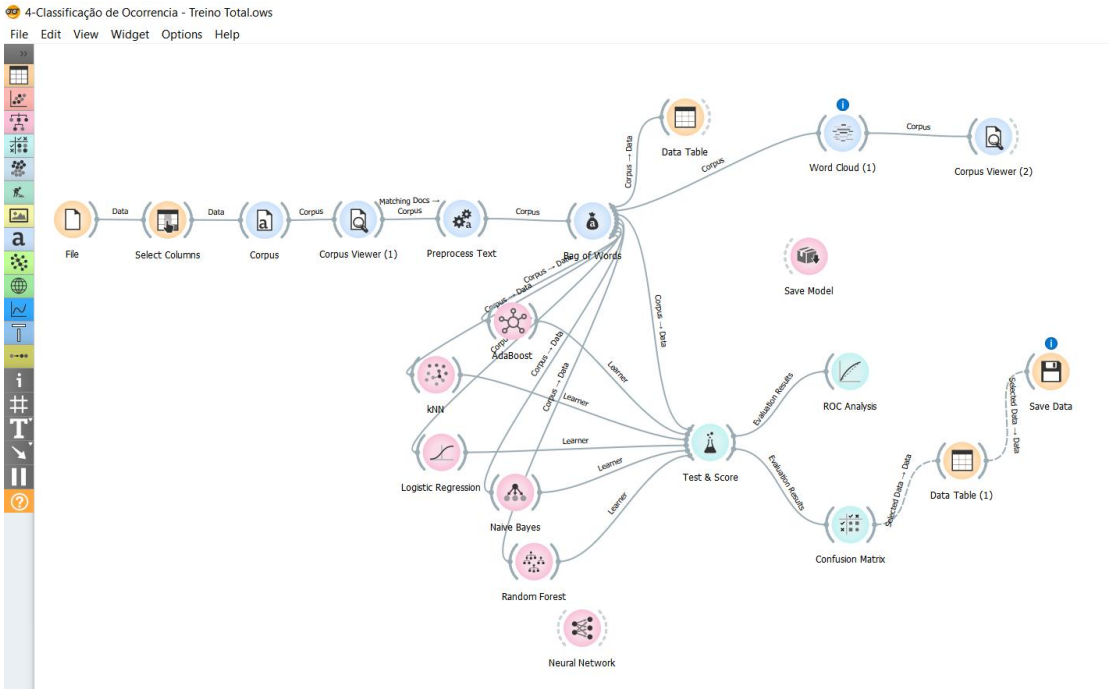
With multidimensional scaling (MDS) there is a way to visualize the level of similarity of individual cases of a data set and the regions of the Clusters.

In addition, there is also an idea of the ease or difficulty of the model in making its predictions, the more grouped the occurrences in a given region, the greater the probability of the model being correct.

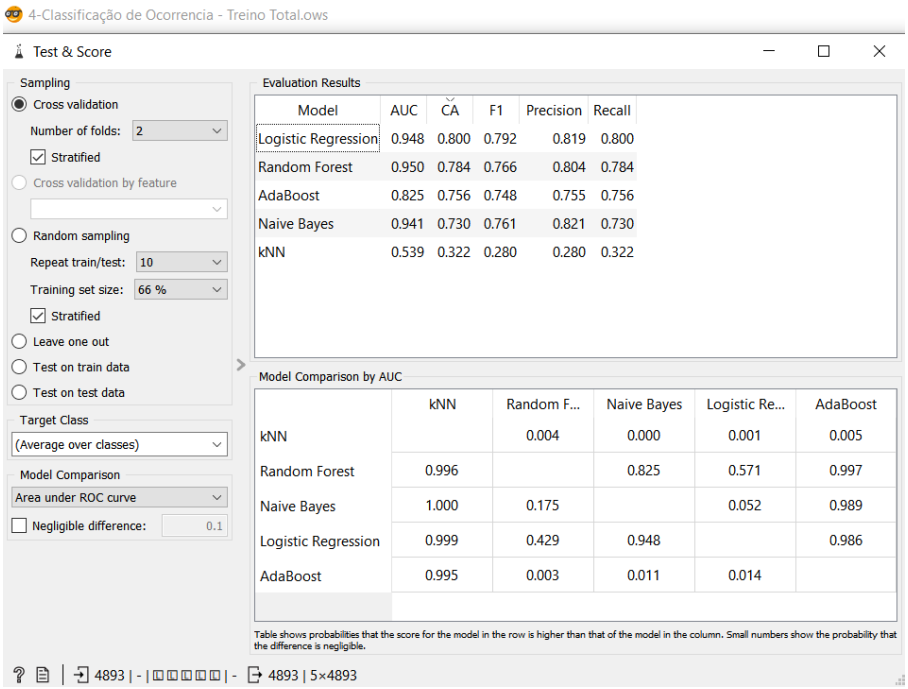


Phase 1 - Step 2: Model training

The second step is to train the models using the following Machine Learning algorithms: AdaBoost, kNN, Logistic Regression, Naive Bayes and Random Forest.



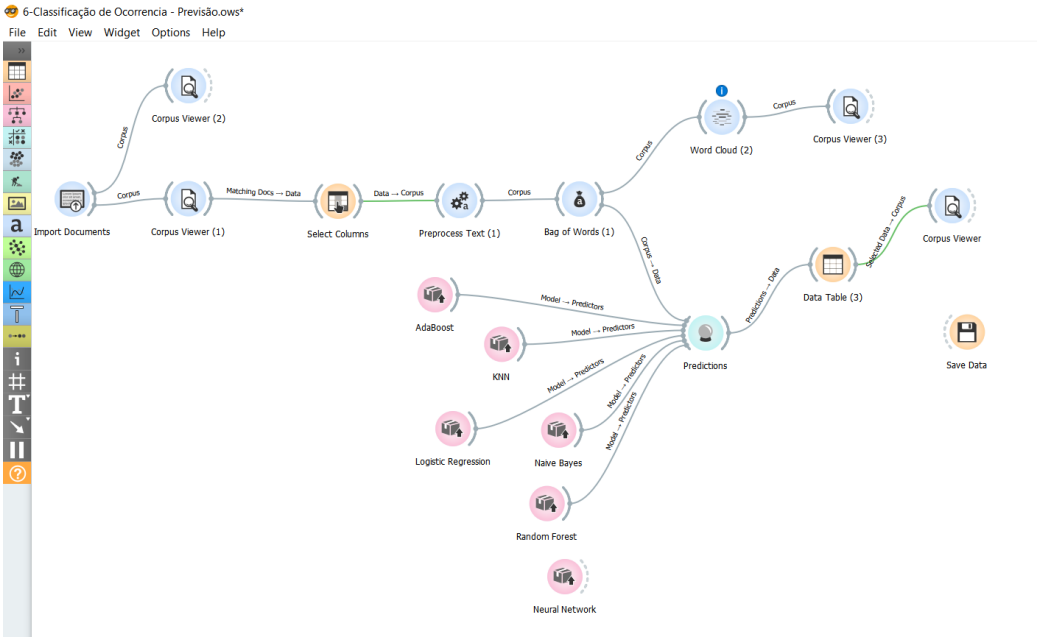
The overall performance of models can be measured through their Accuracy (AC), the proximity of a result to its real reference value. Thus, the greater the accuracy, the closer to the reference or real value is the result found.



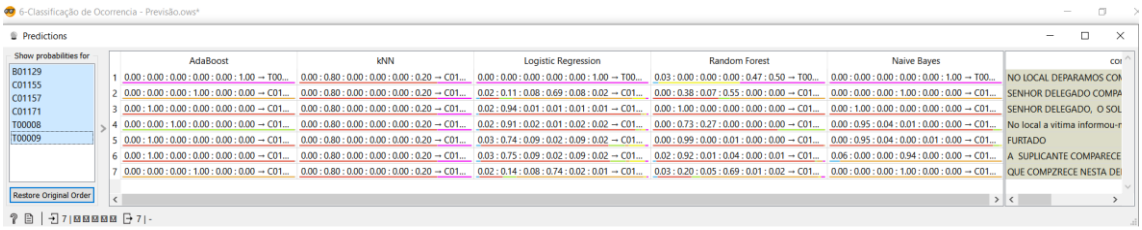
The successes and errors identified in the result can be analyzed through the Confusion Matrix. On the main diagonal of the matrix are the hits, correct predictions according to the real set. Errors are off the main diagonal, incorrect predictions according to the real set.

Phase 1 - Step 3: Forecasting the categories of occurrences

The last stage of the prototype, phase 1 of the project, is the prediction of the categories of occurrences, performed by each machine learning algorithm.



The result can be obtained through the probabilistic classification of observations, characterizing them in pre-defined classes. The predicted class will be the one with the highest probability:

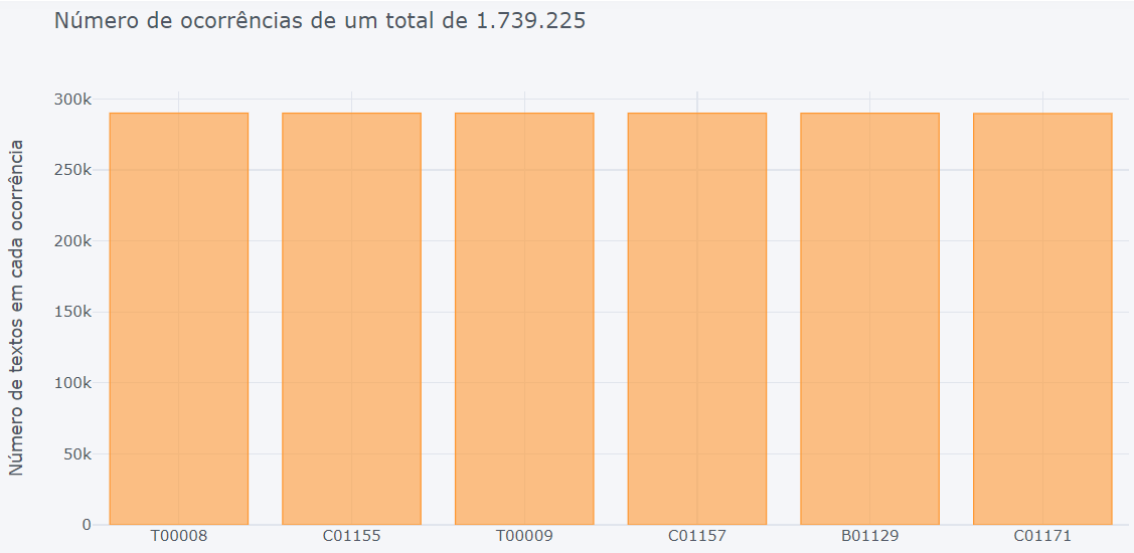


Phases 2, 3 and 4

For phases 2, 3 and 4 of the project, programs were developed in Python language for analysis, training and prediction of occurrences, using the following Deep Learning algorithms: LSTM, GRU and CNN.

Samples of 100,000 and 1,700,000 occurrences were provided for training and for prediction samples of 300 and 60,000 occurrences, using the same environment.

The new hit samples were preprocessed and balanced:



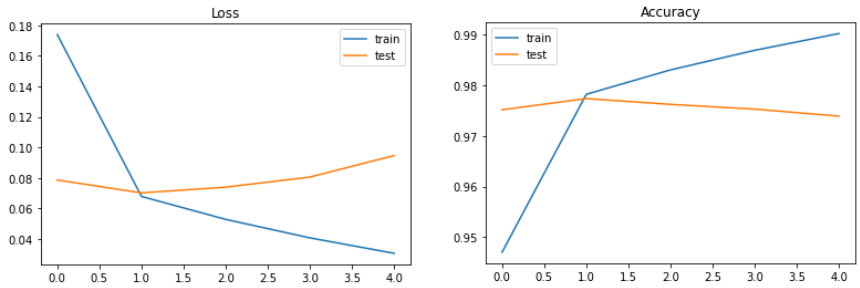
The programs developed were structured respecting the same three steps as in the previous phase: pre-processing and data analysis, training the models and forecasting the categories of occurrences.

In step 1, several data pre-processing techniques were used, similar to those used in the Orange Canvas environment.

In the second stage, different architectures were developed for each Deep Learning algorithm.

Model 1 LSTMs - Neural Network Layers:

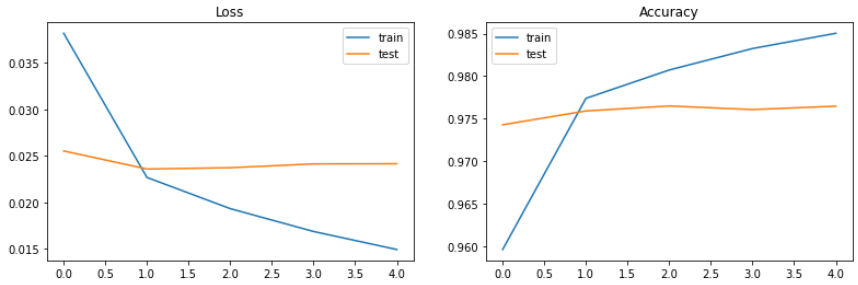
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 250, 100)          5000000
_________________________________________________________________
lstm (LSTM)                  (None, 100)               80400
_________________________________________________________________
dense (Dense)                (None, 6)                 606
=================================================================
Total params: 5,081,006
Trainable params: 5,081,006
Non-trainable params: 0
```



Model 2 LSTMs and CNNs - Neural Network Layers:

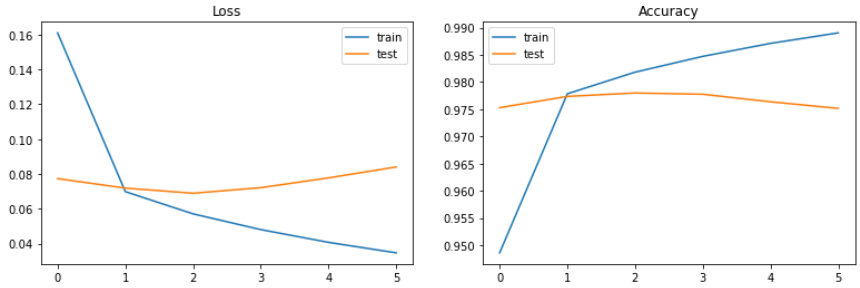
```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 250, 100)          5000000

conv1d (Conv1D)              (None, 250, 32)           9632
_________________________________________________________________
max pooling1d (MaxPooling1D) (None, 125, 32)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 125, 100)          53200
```

```
lstm_2 (LSTM)                (None, 100)               80400
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 606
=================================================================
Total params: 5,143,838
Trainable params: 5,143,838
Non-trainable params: 0
```



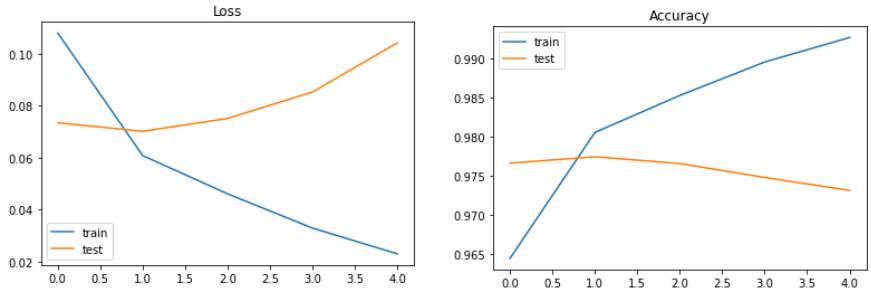
Model 3 LSTMs with Dropout - Neural Network Layers:

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 250, 100)          5000000
_________________________________________________________________
lstm_3 (LSTM)                (None, 250, 200)          240800
_________________________________________________________________
lstm_4 (LSTM)                (None, 200)               320800
_________________________________________________________________
dense_2 (Dense)              (None, 6)                 1206
=================================================================
Total params: 5,562,806
Trainable params: 5,562,806
Non-trainable params: 0
```



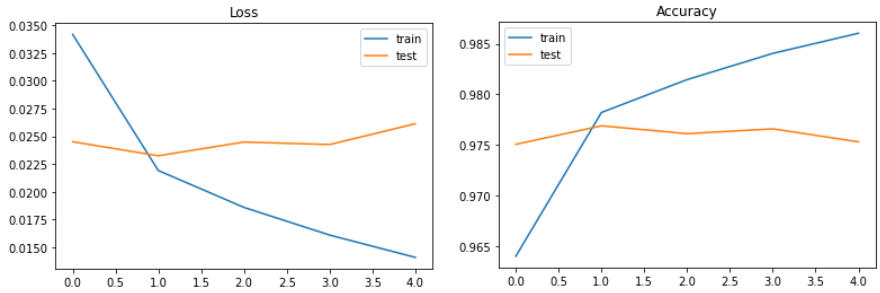
Model 4 GRU - Layers of the Neural Network:

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 250, 100)          5000000
_________________________________________________________________
gru (GRU)                    (None, 100)               60600
_________________________________________________________________
dense (Dense)                (None, 6)                 606
=================================================================
Total params: 5,061,206
Trainable params: 5,061,206
Non-trainable params: 0
```



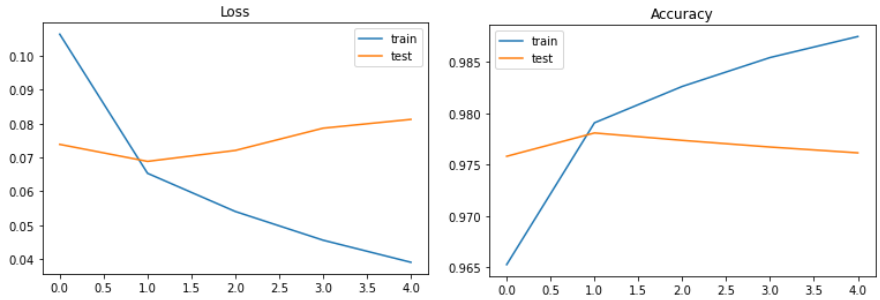
Model 5 GRU and CNN - Neural Network Layers:

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 250, 100)          5000000

conv1d (Conv1D)              (None, 250, 32)           9632
_________________________________________________________________
max pooling1d (MaxPooling1D) (None, 125, 32)           0
_________________________________________________________________
gru 1 (GRU)                  (None, 125, 100)          40200
_________________________________________________________________
gru_2 (GRU)                  (None, 100)               60600
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 606
=================================================================
Total params: 5,111,038
Trainable params: 5,111,038
Non-trainable params: 0
```



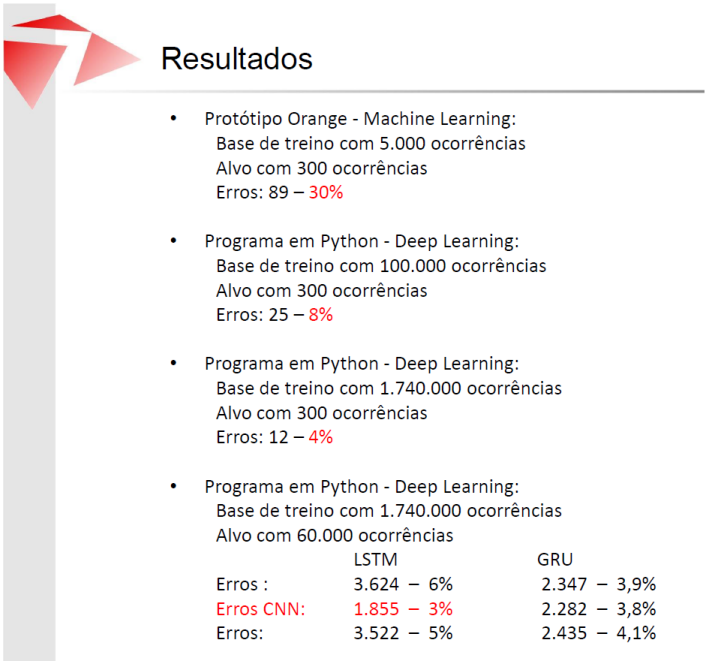
Model 6 GRU with Dropout - Neural Network Layers:

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (None, 250, 100)          5000000
_________________________________________________________________
gru_3 (GRU)                  (None, 250, 200)          181200
_________________________________________________________________
gru 4 (GRU)                  (None, 200)               241200
_________________________________________________________________
dense_2 (Dense)              (None, 6)                 1206
=================================================================
Total params: 5,423,606
Trainable params: 5,423,606
Non-trainable params: 0
```



Conclusion

In this work, without any pretension of exhausting the subject, it was demonstrated that the models based on Deep Learning had a better result than the other algorithms, as shown below:



The performance achieved by the combination of the LSTM and CNN algorithms is considered excellent, with an accuracy of 97%. It is therefore recommended to adopt this model for the development of the Occurrence Forecast application in production.