Capstone Project

Machine Learning Engineer Nanodegree

Igor Oliveira

October Xth, 2017

Machine Learning for Fog Forecasting

Project Overview

This project uses meteorological observations data to forecast the occurrence of fog, an extremely important low visibility event that have profound impact in air transport operations. This application of Machine Learning has been used before, specially the use of Artificial Neural Networks (ANN). The existing research in this field provides a good guidance about the type of data to be considering in the training process. It also provides background about the good performance of ANN methods when compared to traditional techniques.

Problem Statement

Most airports worldwide have meteorological observation systems, where atmospheric variables like temperature, humidity, wind, visibility and others. This information is routinely gathered by automatic or manual sensors and reported as METAR (Meteorological Terminal Aviation Routine Weather Report) messages. A typical METAR contains data for the temperature, dew point, wind direction and speed, precipitation, cloud cover and heights, visibility, and barometric pressure. A METAR may also contain information on precipitation amounts, lightning, and other information that would be of interest to pilots or meteorologists such as a pilot report or PIREP, colour states and runway visual range (RVR).

In this work, we will use METAR data from 2 airports in South America (Montevideo and Buenos Aires) to train an Artificial Neural Network for Visibility prediction. In aviation, the occurrence of low visibility events is one of the most important meteorological impacts for airports operation. The meteorological definition states that fog is a low visibility event that happens when visibility drops to levels below 1km. Using a long time series of METAR data for training, we built a classifier for events of visibility below 1km. The following predictors were extracted from the METAR data:

- Air Temperature
- Dew Point Temperture
- Relative Umidity
- Wind speed and Direction
- Atmospheric Pressure

Metrics

Since the problem in this work is a classification problem with two “types": 'fog' or 'normal visibility’, it is useful to build a two-dimensional contingency table for fog occurrence and compute a few statistical indexes. The table is built as below:

Table 1 - Confusion Matrix

		Observed Events	
		Fog	**No Fog**
Predicted Events	**Fog**	**TP**	**FP**
	No Fog	**FN**	**TN**

Where:

- TP = true positive
- TN = true negative
- FP = false positive
- FN = false negative

These quantities allow the computation of some indexes, such as:

- Sensitivity, recall, hit rate, or true positive rate (TPR):

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$$

- Specificity or true negative rate (TNR):

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP}$$

- Precision or positive predictive value (PPV):

$$PPV = \frac{TP}{TP + FP}$$

- Miss rate or false negative rate (FNR):

$$FNR = \frac{FN}{P} = \frac{FN}{TN + FN} = 1 - TPR$$

Precision and Recall will be the metrics considered in this study due to the operational value they bring for evaluating the algorithm performance. It is extremely useful for the airport operators to asses to fraction of occurrences of low visibility events and also useful to know from the amount of positive forecasts, the likelihood they are right.

Data Exploration

Both airports considered for this work are located in the Atlantic Coast of South America, near the Rio de la Plata basin between Argentina and Uruguay (Figure 1). Time series METAR data from mid-year 2011 until September 2017.

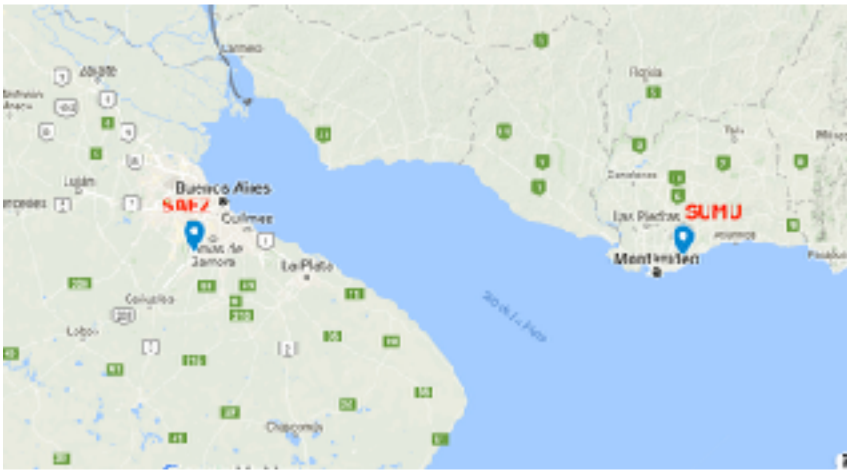

Figure 1 - Geographic location indicated with blue markers for Buenos Aires (SAEZ) and Montevideo (SUMU) airports.

A typical METAR observations contains information on temperature (tmpf), dewpoint(dwpf), relative humidity (relh), wind direction (drct), wind speed(sknt), pressure (alti) and visibility (vsby). An example row of the data can be seen below:

	Unnamed: 0	valid	tmpf	dwpf	relh	drct	sknt	alti	vsby
0	0	2011-01-01 02:00:00	68.0	62.6	82.86	100.0	10.0	29.97	6.21

All METAR data used in this project was acquired from the Iowa State University Website (https://mesonet.agron.iastate.edu/request/download.phtml)

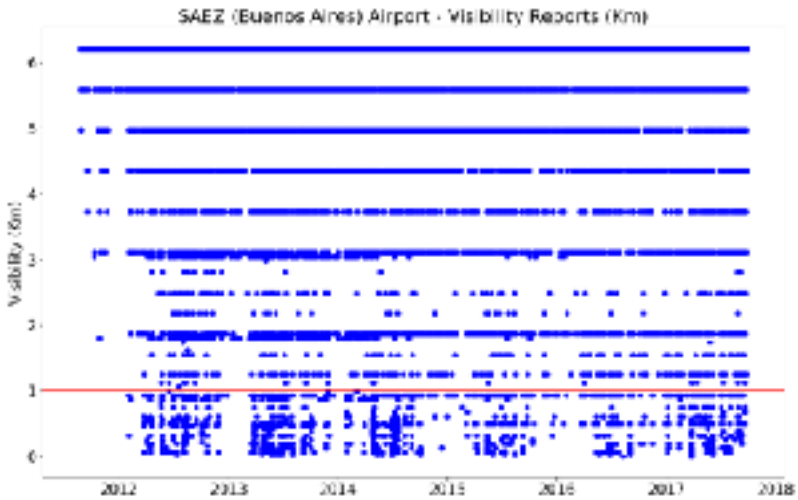

Figure 2 - Visibility reports for SAEZ airport from mid 2011 until present date. Red line indicates 1km visibility treshold

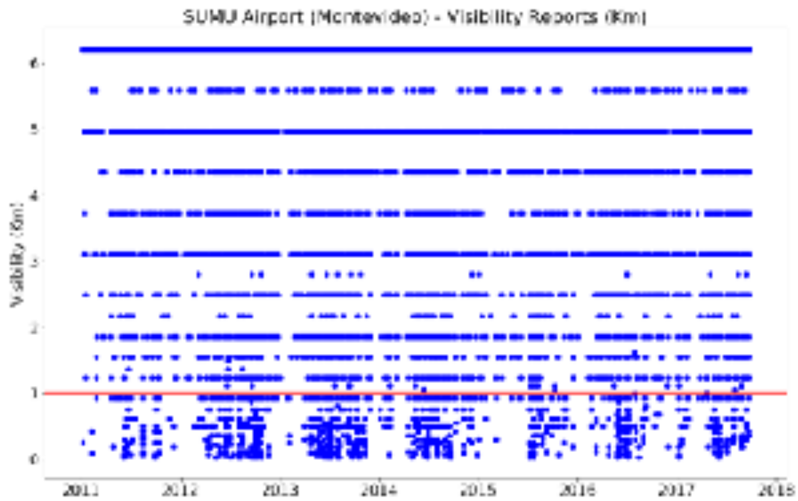

Figure 3 - Same as 2 for SUMU airport

Figures 2,3 and Table 2 describe the distribution of visibility reports for the selected period for the airports of SAEZ and SUMU. In METAR reports, Visibility is reported in kilometres indicating the horizontal distance of clear view. Values of visibility below 1km are being considered for this study a ‘low visibility event’, due to its operational effects in airports routine. This threshold is marked with a red line in figures 2 and 3.

Airport	# of VSBY < 1km reports	Total # of reports
SUMU	1519	45774
SAEZ	1839	37356

Table 2 - Data distribution for both airports

Algorithms and Techniques

For this projects, deep neural networks (DNN) will be used. DNN are specially useful for classification problems when we have a good-sized dataset with label data and a physical problem with high non-linearity. these seems to be a good choice of ML method to use. Neural networks are algorithms consisted of layers (usually called ‘nodes’) which combine a set of inputs and coefficients (called ‘weights’) from previous layers or from the original dataset in the case of the first layer (called ‘Input Layer’). The product of weights and inputs are passed through an activation function that will determine if the signal will be passed through that particular node which will eventually reach the output layer. Initial network architectures usually consisted of an input layer, a hidden layer and an output layer. DNN architectures differentiate themselves by adding more hidden layers (adding depth), as seen in figure 4

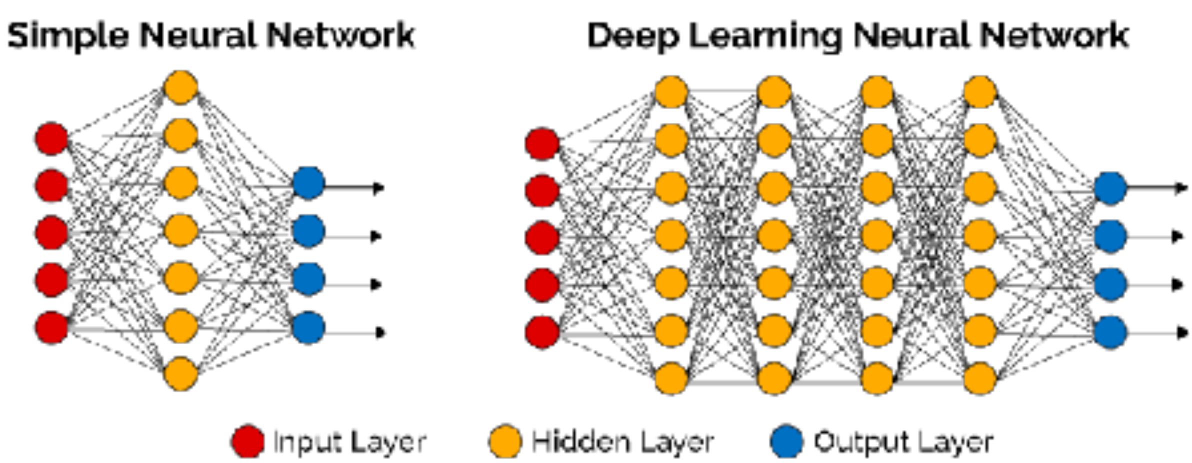

Figure 4 - Network Architectures

Benchmark

This work will consider the previous effort from the article "Application of Artificial Neural Network Forecasts to Predict Fog at Canberra International Airport” (Fabbian et al 2005) (https://doi.org/10.1175/WAF980.1) as benchmark. The referenced work is result of a much longer research, considering much more data sources and got useful operational results with 18-h lead time. For this work, we will consider a benchmark of reaching a value of at least 0.5 for precision and recall index with 3-h lead time as operational useful based on the references suggested by the study

Methodology

Data Preprocessing

In order to produce an operationally viable fog forecast, the algorithm must be trained with METAR data gathered before the occurrence of the low-visibility event. Therefore, it becomes necessary to re-organize the input data reports by producing extra columns containing the data observed 1 to 8 hours before each report. This was achieved using the `prepare_data.py` program, which reads the original METAR data containing hourly reports of Temperature (tmpf), Dewpoint (dwpf), Relative Humidity (relh), Wind Direction (drct), Wind Speed (sknt), Barometric Pressure (alti), Visibility (vsby) and adding to the same line of each report the same variables observed 1 to 12 hours before. In terms of new features, this step means producing 84 new predictors.

Although METAR data is gathered hourly, when significant weather events occurs between full hours (1200, 1300, 1400, etc) an special report is issued. To account for lagged observation for these non-full hours it was considered the time closest to the report hour minus the desired lag. For example, if a special METAR report is gathered at 12:51, the 1 hour-lag observation should be 11:51 (which is inexistent), so the closest full hour of 12:00 is considered.

Extra-steps in `prepare_data.py` were also taken to remove any outlier and unreasonable data from the METAR data files. Unreasonable data points such as relative humidity above 100%, negative wind speeds or pressure near zero were removed. Data as also normalised with sklearn ‘StandardScaler()' function.

Implementation & Improvements

The datasets for Montevideo and Buenos Aires airports were splitted in test (25% of total samples), validation (25%) and training sets (50%). This split was performed using the `test_train_split` function from the sklearn package in such a way that the new sets had a similar distribution compared with the original dataset. Training took place with a batch size of 1024 and 20 epochs and after each epoch the model was tested against the validation set. If the validation loss improved after a given epoch, the weights for the model were saved. The value of 20 epochs was a typical value to accommodate the validation loss improvements after each one.

An ANN for fog forecasting was made using a architecture with only a few hidden layers. The final architecture configuration was achieved after several manual attempts for classification improvement. A few deep learning structures were experimented with no accuracy improvement (like Dropout layers, which added to value to the algorithm). Manual attempts were also performed for tuning of batch_size, number of epochs and optimizer. The results improved with the addition of extra layers. The values for Recall and Precision showed a slight improvement (not shown) as hidden layers were added until the final configuration with 8 layers (7 hidden layers + 1 output layer), the code for network configuration is shown on the snippet bellow:

```
def make_model(shape):
    model = Sequential()
    model.add(Dense(64, input_dim=shape, activation='relu', kernel_initializer = 'normal'))
    model.add(Dense(64, activation='relu'))
    model.add(Dense(128, activation='relu'))
    model.add(Dense(256, activation='relu'))
    model.add(Dense(128, activation='relu'))
    model.add(Dense(64, activation='relu'))
    model.add(Dense(32, activation='relu'))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
```

Results

To evaluate the model performance, a confusion matrix was created for each airport, and two metrics were selected. Recall, which informs about the frequency of positive observations when the forecast is positive, and precision, which informs about the frequency of positive forecasts when the observation is positive. We were interesting in investigating the role of 'lead time' in the classifier performance, to detect if we could forecast a low visibility event with a few hours of antecedence. Results for Real and precision as a function of lead time for both airports are shown in figure 5

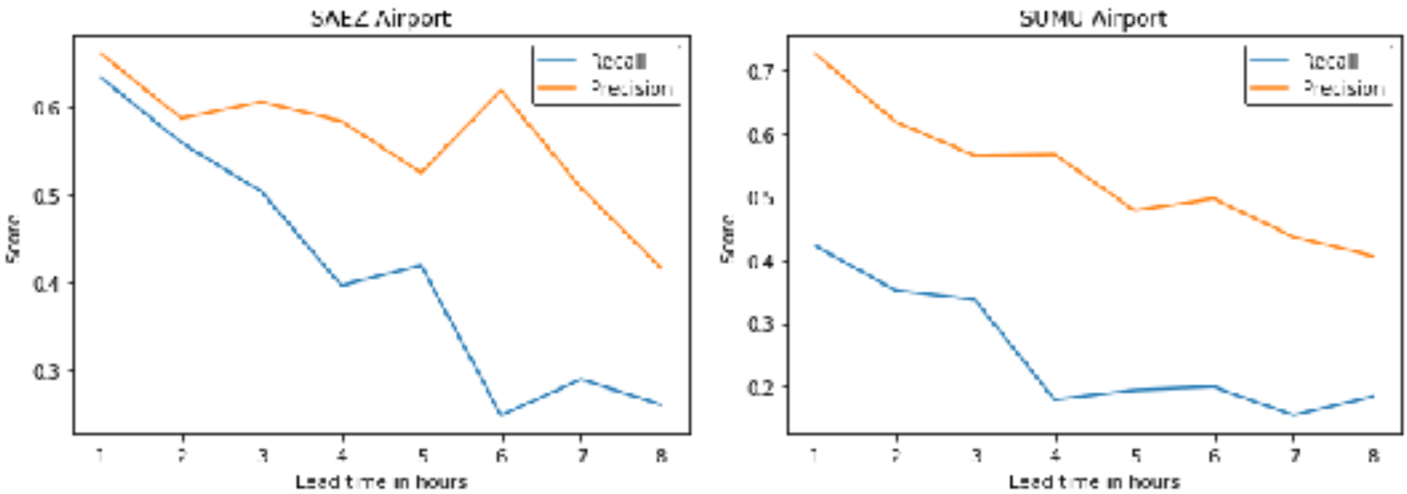

Figure 5 - Precision and Recall metrics for both Montevideo and Buenos Aires airports as a function of lead time

To validate the model robustness, it was made a simple test to perform forecasts using a 'never before seen' dataset, and for that the Curitiba Airport (SBCT) data was used. It is valid to note that the physical process for fog occurrence may be different. However, the results for Recall were similar (Figure 6) and just slightly different for Precision, indicating a proper model configuration.

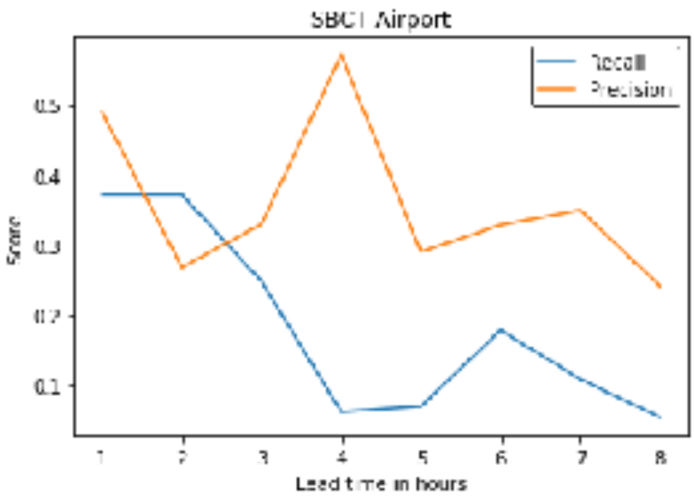

Figure 6 - Precision and Recall metrics for Curitiba Airport

Justification

The results achieved in this study are comparable to the proposed benchmark study performed in Fabbian et al 2005. In that study, a much larger dataset was avail for training (44-years of observation in the Canberra Airport), which was probably the main reason for the higher scores. However, it is possible to note a similar decay in performance as the lead time in increased, as can bee seen in figure 7, showing the ROC curves for Canberra fog forecasting. Considering the smaller dataset used in the present work these results show an appropriate performance, although retraining with larger datasets would probably yield a more suitable model for operational use.

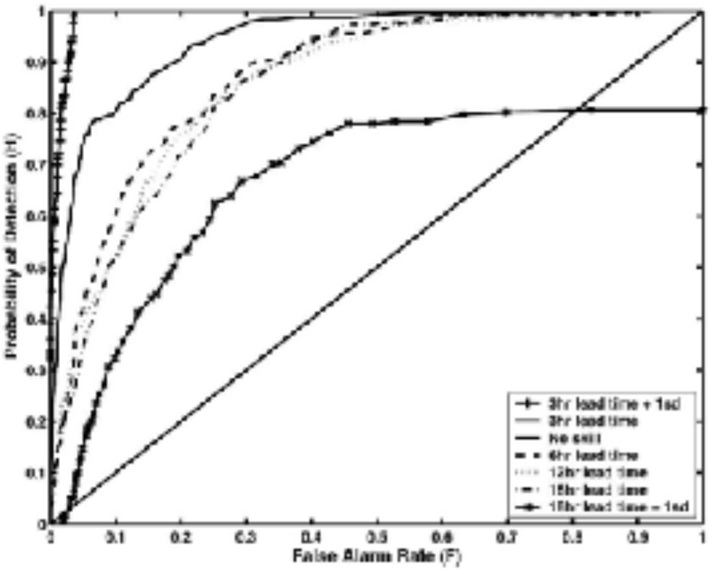

Figure 7 - ROC curves for the optimized Canberra fog forecasting neural nets. Source: Fabbian et al 2005

Conclusion

It was detected that lead time is a very sensitive parameter for fog detection using artificial neural networks. With more temporal proximity to the event, the training data carries more information and can explain better the physics behind the phenomena of atmospheric low visibility. This algorithm probably still lacks performance for an operational use, however it showed very promising for future development. The performance of the algorithm would significantly increase if other features were included in the training, validation and testing sites. The dataset used included only information meteorological data from surface levels, while fog events are also influenced by upper-levels circulation. Another feature that could bring some improvement would be sea surface temperature near the airports, specially for Montevideo which is closer to the ocean.

The bigger challenges faced in this study were the model architecture configuration, which demanded some manual attempts until a good configuration was reached. It is also noted the importance of a carefully data curation in order to fix for spurious and/or missing data.