Data Mining and Text Mining Project Report

920715 Mattia Surricchio MATTIA.SURRICHIO@MAIL.POLIMI.IT

963503 Paolo Fornoni PAOLO.FORNONI@MAIL.POLIMI.IT

916702 Sergio Placanica SERGIOPIERMARIO.PLACANICA@MAIL.POLIMI.IT

1. Introduction

The goal of the project is the predictive maintenance of air conditioning systems in electrical control units, operating next to telecommunications antennas. The provided dataset is composed by 2605 sites (control units), split in training set (2071 sites) and test set (534 sites) in a time frame spanning from April 2019 to Feb 2020 (10 months).

2. Data exploration

2.1. Target variable distribution

We started by assessing the balance of the target class in the dataset. The kind of problem we are tackling suggest the dataset will be imbalanced, and indeed the percentange of negative example surpass by far the positive one.

Table 1: Target class distribution

Class	Frequency	Percentage
Positive	3583	0.58%
Negative	617717	99.42%

We also explored the incidence of fault in AC equipment over the time frame and noted that is was mostly concentrated in the period between April and September with his peak in August. This intuitively suggest a correlation between fault and high temperature and also hint at the necessity to generate time features to capture this pattern.

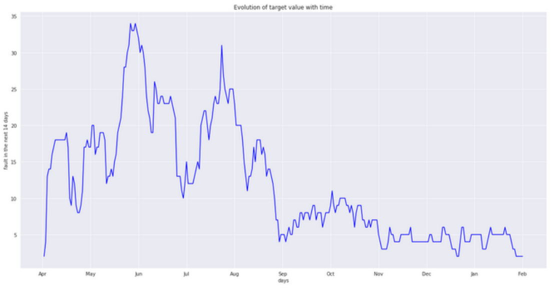

Figure 1: Fault occurence over time

2.2. Missing values

We found out that 15 sites don't belong to any CELL_TYPE_X. No other missing values were found.

2.3. Categorical Features

The categorical features of the dataset are : CELL_TYPE_X and GEOGRAPHICAL_CLUSTER_K_x, both one-hot encoded. We noted that CELL_TYPE_X has an unbalanced distribution.

2.4. Numerical Features

2.4.1. Correlation

The dataset contains a large amount of numerical features regarding weather condition in the past and weather conditions forecast. Our first idea was to establish the correlation between these features to attempt PCA dimensionality reduction, but we preferred to completely remove those variables. The results show that features relative to the same weather property in general are highly correlated.

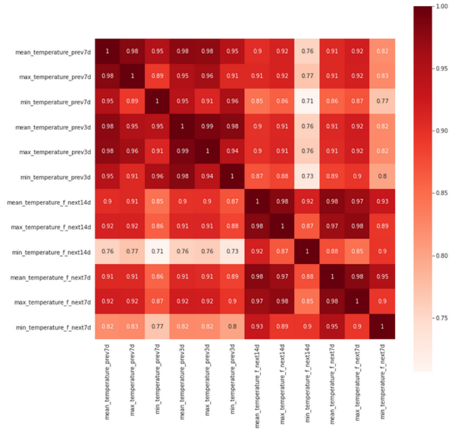

Figure 2: Correlation for temperatures

2.4.2. Target against Numerical Features

Given the high amount of features, we wanted a fast and intuitive way to understand which features could be deemed important to our classification task. We plotted the density of the features for the positive and negative classes. The result show that almost all features density distribution overlaps. At first sight, there's no clear "magic feature".

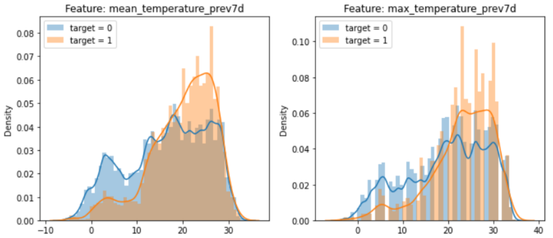

Figure 3: mean and max temperature density over target classes

2.4.3. Redundancy

We noticed that previous 7 days features and the forecasted ones have the exact same values, they are just shifted of 8 days one each other.

3. Data Engineering

3.1. Time Encoding

Given the 10 months period and considered the observations at 2.1, we applied the following preprocessing step on DATE feature:

- took apart the `DATE` in `month` and `day` discarding the year.
- encoded `month` as categorical feature from 0 to 9.
- encoded `day` as 2 numerical features `day_sin` and `day_cos`.

3.2. Numerical Feature

By plotting 3 days long features vs 7 days long features, we noticed that the latter were less noisy and had a smoother behaviour.

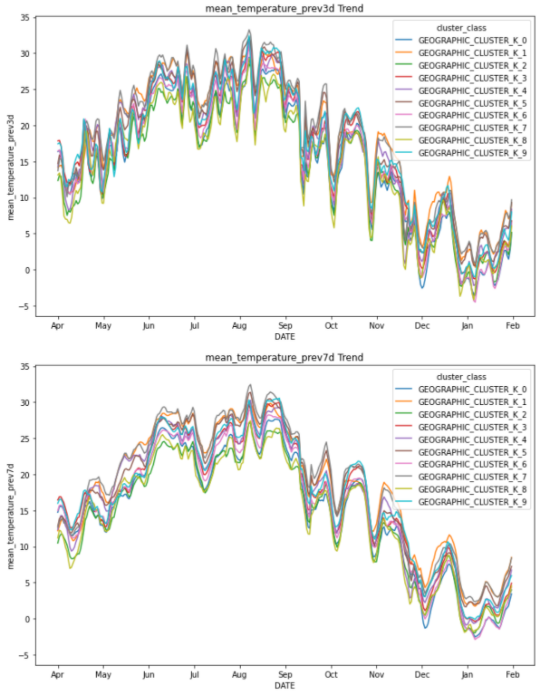

Figure 4: Comparing 3 days feature with 7 days feature

Furthermore, using different tree based algorithm, every model gave more importance to features belonging to a bigger time frame. Given the previous results and the fact that our target has a 14 days time frame, we artificially generated a new set of features by averaging two subsequent weeks, obtaining 14 days long features. We discarded all the variables having a time frame of 3 and 7 days. We 5 fold cross-validated a Random Forest model using the full set of features and the engineered one (14 days only), then compared the results using precision and recall as metrics.

We used a t-test with 99% confidence level and the results were statistically significant.

3.3. Dropped Features

we dropped the following features:

Precision	Recall	Precision_14	Recall_14
78.70%	23.74%	88.45%	47.06%
72.02%	19.41%	85.58%	39.80%
73.09%	20.08%	85%	42.67%
73.97%	22.59%	86.24%	41.98%
77.98%	23.70%	88.03%	43.09%

Table 2: 5-fold cross-validation performances

- `CELL_TYPE_X ->` not significant
- `aircon_sum_prev_x ->` data leakage of the target, by shifting this variable back of fourteen days you get the exact distribution of the target.
- `SITE_ID ->` could actually be detrimental causing the learner to recognize specific sites instead of generalizing.
- `Forecasted Features ->` Since they are identical to the past ones, we decided to drop them
- `Correlated Features ->` We drop the remaining weather and alarm features with a correlation higher than 90%

4. Proposed models

Given the large numbers of features, we decided to focus our work on trees ensemble methods. In particular we used Random Forest, for features evaluation and fast prototyping, and XGBoost for our final model.
We tested our model on a hold-out set removing the 20% of the SIDE_IDS.

4.1. Base model

Our baseline model is a simple Decision Tree. We decided to use this model since it is the easiest model capable of managing the high number of features in our dataset. Models like logistic regression had convergence problems due to the high dimensionality of the problem.

Table 3: Baseline test performances

Precision	Recall	F1-score
3.41%	4.87%	4.01%

4.2. XGBoost

We tuned a XGBoost classifier using 5 fold Stratified cross-validation to keep the class balanced across different folds. We used *scale post weight* to balance the learning of the unbalanced targets.

Metric	Cross-Val
Precision	82.57%
Recall	86.27%
F1-Score	84.38%
AUC Prec-Rec	89.15%

Table 4: 5-fold stratified cross-validation

Metric	Test
Precision	3.86%
Recall	13.96%
F1-Score	6.05%
AUC Prec-Rec	1.65%
Weighted Recall	10.27%

Table 5: SITE ID hold out

4.3. Brute Force Model

We also tested a non-ML model which computes the predictions by just shifting and binarizing the aircon_sum_wo_prev14d feature. The results on the test set were pretty high.

Table 6: Brute force model test performances

Precision	Recall
97.16%	98.35%

5. Conclusions

Our model performs well if using a random hold-out criteria, but it is not capable of generalizing over unseen sets of SITES. The available data might not be suitable for achieving the requested task:

- distribution of features are almost perfectly overlapping for the two target variables
- target variables are heavily unbalanced
- there is a leakage between target variable and `aircon_sum_prev_x` feature

Given the previous results, we suggest to change the collected data or the evaluation criteria.
If the goal is the predictive maintenance of new sites non available in the training set, we suggest to collect technical data of the air conditioner via on-board sensors and then perform anomaly detection on its performances.
On the other hand, if the goal is the predictive maintenance of a known number of sites in a future unseen time-frame, we suggest a temporal hold-out for testing, as reported in Machine Learning Predictive Maintenance on Data in the Wild