Student: Vincenzo Cappalonga

Matr: 201458

Presentation of the Problem

The aim of the thesis is to create a calendar for predictive maintenance through a predictive model capable of accurately anticipating failures. The data frames I used to create the data warehouse represent the intrinsic characteristics of the machines, such as the type of model or age, and processing characteristics, i.e. the number of failures, errors and maintenance performed on the machines. The main aspect of the dataframes is the precision with which the data has been organized: in fact, they have been collected by sensors that round the processing parameters to the exact hour in order to have a complete, hour-by-hour, and detailed overview. Specifically, the dataframes are:

- errors, five types of errors are reported, which do not interrupt the processing of the machines.

- failures, failures of the four main components are reported, all present in all the machines. These types of failures stop the processing of the machines.

- maint, maintenance of the individual components of all the machines is reported. Maintenance occurs both when failures occur and in advance to prevent failures from occurring.

- machines, all information relating to the model and age of the machines is reported.

- telemetry, the processing parameters of the machines are reported, hour by hour, relating to vibrations, pressure, energy consumption and rotation.

**Data Aggregation**

The first step is to aggregate the data into a single dataset to get all the necessary information, but I thought of aggregating the dataframes in two different ways to get different but crucial information for data analysis. I then created two different datasets: "df\_unito" and "tabella\_tot".

The "df\_unito" dataframe was created to answer the main question of the problem: to create a model capable of predicting failures, I have to predict the maximum time of use of a component before it fails. Before merging the dataframes together, the columns containing the failures, errors and maintenances were transformed into columns for the single errors, single maintenances and failures for each component. Then by merging the dataframes with telemetry and eliminating the duplicates I obtained a dataset that, hour by hour, reports for each component and for each machine if there has been a failure or an error, with a 1 in the specific columns if in that hour there has been one of the previous significant events. Having the exact time of the failure, I was able to accurately calculate the hours between one failure and another, for each component, between two errors of the same type and two maintenance operations.

In the "tabella\_tot" dataset I focused on the analysis of the total number of failures, errors and maintenance for each machine, making the difference between maintenance for failures and preventive maintenance. Adding to this dataset the maximum times between each event (failures, errors and maintenance) we obtain the datawarehouse on which we will train and test the model. And finally, attributes were calculated on the processing parameters of the machines. These parameters, not having any information about them, were interpreted as positive or negative components based on my assumption that the machines in question were milling machines or lathes, since the four processing parameters reflect the characteristics of those machines. And they are:

- "efficiency\_rotate", the rotation with which the tool is made to work determines the efficiency of the processing, the higher it is, the more the machine will work without difficulty;

- "volt\_consume", the energy consumption increases when the machine has difficulty in processing;

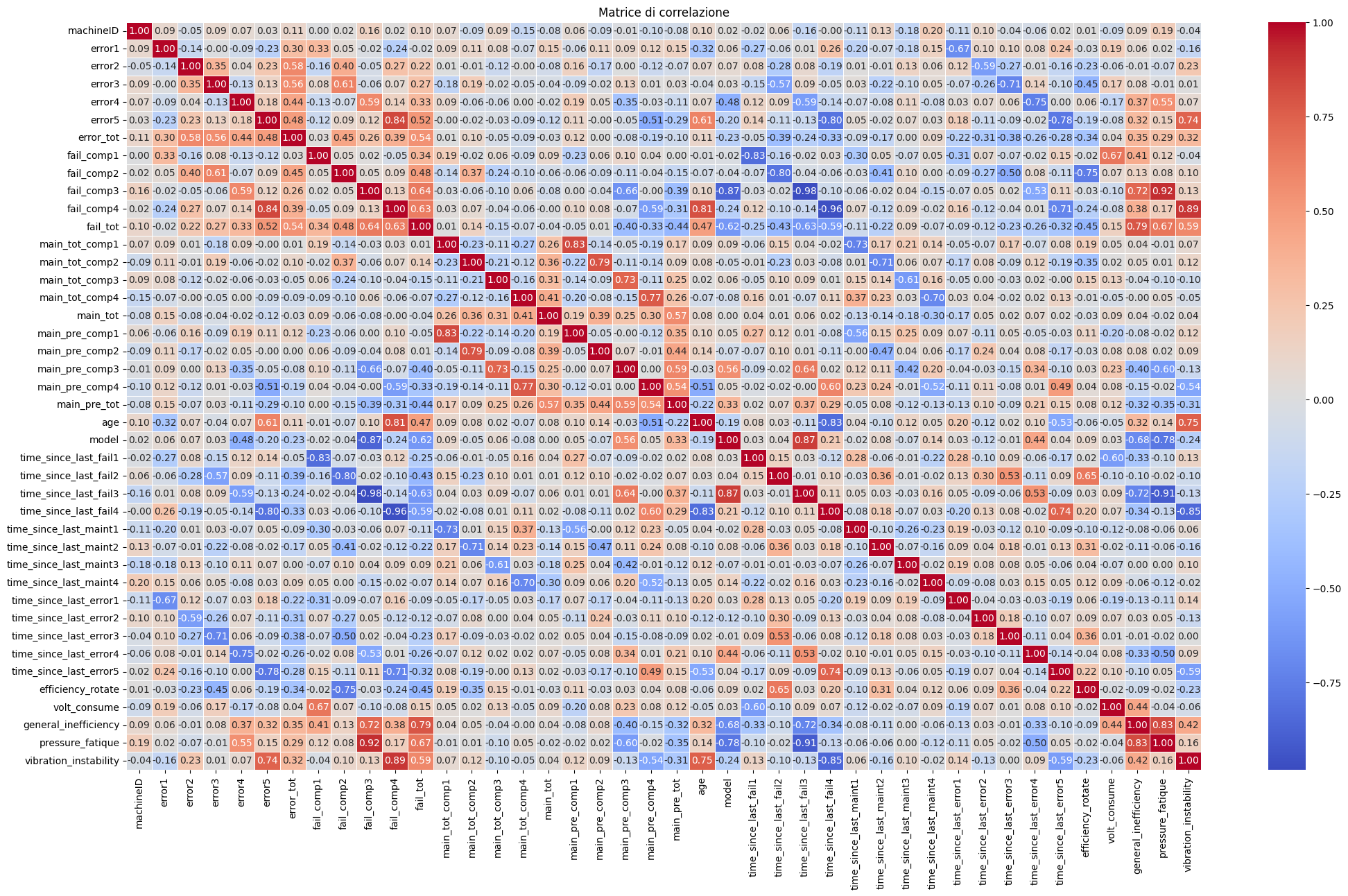
- "general\_inefficiency", the ratio between pressure, vibration and consumption and their maximums represents the inefficiency of the machines since these are parameters that represent difficulties in processing;

- "pressure\_fatique", the more resistant the material will be to processing, the more the tool will have difficulty;

- "vibration\_instability", the vibration of the entire machine and the processing platform.

**Correlation Matrix**

I thought that, once the dataset with the complete data was created, it was important to understand the correlation between the various calculated features, so as to create more performing models and have more information on failures and all other events (maintenance and errors). I therefore built the following correlation matrix:



We immediately notice how the number of failures of the four components is correlated with the previously calculated processing parameters:

- component no. 1 has a number of failures that increases with energy consumption (0.67), as well as slightly with general inefficiency (0.41). It is probably connected to the electronic resistances of the machine or to the digital controls;

- component no. 2 is correlated with the efficiency of the rotation (-0.75), so if the number of failures increases the efficiency decreases. It is probably very close to the tool used in the processing or to the devices connected to it;

- component no. 3 is correlated with pressure (0.92), the more the failures to it increase and the more, probably, the resistance of the material to processing would increase;

- component no. 4 is correlated with instability due to vibrations (0.89), the more the failures to the component increase and the more unstable the machine is.

The first analysis to be done is the influence of age and model on machine failures. They are in fact reported in the following graphs:

Immagine che contiene testo, schermata, linea, Diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, schermata, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

It is noted that the type of model affects the overall number of failures. Instead, the age/number of errors per component graph shows us that:

- component number 4 fails mainly in machines aged 14 years and older and with high frequency;

- the number of failures increases with age;

- component number 3 does not fail for some machines.

The last graph should be viewed in contrast with the age/number of preventive maintenance graph per component:

Immagine che contiene testo, schermata, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

This shows us how in every machine there are all the components and on those in which less preventive maintenance is done there are more errors.

Predictive Model

Now that we have all the necessary information we can create a predictive model, based on the maximum time between two failures of the same component, for each component, because depending on which one I want to predict the failure I add different attributes to optimize the model. For each component I created four models with different performances to create the most accurate solution possible.

**Component n°1**

Let's start by understanding which attributes are most correlated with the time between one failure and another:

Immagine che contiene testo, schermata, Parallelo, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

I develop some of the following features for this component:

- the ratio between the number of preventive maintenance and the number of component tastes;

- the ratio between error 1 and the number of failures and many others, always considering the most closely related.

The models used are regressor models of the type: Random Forest, Xgboost, Linear Regressor and KNN Regressor.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Comp.1 | MAE | MAPE | MSE | R2 |
| RF | 37.45 | 0.19 | 52311.42 | 0.72 |
| XGB | 31.68 | 0.16 | 38600 | 0.79 |
| LR | 47.46 | 0.23 | 71938.19 | 0.62 |
| KNN | 68.65 | 0.29 | 171594.8 | 0.08 |

Here are the graphs representing the model predictions and the actual values:

Immagine che contiene testo, linea, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, linea, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

Immagine che contiene testo, linea, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, linea, diagramma, Diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

**Component n°2**

Let's start by understanding which attributes are most correlated with the time between one failure and another:

Immagine che contiene testo, schermata, linea, Parallelo

Il contenuto generato dall'IA potrebbe non essere corretto.

I develop some of the following features for this component:

- the ratio between the volt consume and the number of component tastes;

- the ratio between error 3 and the number of failures and many others, always considering the most closely related.

The models used are regressor models of the type: Random Forest, Xgboost, Linear Regressor and KNN Regressor.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Comp.2 | MAE | MAPE | MSE | R2 |
| RF | 39.01 | 0.23 | 51366.54 | 0.65 |
| XGB | 37.19 | 0.22 | 46022.11 | 0.69 |
| LR | 43.85 | 0.23 | 86223.15 | 0.41 |
| KNN | 52.94 | 0.28 | 133500 | 0.09 |

Here are the graphs representing the model predictions and the actual values:

Immagine che contiene testo, linea, diagramma, Diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, linea, diagramma, Diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, linea, diagramma, Diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, diagramma, Diagramma, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

**Component n°3**

Let's start by understanding which attributes are most correlated with the time between one failure and another:

Immagine che contiene testo, schermata, Diagramma, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

I develop some of the following features for this component:

- the ratio between the general inefficiency and the number of component tastes;

- the ratio between error 4 and the number of failures and many others, always considering the most closely related.

The models used are regressor models of the type: Random Forest, Xgboost, Linear Regressor and KNN Regressor.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Comp.3 | MAE | MAPE | MSE | R2 |
| RF | 8.28 | 0.04 | 9363.91 | 0.96 |
| XGB | 7.01 | 0.04 | 5336.31 | 0.98 |
| LR | 21.25 | 0.09 | 19316.62 | 0.91 |
| KNN | 31.5 | 0.17 | 118379.8 | 0.46 |

Here are the graphs representing the model predictions and the actual values:

Immagine che contiene testo, linea, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, linea, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, linea, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, linea, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

**Component n°4**

Let's start by understanding which attributes are most correlated with the time between one failure and another:

Immagine che contiene testo, schermata, linea, Parallelo

Il contenuto generato dall'IA potrebbe non essere corretto.

I develop some of the following features for this component:

- the ratio between vibration instability and the number of component tastes;

- the ratio between error 5 and the number of failures and many others and with age, always considering the most closely related.

The models used are regressor models of the type: Random Forest, Xgboost, Linear Regressor and KNN Regressor.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Comp.4 | MAE | MAPE | MSE | R2 |
| RF | 13.98 | 0.1 | 15919.39 | 0.95 |
| XGB | 16.72 | 0.11 | 16907.57 | 0.94 |
| LR | 24.9 | 0.15 | 29778.62 | 0.90 |
| KNN | 19.18 | 0.13 | 29591 | 0.90 |

Here are the graphs representing the model predictions and the actual values:

Immagine che contiene testo, linea, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, linea, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, linea, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, diagramma, linea, Diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

**Model Used**

**Random Forest Regressor**

Random Forest (RF) is an ensemble learning method that operates by constructing multiple decision trees during training and averaging their predictions to improve accuracy and reduce overfitting. Each tree is built on a random subset of the training data, and at each split, a random subset of features is considered, ensuring diversity among trees. This randomness reduces variance while maintaining a low bias, making RF a robust model for regression tasks. The final prediction is obtained by averaging the individual tree predictions.

**Random Forest with Param Grid and GridSearchCV**

To optimize the hyperparameters of Random Forest, a parameter grid is defined, specifying different values for critical parameters such as:

* n\_estimators: Number of trees in the forest.
* max\_depth: Maximum depth of each tree.
* min\_samples\_split: Minimum number of samples required to split a node.
* min\_samples\_leaf: Minimum number of samples required to be a leaf node.

GridSearchCV is then employed to perform an exhaustive search over all possible parameter combinations using cross-validation. This ensures that the selected hyperparameters generalize well to unseen data.

**XGBoost Regressor**

XGBoost (Extreme Gradient Boosting) is a powerful gradient boosting algorithm that optimizes the standard Gradient Boosting framework by incorporating additional regularization techniques, parallelization, and a more efficient handling of missing values. Unlike Random Forest, which builds independent trees in parallel, XGBoost constructs trees sequentially, where each tree corrects the errors of the previous one. The optimization is performed through a second-order Taylor expansion, allowing faster convergence and higher accuracy.

**XGBoost with Param Grid**

To enhance model performance, a parameter grid is defined to tune key hyperparameters, including:

* n\_estimators: Number of boosting iterations.
* learning\_rate: Step size shrinkage to prevent overfitting.
* max\_depth: Maximum depth of each tree, controlling model complexity.
* min\_child\_weight: Minimum sum of instance weight needed to split a node, preventing overfitting.
* subsample: Fraction of observations used for training each tree, improving generalization.
* colsample\_bytree: Fraction of features used per tree, reducing correlation among trees.
* reg\_alpha, reg\_lambda: L1 and L2 regularization terms, respectively, to penalize complex models.

By systematically evaluating multiple combinations of these hyperparameters, the model can achieve an optimal trade-off between bias and variance, ensuring high predictive performance.

**Linear Regression**

Linear Regression (LR) is a fundamental regression technique that models the relationship between a dependent variable and one or more independent variables by fitting a linear equation to the observed data. LR is highly interpretable but may struggle with complex, non-linear relationships.

**K-Nearest Neighbors Regressor (KNN)**

KNN Regressor is a non-parametric, instance-based learning algorithm that predicts the target value of a new data point by averaging the target values of its **K nearest neighbors** in the feature space. The similarity between data points is commonly measured using **Euclidean distance**. KNN is highly flexible and adapts well to non-linear patterns but can be computationally expensive for large datasets, as it requires storing and searching through the entire dataset during inference. The key hyperparameter is K, the number of neighbors, which controls the bias-variance trade-off:

* A **small K** (e.g., 3) results in low bias but high variance.
* A **large K (e.g., 20**) increases bias but reduces variance.

**Possible Improvements and Refinements:**

1) Having more information about the machines would help to identify the most important characteristics and to measure and analyze them correctly, such as: more info on the output of the machines or on the cost of maintenance.

2) Analysis on the efficiency curve - Times between events.