**Data Preparation/Feature Engineering**

1. **Overview**

The dataset chosen for this machine learning exercise has a synthetic nature and contains results from simulations of grid stability for a reference 4-node star network. The original dataset contains 10,000 observations. As the reference grid is symetric, the dataset can be augmented in 3! (3 factorial) times, or 6 times, representing a permutation of the three consumers occupying three consumer nodes. The augmented version has then **60,000 observations.** This dataset shall help to train our ML model to provide some accurate prediction about how stable we can set the smart grid.

1. **Data Collection**

Our dataset is based on the "**Electrical Grid Stability Simulated Dataset**", created by Vadim Arzamasov (Karlsruher Institut für Technologie, Karlsruhe, Germany) and donated to the **University of California (UCI) Machine Learning Repository** (link [here](https://archive.ics.uci.edu/ml/datasets/Electrical+Grid+Stability+Simulated+Data+)), where it is currently hosted.

Two primary references support this machine learning exercise and demand special mention:

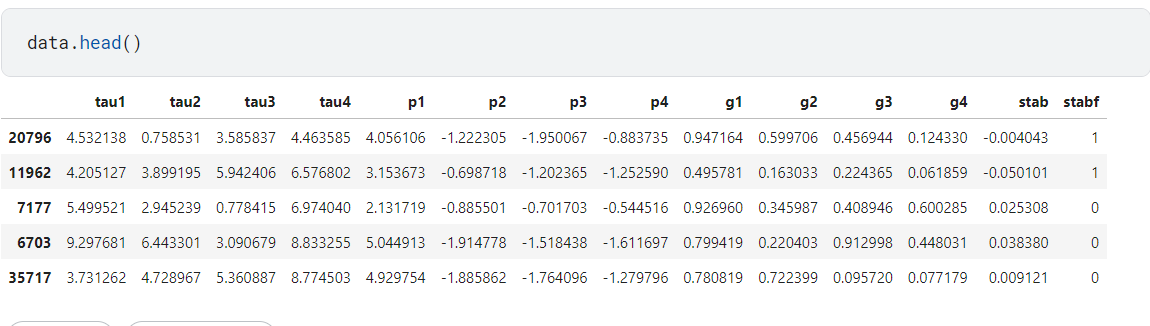
1. "***Taming instabilities in power grid networks by decentralized control***" (B. Schäfer, et al, The European Physical Journal, Special Topics, 2016, 225.3: 569-582), in which Dr. Schäfer (Network Dynamics, Max Planck Institute for Dynamics and Self-Organization - MPIDS, Göttingen, Germany) and his co-authors describe in detail the DSGC (Decentral Smart Grid Control) differential equation-based model to assess stability of smart grids;
2. "***Towards Concise Models of Grid Stability***" (V. Arzamasov, K. Böhm and P. Jochem, 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Aalborg, 2018, pp. 1-6), in which Dr. Arzamasov and his co-authors explore how data-mining techniques can address DSGC model simplifications.

The author is particularly thankful for Dr. Arzamasov's personal guidance and comments on the overall dataset structure.

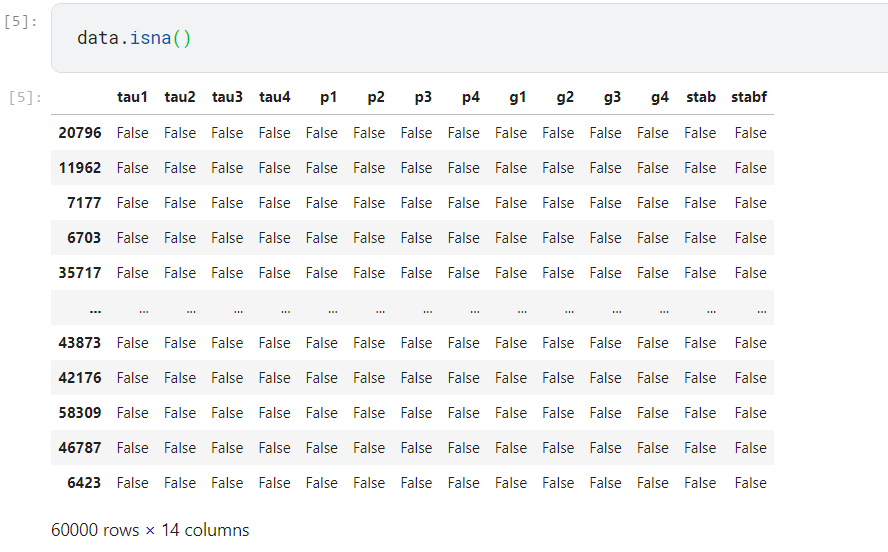
-We load our dataset with pandas

-We map our dataset and changed value of stable and unstable feature by binary entries 0,1.

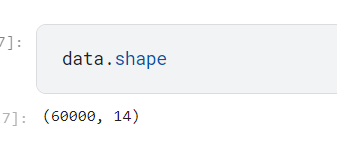
-obsvervation of the dataset rows by using head() function.

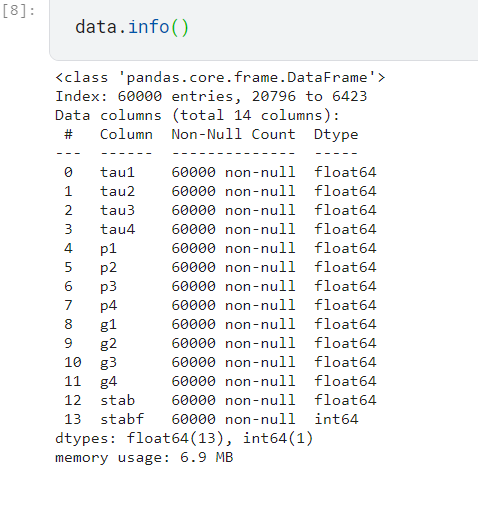


-Check if there is null values and shape

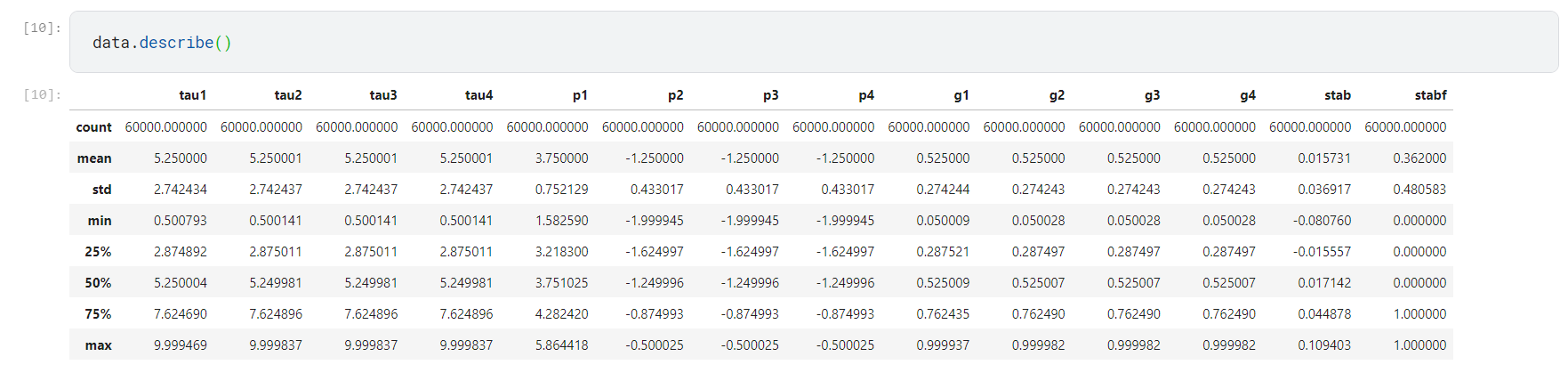


-Shape of our dataset





-Statistic references from dataSet

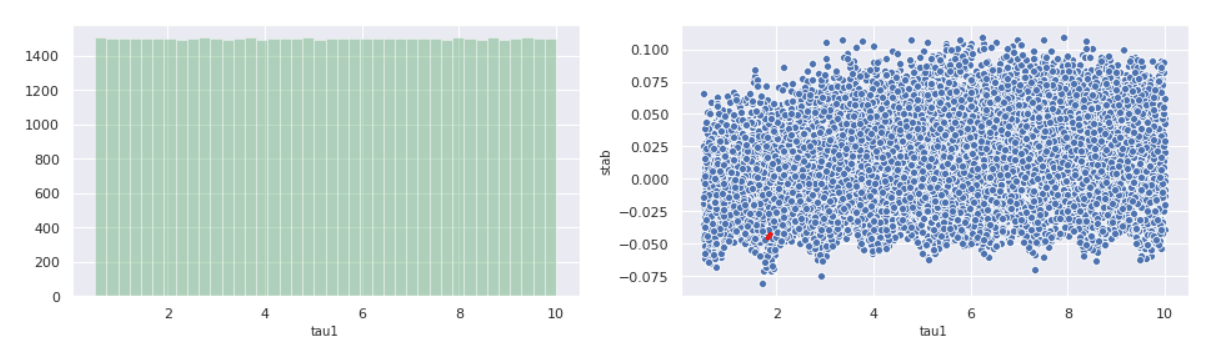


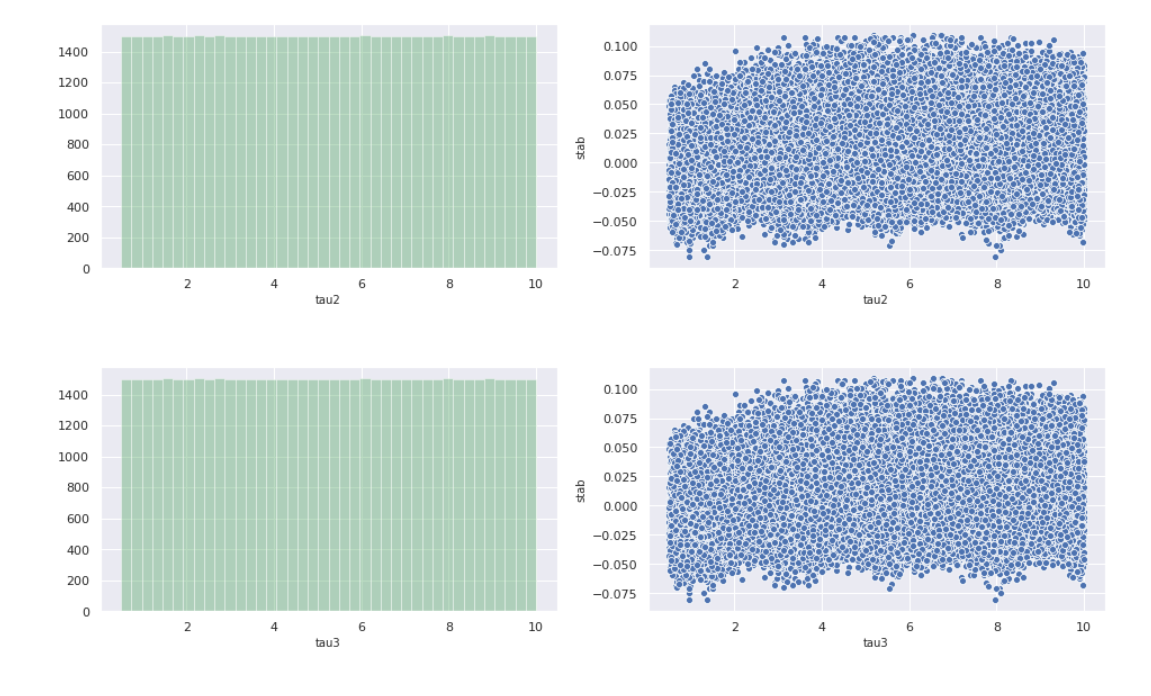
**3. Data Cleaning**

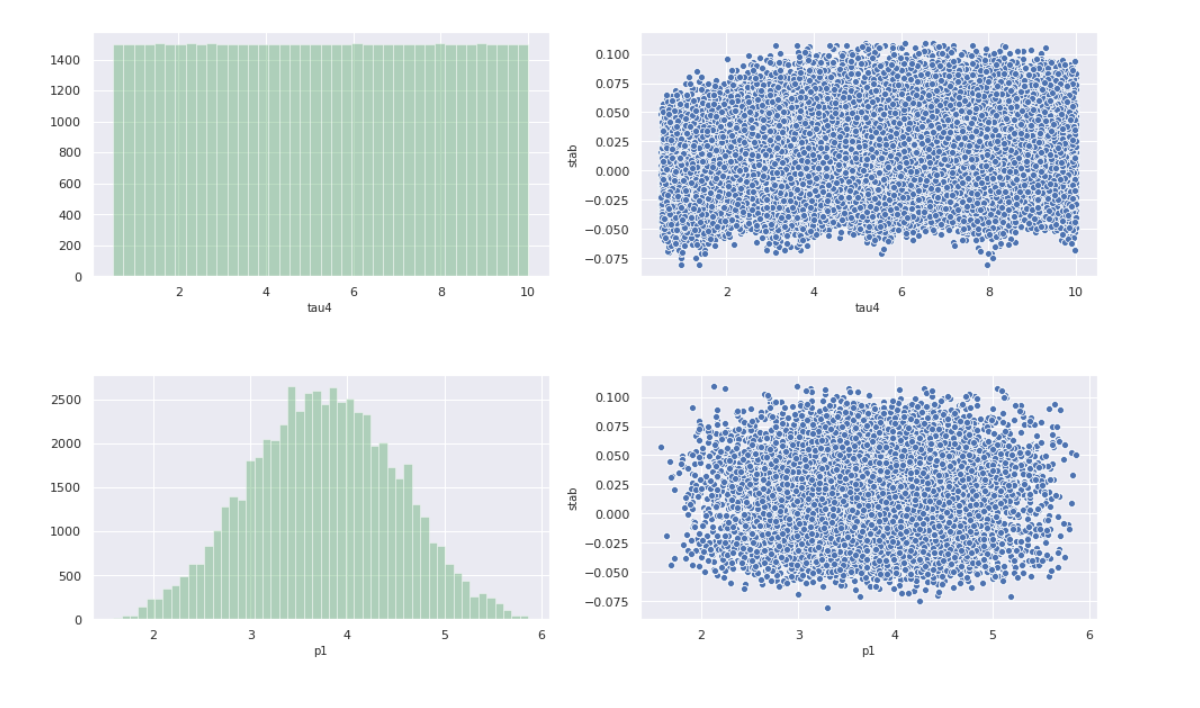
Our dataset don’t contain missings values but we did standard control and dropped if existing missing values.

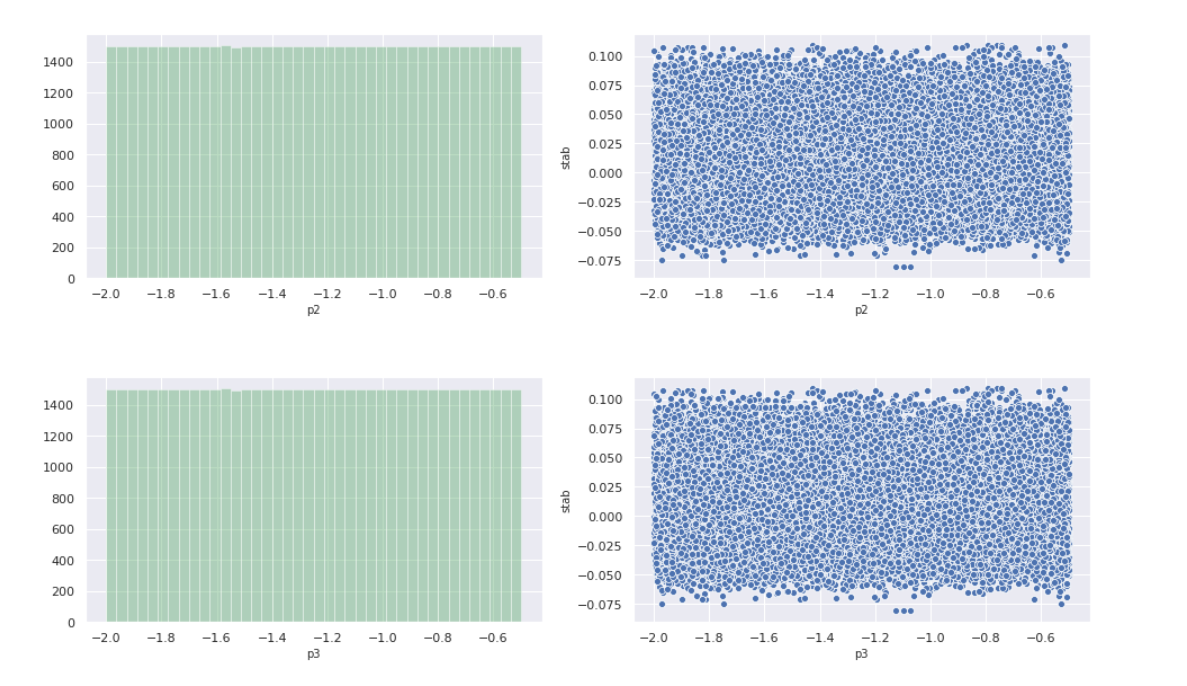
**4. Exploratory Data Analysis (EDA)**

Develops and displays a histogram and a scatter plot for a dependent / independent variable pair from a dataframe and, optionally, highlights a specific observation on the plot in a different color (red).

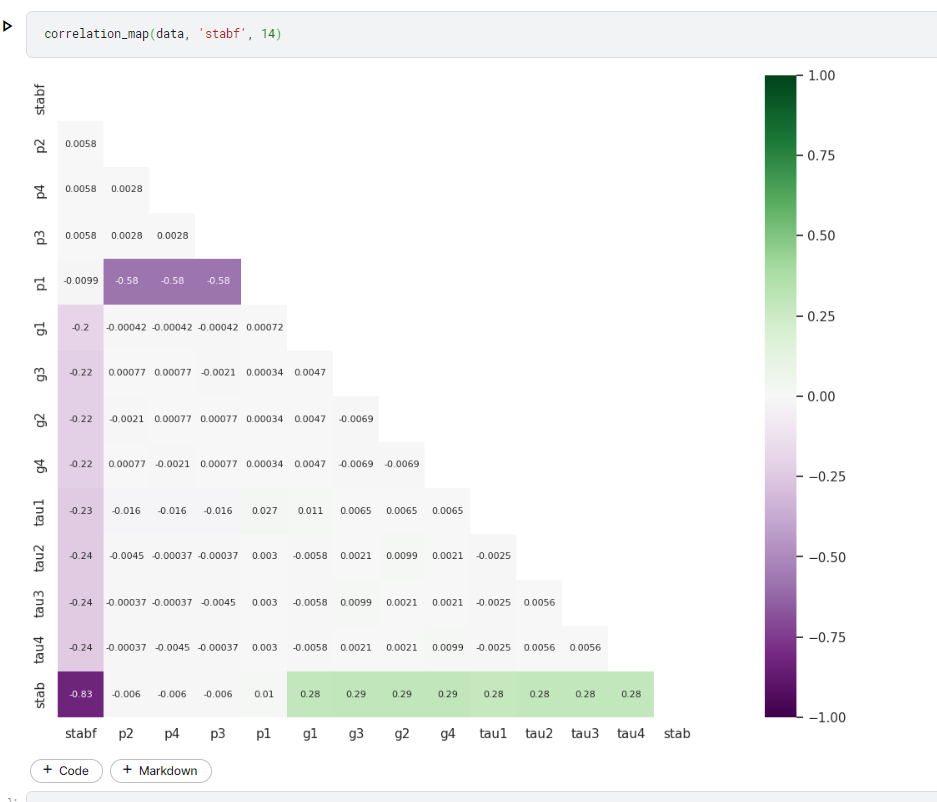








It is important to verify the correlation between each numerical feature and the dependent variable, as well as correlation among numerical features leading to potential undesired colinearity. The heatmap below provides an overview of correlation between the dependent variable ('stabf') and the 12 numerical features. Note that also the alternative dependent variable ('stab') has been included just to give an idea of how correlated it is with 'stabf'. Such correlation is significant (-0.83), as it should be, which reinforces the decision to drop it. Also, correlation between 'p1' and its components 'p2', 'p3' and 'p4' is above average, as expected, but not as high o justify any removal.



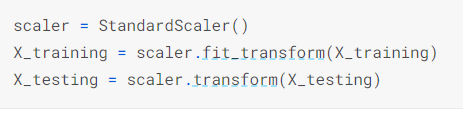


**5. Feature Engineering**

In our case, we did not create new features, we will just use these one for our prediction.

1. **Data Transformation**

Our data is very clean, no need of any scaling, normalizarion or encoding here. But we tried to do it as usual



**Model Exploration**

**1. Model Selection**



Our dataset has been divided in train and test set. We will use classification and regression model then evaluate their accuracy and R^2.





Classification Test Accuracy: 97.79%





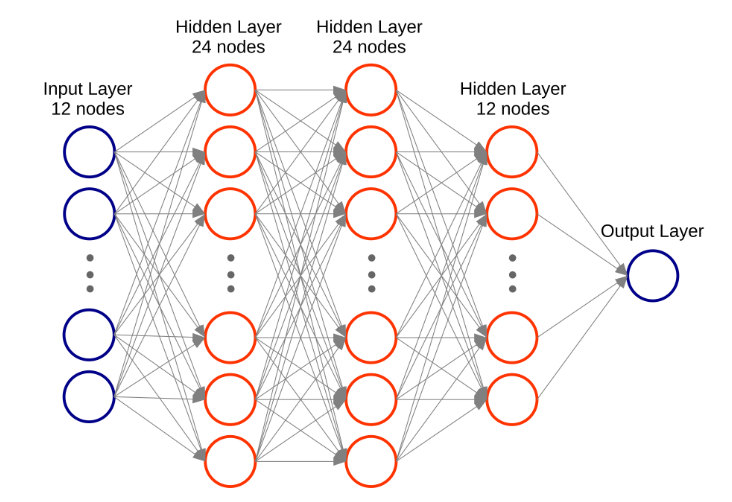
Regression Test R^2 Score: 0.95782

Our last model is deep learning model:

The artificial neural network (ANN) architecture depicted below is the optimal one evaluated in this study. It reflects an sequential structure with:

* one input layer (12 input nodes);
* three hidden layers (24, 24 and 12 nodes, respectively);
* one single-node output layer.

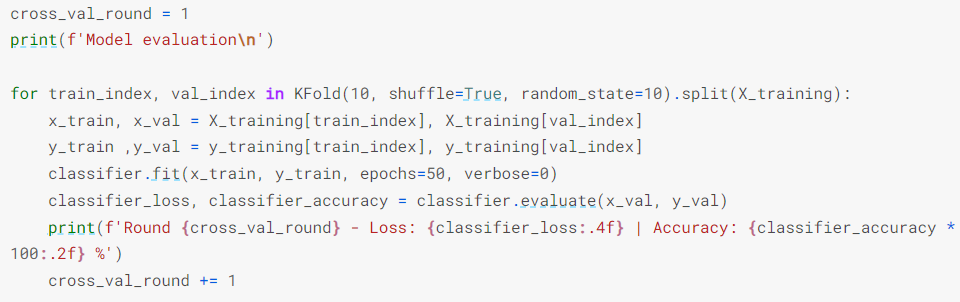
Alternative architectures were evaluated with variations of the code below.



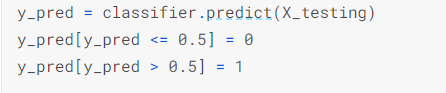


**2. Model Training**

Even considering that data is well behaved and in general uniformly distributed, a cross-validation based fitting is proposed. KFold is the cross-validation engine selected, and 10 different validation sets will be utilized.

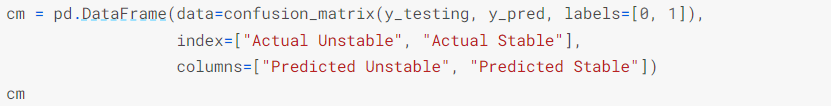


After fitting the model to the training set, it is time to extract predictions for the testing set and segregate those above the 'threshold' of 0.5 ('unstable' cases below the threshold, 'stable' cases above it).



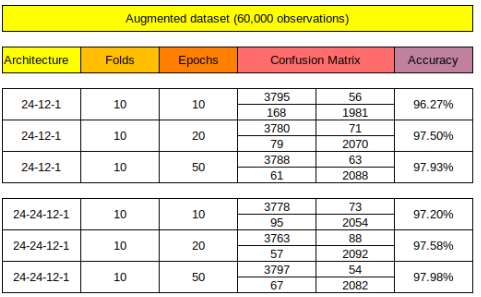
**3. Model Evaluation**

Confusion matrice





The architecture and the hyperparameters selected above led to the best prediction performance on the test set.



1. **Code Implementation**

We have implemented code for each question above.