Capstone project - Machine Learning Nanodegree

Manuel Maqueda Vinas, 15 August 2020

Definition

Project Overview

Pulmonary fibrosis, a disorder with no known cause and no known cure, created by scarring of the lung. Outcomes of the disease can vary from long-term stability to rapid deterioration, but doctors are not easily able to understand the exact outcome in advance.

In addition some of the current methods implies long times and substantial effort in order to produce an accurate prognoses. This increases patients anxiety and delays the application of the right mitigation action.

In this project I am creating a model which can help doctors and patients to understand in advance the evolution of the disease by predicting lung capacity of the patients in future weeks.

This project is based in the **OSIC pulmonary fibrosis progression** [Kaggle competition]

Problem Statement

The goal of the project is to predict patients' lung capacity based on a given train dataset. The main target is to predict weekly FVC measurements for 5 patients in the following 2-3 years.

In order to achieve the previous goal, this project is divided in the following sections:

- Exploration Data Analysis
  - Understanding both data flavours, tabular and images, in the given dataset
  - Exploring tabular data in details to have a good intuition of the information provided
- Data preparation
  - Extracting main features for the modelling part
  - Normalization of data features
  - Preparing the data for the training and test exercise
- Modelling and Training
  - Defining a custom neural network implemented in PyTorch
  - Creating a training algorithm using a quantile regression strategy
  - Showcasing the competition evaluation metric with a validation dataset
- Model deployment and inference
  - Upload the code in AWS SageMaker for training and model endpoint deployment
  - Performing inference using the test data in the competition
  - Understanding predictions

It is also worth mentioning that some other improvements are out of the scope of this project due to timing constraints. Those are described in the Future Work section at the end of this report.

Metrics

In order to create our model a couple of metrics has been used, *quantile loss* for the backpropagation step while training the PyTorch neural network and the other a modified version of the *Laplace Log Likelihood (LLL)* which is used by the Kaggle competition to evaluate the submissions. In our case the *LLL* has been implemented just on the validation dataset in other to illustrate its implementation and to enable further future work to control training epochs based on that metric.

QUANTILE LOSS

This project implementation relies in the quantile loss metric to train the custom neural network. There is more documentation available about this metric [here]

Below it is specified the loss function of Quantile Regression:

$$\rho_\tau(y) = y(\tau - \mathbb{I}_{(y<0)})$$

LAPLACE LOG LIKEHOOD (LLD)

The Kaggle competition is evaluated on a modified version of the Laplace Log Likelihood. In medical applications, it is useful to evaluate a model's confidence in its decisions. Accordingly, the metric is designed to reflect both the accuracy and certainty of each prediction.

Below it is described the formula used. Also from the point of view of this project this is implemented for the validation dataset.

$$\sigma_{clipped} = max(\sigma, 70),$$

$$\Delta = min(|FVC_{true} - FVC_{predicted}|, 1000),$$

$$metric = -\frac{\sqrt{2}\Delta}{\sigma_{clipped}} - \ln(\sqrt{2}\sigma_{clipped}).$$

Analysis

Data Exploration

In this part of the project the provided data has been analysed, focusing in the tabular data as mentioned above.

The first analysis performed is in the *train.csv* dataset. Below it is shown some of the characteristics derived from the analysis:

- Number of rows is 1549
- Number of unique patients is 176, which means that each patient has several rows corresponding to each of the individual week visits to measure the FVC.
- Smoking values are `Ex-smoker` `Never smoked` and `Currently smokes`
- Each of the rows contain the following data:
  - Weeks- the relative number of weeks pre/post the baseline CT (may be negative)
  - FVC - the recorded lung capacity in ml
  - Percent- a computed field which approximates the patient's FVC as a percent of the typical FVC for a person of similar characteristics
  - Age

- Sex (Gender)
- SmokingStatus

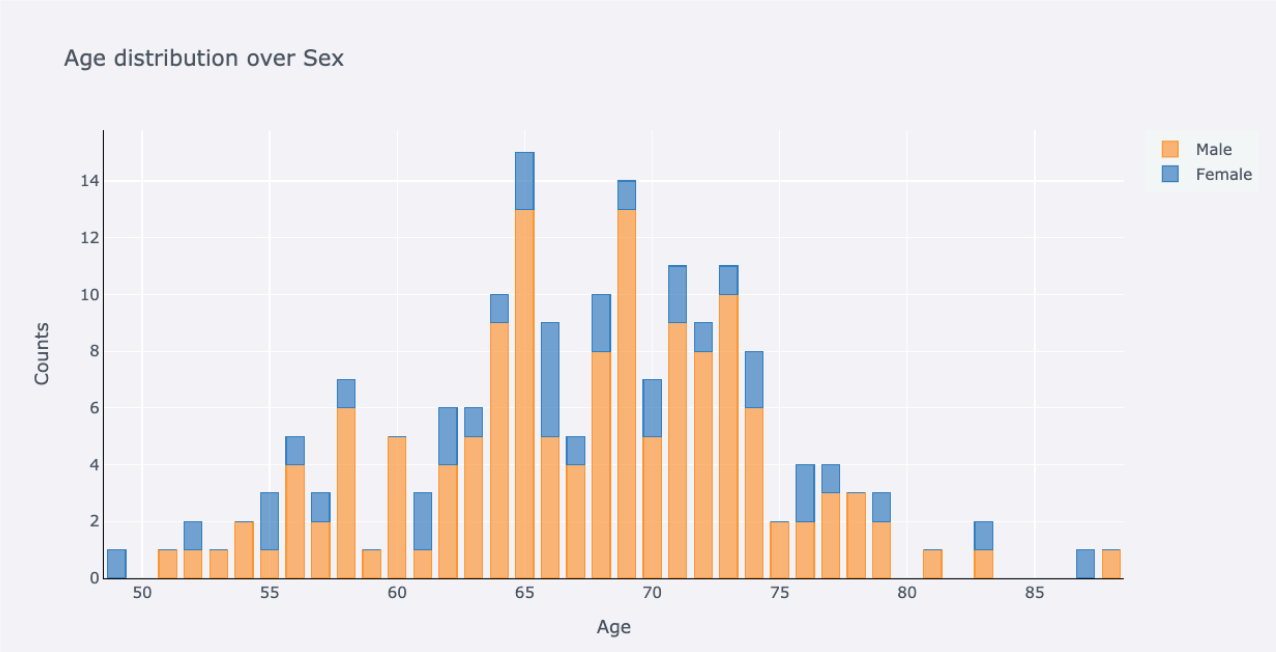
In terms of the imaging data the following has been verified in order to support any future work:
- Every patient in the tabular data has a directory containing the images of the baseline CT scan.

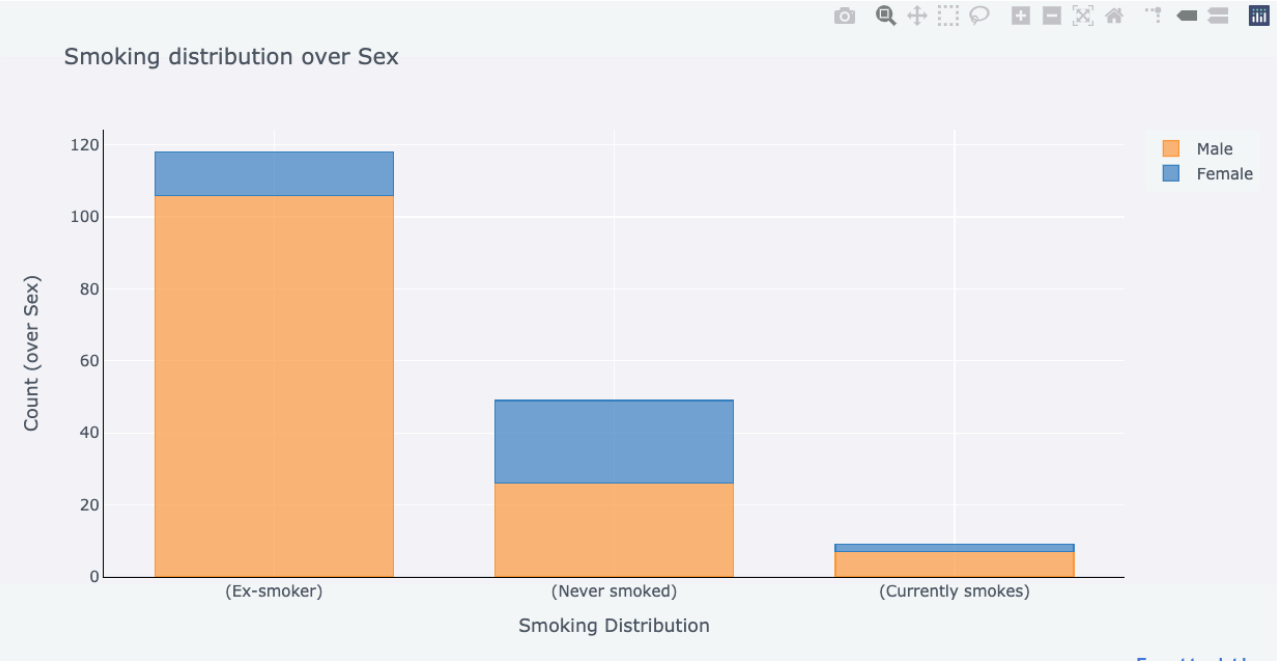
Data Visualisation

In this section an inspection of the tabular data has been performed in order to obtain more details of the data distribution per each of the available features: FVC, percent, age, sex, smoking status and weeks.

AGE AND SMOKING DISTRIBUTION OVER SEX

As it is shown in the figure from below, the majority of the patients in the training dataset are males, and also they are aged between 60 and 75 years old. In addition we can see that more than the 50% of the patients are also *`Ex-smokers`*

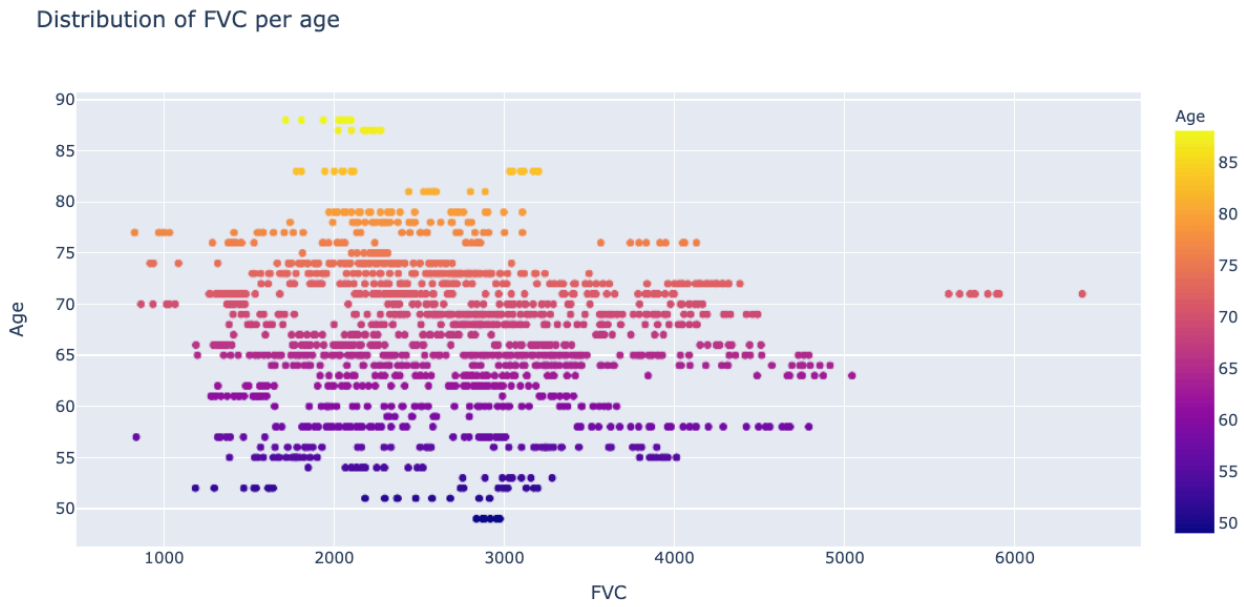


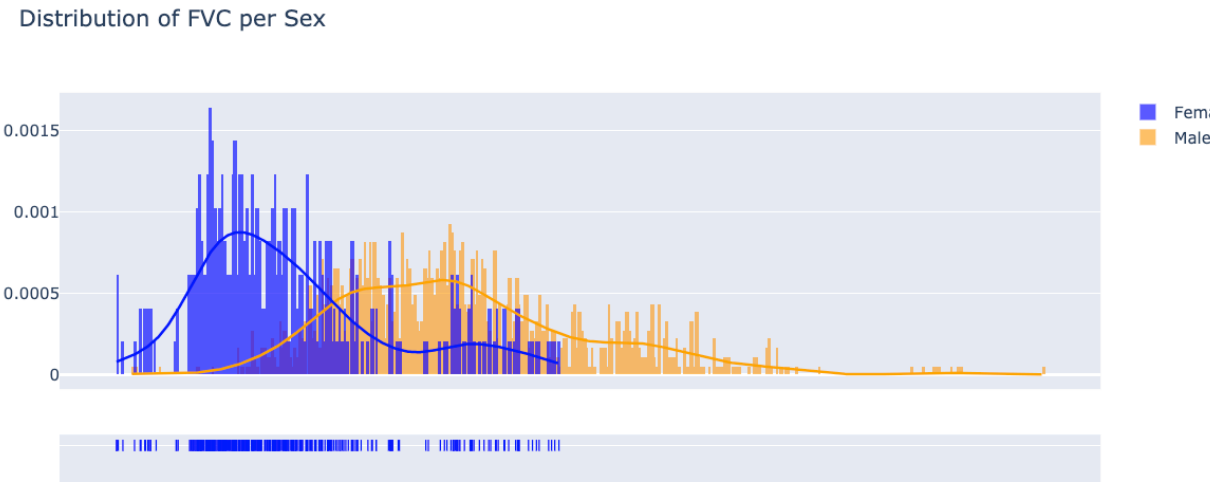


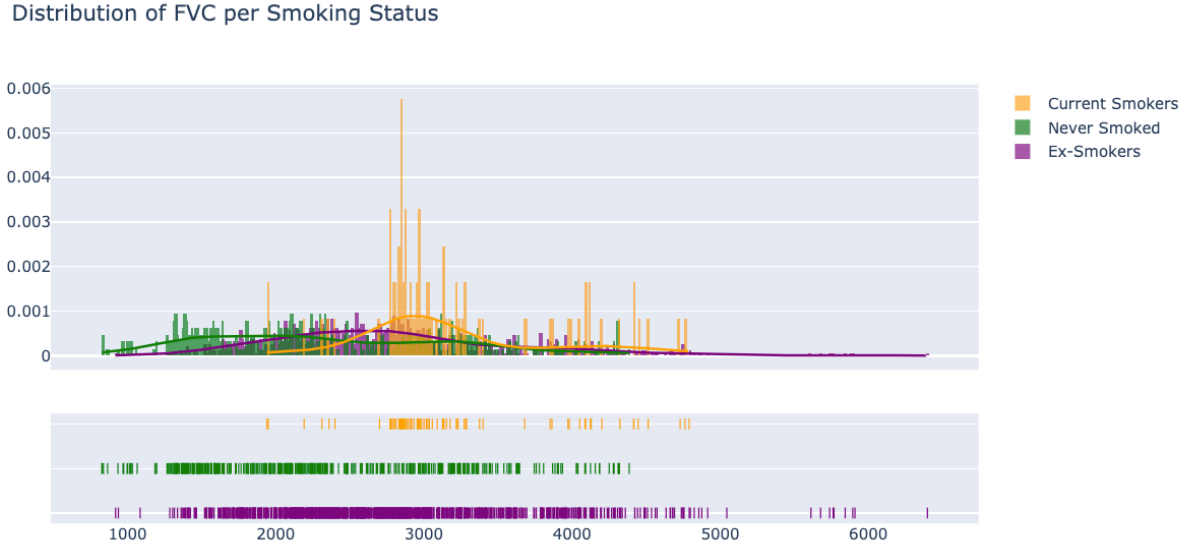
FVC (LUNG CAPACITY) PER AGE, SEX AND SMOKING STATUS

When looking at the FVC distribution over Age, Sex and Smoking Status, it is observed that in overall older people tend to have less lung capacity (FVC) and that average lung capacity is substantially lower for females than for males. These results are not a surprise and it might be more linked to the biological capacity of females and aged people rather than the actual incidence of the disease and the potential evolution of itself.

Another conclusion is that it is not obvious that ex-smokers or current smokers have less lung capacity than people that never smoked. In order to understand that in more detail this data has been segmented in the last part of this section to analyse Percent by Sex.

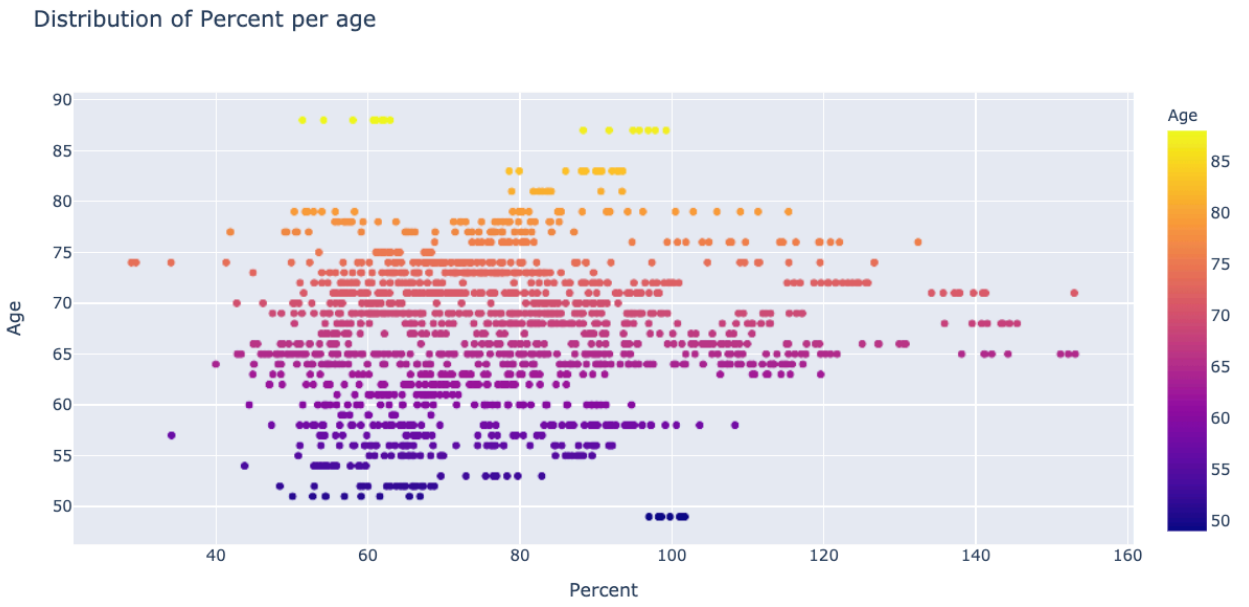


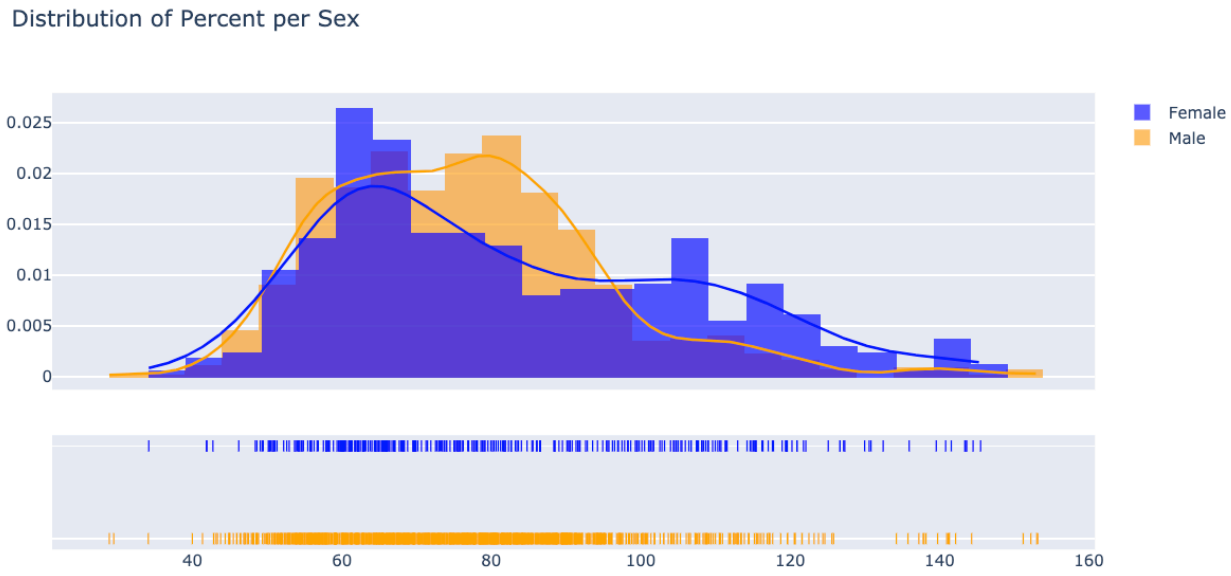


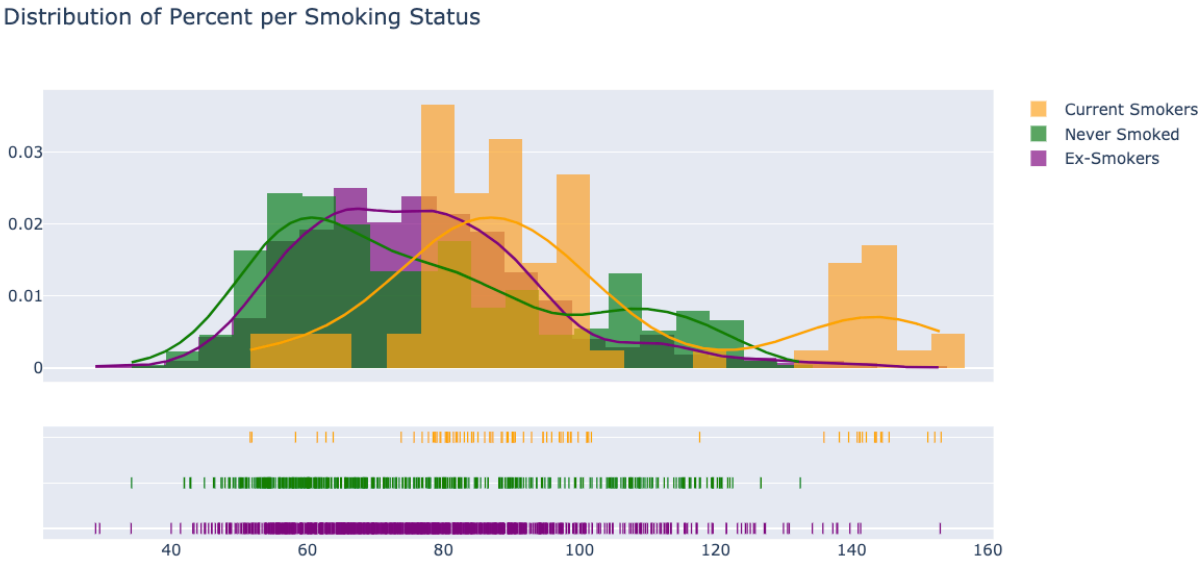


PERCENT (LUNG CAPACITY) PER AGE, SEX AND SMOKING STATUS
Looking at the distribution of percent against the rest of the properties seem to be the most relevant to actually understand the distribution of the lung capacity of the patients in the train dataset.

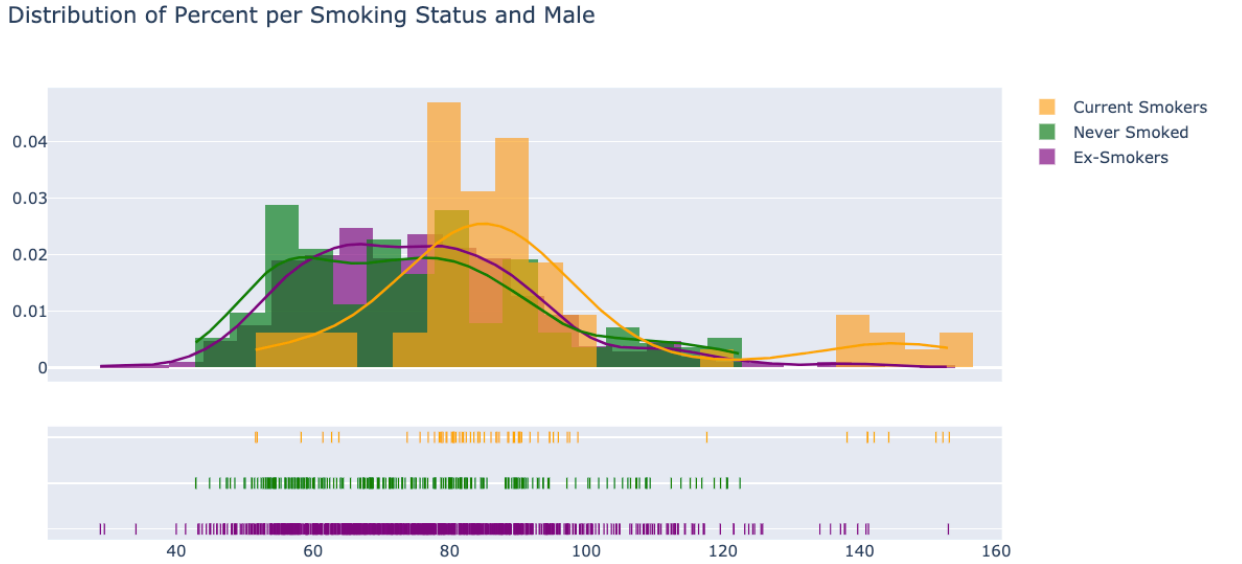
As part of the conclusions looking at percent distributions, it is observed that in overall younger patients have less lung capacity of what it is expected for them. Also the capacity is balance per sex, which indicates that there is no evidence that a particular sex is more prone than the other to have a more rapid deterioration. In addition, it is observed again that there is not evidence that `*ex-smoker*` or `*smoker*` persons have less lung capacity than `*never smoked*` persons.

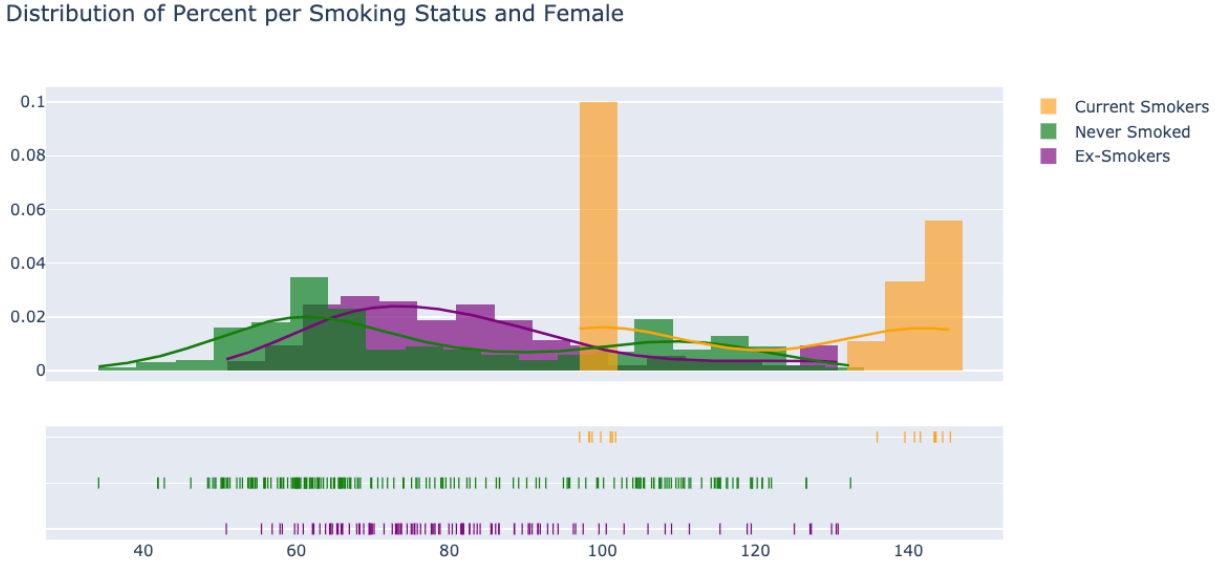




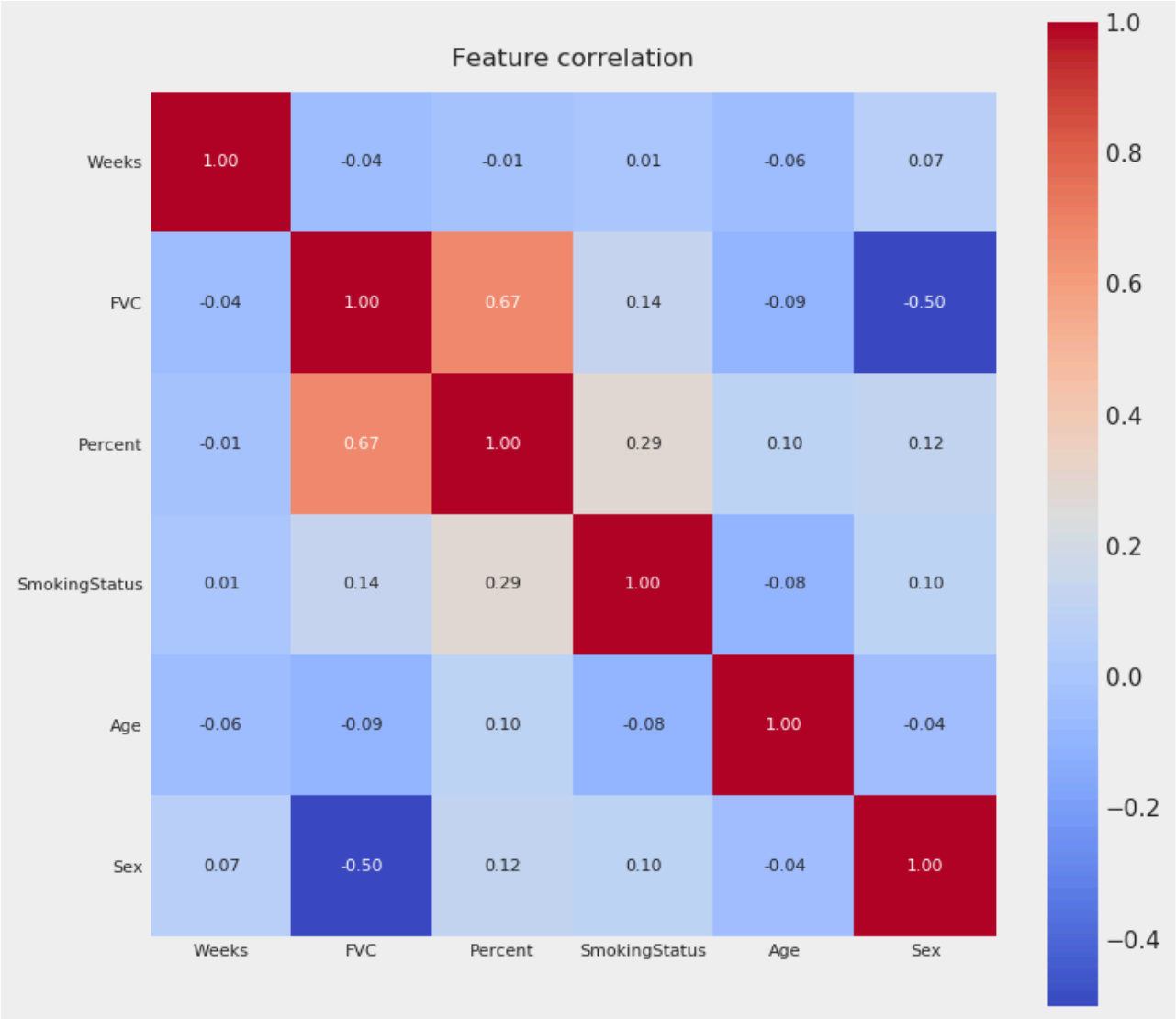


PERCENT (LUNG CAPACITY) AND SMOKING STATUS PER BOTH SEX: FEMALE AND MALE
In the following two graphs it is shown the distribution of percent per smoking status and particular gender. Some of the observations that are derived from both images are that we can confirm there is no evidence that smokers or ex-smokers have less lung capacity than never smoked person. However, it is worth mention that the evolution of the capacity might not follow the same trend. That is also mentioned as part of the future work.





FEATURE CORRELATION
As part of the analysis of the tabular data, a correlation matrix has been generated. From that matrix we can observer the low correlation between our features apart from FVC and Percentage. This correlation between percentage and FVC is expected as both are measuring lung capacity. It is also important to note that there is a bit of correlation between SmokingStatus and Percent which might indicate that **it might be** a relationship between both features over time, implying that there might be an impact on the SmokingStatus in the evolution of the disease.



Algorithms and Techniques

The first thing that done is to select the features to be used and preprocess it properly, including the normalisation of the input variables and preparing the dataset to be consumed by the training algorithm.

In order to obtain lung capacity prediction a custom neural network has been created using PyTorch. This is a 3-layered network which uses a quantile regression technique in order to adjust its weights as part of the training exercise.

In particular we have used the two metrics explained above, the first one quantile error is used for the back propagation of the neural network and the second one Laplace Log Likehood it has been just added to illustrate with a validation dataset how it is improved over the training exercise since this is the metric used by the Kaggle competition to evaluate the solution.

As part of the inference exercise the trained model provides three different values per each of the quantiles specified in training exercise. This is used to obtain the main prediction but also the confidence of the prediction which is just calculated as the difference between the inference of the higher quantile minus the inference corresponding to the lower quantile.

The model trained has been also tested as a model endpoint in the cloud using AWS Sagemaker to deploy it.

Benchmark

As described in the proposal of this project is benchmarked against a couple of notebooks of the Kaggle competition.

Methodology

Data Preprocessing

Before feeding the neural network with the input data, a pre-processing module has been included as part of this project. This module is in charged of splitting the train, validation and test datasets of the tabular data and of normalising the input features for the training and the inference exercise. For that in this project I have used the *preprocessing* library of *sklearn*.

At the end of the preprocessing function the train and test input datasets are prepared in the respective *csv* files, so those can be consumed when deploying in AWS Sagemaker.

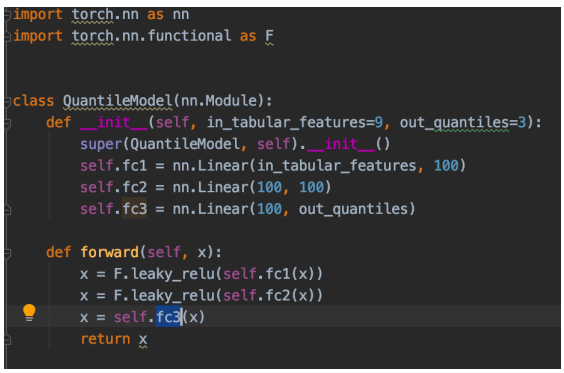
Implementation

This work has been implemented into two separate Jupyter Notebooks and a source code in python implementing the preprocessing logic, training and predict scripts and the neural network model using PyTorch.

- The **EDA-feature-extraction** notebook focuses in the analysis of the tabular data as explained above in this report. It also implements the preprocessing for the training and inference exercise. The function *preprocess_data* implements the logic to normalise the data and also generates the correspondent csv files to be used by the module *train.py* and *predict.py.* The generated files are:
  - pp_train.csv - the first column is the label (FVC) and the rest of the columns are normalised input data features
  - pp_test.csv - this file contains the input data features for every week of the 5 patients in the *test.csv* dataset.

The **source** folder contains various python modules:

- **preprocess.py -** this is a utility module to perform the preprocessing logic explained above.
- **model.py** - definition of the 3-layered neural network in Pytorch



- **train.py** - contains the code for the training of the neural network. The logic in this module is the following:
  - Parses the input hyper-parameters as the function can be called from a PyTorch estimator in SageMaker or can run locally if the input hyper-parameters are correctly passed to the script
  - Splits the train dataset into train and validation, initialising a pytorch Dataloader to be passed into the train function
  - Initialises the rest of the arguments for the train method: neural network, optimiser, stores the parameters
  - ***train()*** function: this is the main function on this module which for a number of epochs given iterates through the train and validation dataset and passes the input data features to the neural network defined in model.py in order to adjust its weights. This is performed by calculating the quantile loss for the given FVC measurement and the output of the neural network. In addition, the this function executes a validation step using the Laplace Log Likehood function and printing it in the output of the program. At the end of each iteration the adjusted model is saved into a file in order to use it in the next execution. When all the epochs are performed the trained model it is stored in the *model_dir* folder.
- **predict.py -** this module is in charged of serving a request when the model is trained and deployed as a Model Endpoint in AWS Sagemaker. This script calls the model in evaluation mode to obtain inference for the input test dataset in order to obtain FVC measurements predictions for the patients.

The **Training-and-inference-Pytorch** notebook contains the scripts for training and testing the defined model in AWS SageMaker. It also analyses the results given by the inference exercise showing the evolution of the predictions per test patients and the confidence of the predictions using the quantile regression technique mentioned at the beginning of this report.

Refinement

In a first approach the *source* folder had been designed to train and run the model in a local environment, just by invoking the *training.py* and *predict.py* passing the right parameters in each of the functions. This helped in the debugging part of the development as I was able to understand and tweak the algorithm until it was ready.

When the code was ready, another iteration of the code has been performed adapting it in order to be deployed in a AWS SageMaker environment. Then the model was trained and deployed in the cloud as a model endpoint. This has helped to understand and to probe the potential industrialisation of a pulmonary fibrosis evolution model as a service.

Results

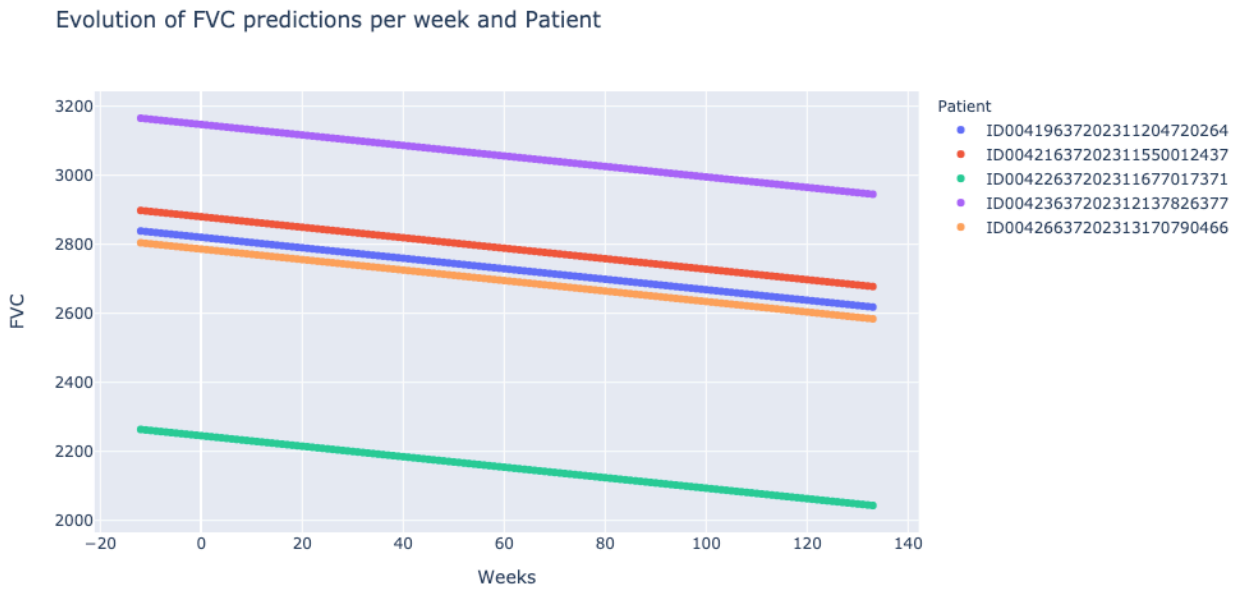
Model Evaluation and Validation

In this project we have evaluated the project using our test dataset provided as part of the Kaggle competition. As mentioned above the *pp_test.csv* contains a dataset with model input features for every week of the 5 patients presented in the original *test.csv*. Also a *results.csv* has been created with the structure of the output file:

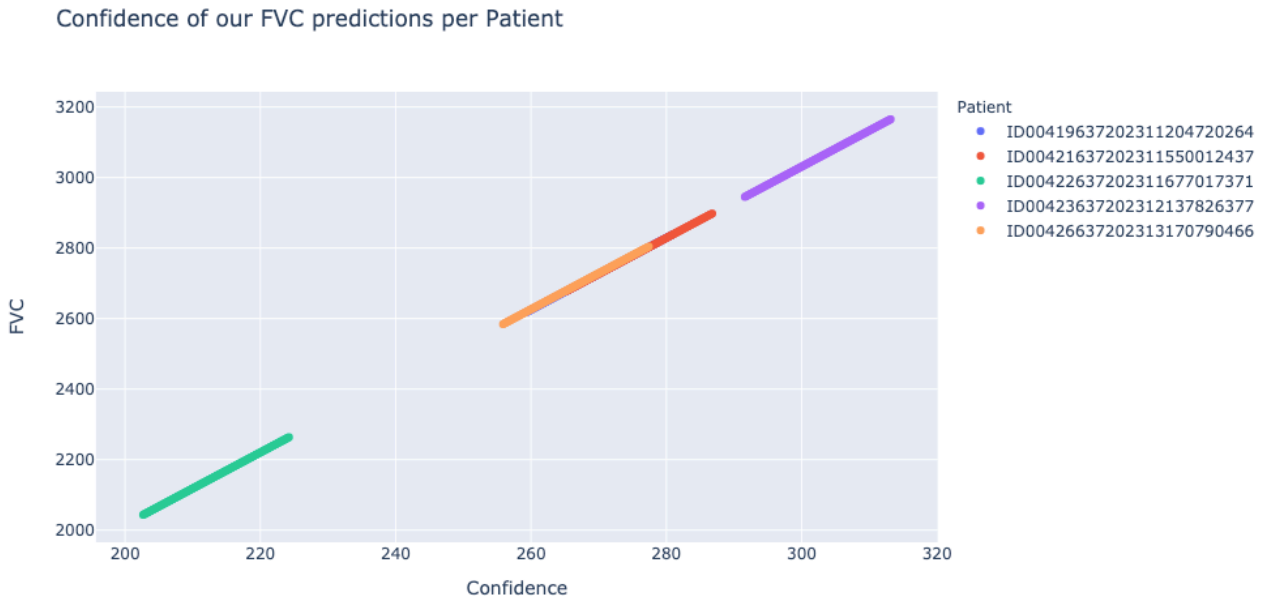
- Patient - identifies a unique patient
- Week - week for the predicted value
- FVC (prediction) - predicted lung capacity (in ml) as a result of the model evaluation

• Confidence - estimated error range of the FVC prediction (in ml)

In the image below we can see the prediction of the FVC by weeks for the 5 patients in the test dataset.



In the next graph it is shown the confidence of each of the predictions in ml per patient. This might indicate that there is a relationship between a FVC prediction and the confidence give, which should be analyse in any further work.



Justification

The predictions obtained with the model trained in this project provides FVC measurements for the 5 patients in the test dataset with reasonable confidence.

Future Work

It is worth noting that there are number of suggestions that can be done on the back of this project to further improve the predictions given and to position it better into a potential industrialised solution. Some of them are the following:

- Analyse the train dataset to understand the evolution of the capacity over weeks for different patients. The main idea is to look from evolution perspective to identify any dependency with the rest of features: Sex, Smoking Status and Age.
- Process CT scan images and come up with new features such us tissue density types or air volume. Include those new features into our custom PyTorch neural network.
- Implement the Laplace Log Likehood when calculating the loss, instead of just printing how it improves. This would help the implementation to perform better in the Kaggle competition
- Analyse in detail the confidence given per each of the predictions and attempt to narrow it in order to have more accurate predictions.
- Industrialise the deployed model within a platform (or application) to allow doctors and patients to benefit from this model.

References

- https://medium.com/the-artificial-impostor/quantile-regression-part-2-6fdbc26b2629
- https://www.wikiwand.com/en/Quantile_regression
- https://www.kaggle.com/carlossouza/quantile-regression-pytorch-tabular-data-only
- https://www.kaggle.com/titericz/tabular-simple-eda-linear-model
- https://www.kaggle.com/havinath/eda-observations-visualizations-pytorch
- https://www.kaggle.com/avirdee/understanding-dicoms