COMP9417 Group Project

Kaggle project: OSIC Pulmonary Fibrosis Progression

Xinyi Cui [redacted]

Table of Content

1. Introduction *1*

2. Implementation *2*

2.1 Clinical data and FVC measurement **2**

2.1.1 An overview of the data 2

2.1.2 Visualising the data relation 3

2.1.3 Data Pre-processing 3

2.2 Extracting feature from the CT scan **4**

2.2.1 An overview of the data 4

2.1.2 Hounsfield Unit (HU) scale 5

2.1.3 Lung tissue and disease tissue proportion 5

2.3 Build the machine learning model **6**

2.3.1 CNN model 6

2.3.2 Quantile-Regression 7

2.3.3 Random Forest Regressor 9

3. Evaluation *9*

3.1 Laplace Log Likelihood **9**

3.2 Comparing different ML models **9**

4. Conclusion *10*

Reference *11*

Appendix A *12*

1. Introduction

Our team chose a Kaggle project, namely OSIC Pulmonary Fibrosis Progression. Idiopathic pulmonary fibrosis is a progressive interstitial lung disease with no known cause (Fernández Fabrellas, Peris Sánchez, Sabater Abad and Juan Samper, 2018). Once diagnosed, the patient will suffer from varying degrees of inflammation and fibrosis in the lung resulting in a progressive decline in lung function and life quality. Moreover, the progression of the disease is hard to predict. In some individuals, the disease advances dramatically producing a stepwise loss of lung function and a rapid death, while in other patients, it progresses insidiously with a long incubation period (Robbie, Daccord, Chua and Devaraj, 2017). Without a clear prognosis, patients with IPF are continuously tortured by a huge amount of mental pressure and anxiety.

Our goal in this project is to utilise the machine learning technique to predict the patient's severity of decline in lung function based on the CT scan of their lungs (OSIC Pulmonary Fibrosis Progression | Kaggle, 2020). All the data used in this project can be found in the competition data section. The lung function can be reflected by the forced vital capacity (FVC) which measures the volume of air exhaled. Associated clinical information such as age, gender, smoking status and an entire history of FVC measurement will be provided to support the prediction. The basic approach starts with analysing and pre-processing the clinical data in order to find the best feature to predict the FVC measurement in the upcoming weeks. The CT scan for each patient will also be analysed slice by slice and any information correlated with the FVC measurement will be extracted. Finally, different machine learning algorithms will be trained and evaluated based on their performance score.

The motivation of this project is to improve the quality of information communicated to the patients and their carers and hence help them understand the prognosis better. A reliable prediction of disease progression will also enable the earlier identification of complications such as pulmonary hypertension and heart failure and help with deciding an appropriate time for lung transplant operation (Robbie, Daccord, Chua and Devaraj, 2017).

The rest of the report is organised as following: Section 2 is the implementation of the machine learning method where the clinical data is analysed and pre-processed in section 2.1; extra features from the CT scan are extracted in section 2.2; three machine learning models (CNN, Quantile-Regression, random forest) are built and tested in section 2.3. Section 3 is the evaluation of the machine learning method where the evaluation metric of the competition is explained in section 3.1; the advantages of each model as well as their scores are compared in section 3.2. Section 4 is the conclusion where the Quantile-Regression model with a score -7.571 is selected as our final decision.

2. Implementation

2.1 Clinical data and FVC measurement

2.1.1 An overview of the data

Clinical information such as Patient ID, age, sex, smoking status and an entire history of FVC measurement of each week will be provided to support the prediction. There is a total of 1549 data in the training set which contains 176 unique patient IDs. A detailed description of each data is listed in Fig 1.

	Patient	Sex	SmokingStatus
count	1549	1549	1549
unique	176	2	3
top	ID00421637202311550012437	Male	Ex-smoker
freq	10	1224	1038

	Weeks	FVC	Percent	Age
count	1549.000000	1549.000000	1549.000000	1549.000000
mean	31.861846	2690.479019	77.672654	67.188509
std	23.247550	832.770959	19.823261	7.057395
min	-5.000000	827.000000	28.877577	49.000000
25%	12.000000	2109.000000	62.832700	63.000000
50%	28.000000	2641.000000	75.676937	68.000000
75%	47.000000	3171.000000	88.621065	72.000000
max	133.000000	6399.000000	153.145378	88.000000

Figure 1 Detailed description of each data group

The “FVC” is the measured lung capacity in ml used to predict the lung function and the “Percent” approximates the patient FVC as a percentage of the typical FVC of a person with similar characteristics. The “Weeks” data is the time stamp of each FVC measurement relative to the time when the CT scan was obtained. Hence a positive value means the FVC was measured after the scan was obtained and a negative value means the FVC was measured before the scan was obtained. Since the training set is real medical data, the relative timing of FVC measurement varies widely for each patient. The histogram of the timing is shown in figure m, where the timing spans from -5 to 133 week. The FVC measurement of each week for one patient is displayed in Fig 2.

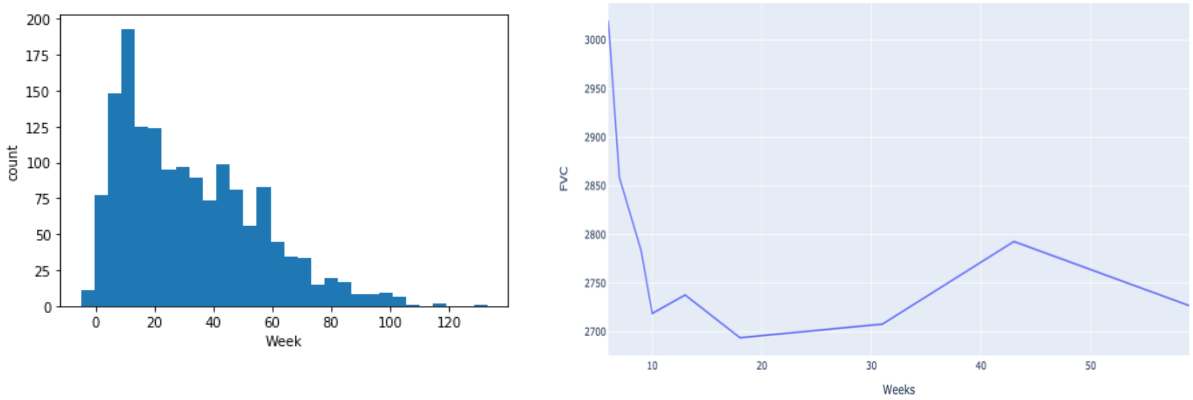

Figure 2 Histogram of the timing and FVC measurement of each week for one patient

2.1.2 Visualising the data relation

Figure 12 in appendix A shows tabular data attribute distributions and interactions between 'Sex' groups, blue indicates male and orange indicates female. The diagonal axes show the distribution of the "FVC", "Weeks", "Percent" and "Age" data of male and female. It is observed that FVC varies with gender and age. Among them, males generally have higher FVC than females. It does not make sense to compare the 'FVC' versus 'Weeks' of different genders, but it shows that the tested males generally have higher FVC values in the cycle than the females. As percent is computed by approximating one patient's FVC based on typical FVC of who has similar characteristics, 'Percent' distributions are different from gender and age, similar to FVC distribution. Among the tested population, women have a larger age span than men and have a shorter peak.

Figure 13 in appendix A emphasizes tabular data attribute distributions and interactions between 'Smoking Status' groups, blue, orange, green indicates ex-smoker, never smoked, currently smokes respectively. The diagonal axes show the distribution of the "FVC", "Weeks", "Percent" and "Age" data of different smoking status. Mean 'FVC' value in the currently smoked group is higher than other two groups of patients, which is unexpected. Distribution of 'Weeks' is similar for different 'Smoking Status'. As the percentage is derived from FVC value, the 'percent' distribution of the different Smoke Status groups is similar to the FVC distribution, but the peak of the ex-smoker and never smoked group is higher. 'Age' has almost nothing to associate with 'Smoking Status'.

2.1.3 Data Pre-processing

There are identical IDs in the 'Patient' column as multiple records are reserved for one patient in different weeks. As the 'Patient' field contains a unique ID of each patient, this column can be deleted from the dataframe. There are categorical data in 'Sex' and 'SmokingStatus' column. Then it is necessary to convert the categorical data to numeric data. To do this, we can either create dummy columns or map unique numbers to different categories. We choose the latter one, by mapping male to 0, female to 1 and mapping {0, 1, 2} to three different groups in SmokingStatus respectively. Moreover, we need to check if there are any missing or abnormal values. For the training dataset, there aren't any, so we don't have to do any editing here.

The only problem left is some values in the 'Sex' and 'SmokingStatus' field are quite small compared to value in 'FVC' or 'Percent' field, which may affect the result of the prediction as our model may treat the larger value field to be more significant than the smaller ones. Scaling the features helps us to prevent one or more features dominating the smaller scale variables. Therefore, scaling the numeric data is necessary. Note we choose to scale our target field 'FVC' as well, since the distribution of the values of FVC are a bit dispersed and its values are much larger than the others. In the scikit-learn preprocessing package, there is a useful tool called standardScaler which can help us to solve this problem. The scaling process the standardScaler uses is like:

$$z = \frac{x - u}{s}$$

where x is our sample to be scaled, u is the mean of training samples, s is the standard deviation of our training sample. Instead of using standardScaler, we can also do this mean removal

	Weeks	FVC	Percent	Age	Sex	SmokingStatus
0	-1.543106	-0.451025	-0.979923	1.674174	-0.515289	-0.423571
1	-1.155843	-0.572346	-1.108174	1.674174	-0.515289	-0.423571
2	-1.069785	-0.756129	-1.302454	1.674174	-0.515289	-0.423571
3	-0.983726	-0.656430	-1.197060	1.674174	-0.515289	-0.423571
4	-0.897668	-0.746519	-1.292296	1.674174	-0.515289	-0.423571

Figure 3 A few rows of the data frame

process manually. After applying the fitting and transforming by standard scaler, our dataset looks pretty neat. A few rows of the data frame are shown as Fig 3.

2.2 Extracting feature from the CT scan

2.2.1 An overview of the data

Lung CT scan is a three-dimensional crossing sectional imaging method which displays the internal structure of the lung (EGRIBOZ et al., 2019) as shown in figure 4. The lung is sliced from top to bottom and each scan contains information about the cross section of one particular slice. All the image data is saved in DICOM format standing for digital imaging and communications in medicine which is an international standard for medical imaging (EGRIBOZ et al., 2019).

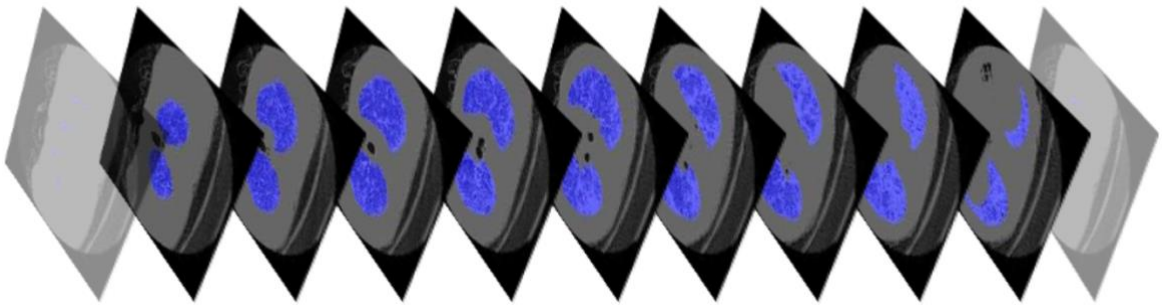

Figure 4 CT scan slices

A total number of 176 lung CT (as there are 176 patients in the training set) is used for extracting features to train the machine learning model. All the CT images are obtained from the Kaggle data page stored in a file named Train. There is a huge imbalance in the number of slices per patient where the maximum slice per patient is 1018 and the minimum slice per patient is 12. The histogram of the CT slice number is shown in figure 5.

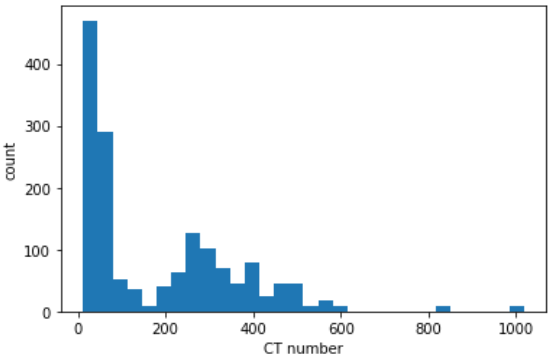

Figure 5 The histogram of the CT slice number

In order to have a better visualization, a CT scan for one patient (ID00011637202177653955184) which has 31 slices in total is displayed in figure 6.

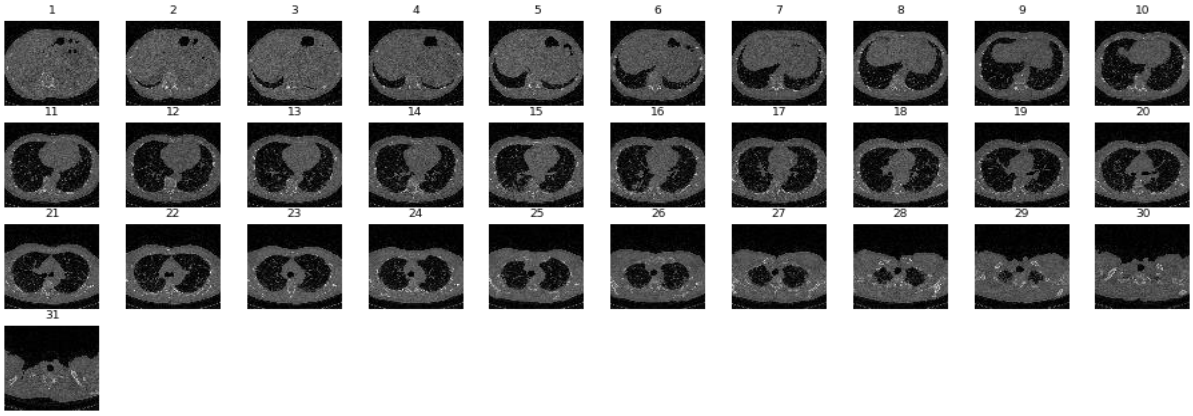

Figure 6 CT scan for one patient

2.1.2 Hounsfield Unit (HU) scale

The unit of measurement in CT scan is called the Hounsfield Unit which measures radiodensity of different tissues. Tissues containing similar substances will have a similar HU value as shown in Fig 7 (EGRIBOZ et al., 2019).

Pixel data from the DICOM image can be transferred to HU scale by the following formula.

$$HU = m * P + b$$

Where m is the RescaleSlope attribute of the DICOM image, b is the RescaleIntercept attribute of the DICOM image and P is the Pixel Array.

HOUNSFIELD UNIT SCALE VALUES

Substance	Value(HU)
Air	-1000
Lung	-700 to -600
Water	0
Blood	13 to 50
Kidney	20 to 45
Bone	200 to 3000
Gold	30000

Figure 7 HU value

The histogram of pixel data in HU scale for one patient is shown as figure 8. The frequency represents the portion of the tissue in different HU values.

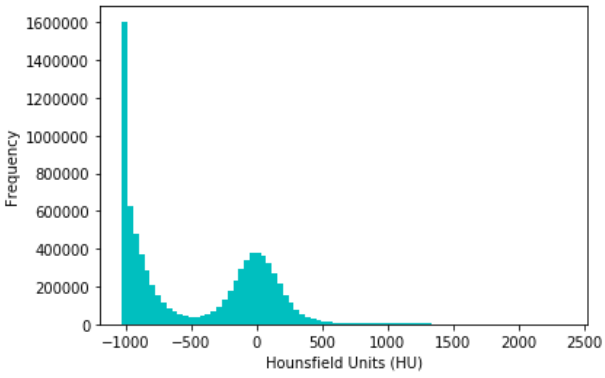

Figure 8 The histogram of pixel data in HU

2.1.3 Lung tissue and disease tissue proportion

As the Pulmonary Fibrosis progresses, the proportion of lung tissue will decrease and the proportion of the fibrous tissue will increase, which suggests we can predict the severity of the disease by measuring the proportion of both of these two tissues. The fibrous tissue has a HU

value of 50 to 70, while the lung tissue has a HU value of -700 to -600. By calculating the frequency of tissue in these two ranges, we can estimate the tissue density for the lung and fibrous part. As the total number of pixels varies for different patients, the lung tissue and fibrous tissue are normalised through dividing by the total number of pixels. Among all the 176 patients, there are four patients whose CT scan cannot be processed and result in an error, hence they are removed from the train set.

All the CT images will be processed using the CT_process.py which will compute the lung tissue and disease tissue for each patient. These two features will be appended to the training data frame. The CT_process.py takes around 3 hours to process all the images.

2.3 Build the machine learning model

2.3.1 CNN model

2.3.1.1 Process and Consolidate Table data

Process sample_submission file, split 'Patient_Week' into 'Patient' and 'Weeks' and remain 'Patient_Week' column. Drop 'Weeks' column in test file temporary and merge it with sample_submission file, associate them on 'Patient'. Mark train set as 'train' block, test set as 'value' block and sample_submission set as 'test' block, then append value set and test set after train set. All data after integration is placed in a table called data. In the data file, for each 'Patient', find the min value in 'Weeks' column, and create a new column 'min_week' to store this value. Meanwhile, set 'min_week' in the 'test' block not to be a number, in order to avoid affecting further steps.

Then create a base table to record the cases that 'Weeks' value is equal to 'min_week' value, this table is used to record the initial 'FVC' value and other information of each 'Patient'. In this base file, maybe one Patient has more than one 'initial_FVC', because it extracts from different block before, so apply cumulative sum over dataframe axis, and replace the lower 'initial_FVC', remain the cumulative sum be the 'initial_FVC'. In this way, the base table has two columns which are unique 'Patient' with its 'initial_FVC'.

Merge base table with data table on attribute 'Patient', base table on the left, create a new column 'week_from_min_week' by computing the difference between value in 'Weeks' column and value in 'min_week'. This is the current week 'Patient' takes CT scan compared with their first time. Then extract 'Sex' and 'Smoking Status' from data file, store different types of 'Sex' and 'Smoking Status' into an array 'FE'. Get the position of each feature of 'Patient', which is the percentage of a value between maximum value and minimum value in the range of feature value. Add these values into the array 'FE'.

2.3.1.2 Build CNN model

Load images from training patients' baseline CT scan in DICOM format, concatenated to data file which was processed before. Build a Convolutional Neural Networks (CNN) model used for image classification. Choose 'Patient' as input layer, output is a one-dimension tensor, the number of channels of output shape by the layer depend on the first parameter when the layer is declared (e.g. 9 in input layer). Define multiple dense layers, r1, l1, r3, and pass the output tensor after convolution to multiple dense layers to complete the classification. Layers are connected, each one is connected to the previous layer. Compile and train the model. The sample submission set has 3 classes, so the final Dense layer needs 3 outputs.

```
Model: "CNN"
__________________________________________________________________________________
Layer (type)                 Output Shape              Param #   Connected to
==================================================================================
Patient (InputLayer)         [(None, 9)]               0
__________________________________________________________________________________
r1 (Dense)                   (None, 100)               1000      Patient[0][0]
__________________________________________________________________________________
l1 (Dense)                   (None, 3)                 303       r1[0][0]
__________________________________________________________________________________
r3 (Dense)                   (None, 3)                 303       r1[0][0]
__________________________________________________________________________________
preds (Lambda)               (None, 3)                 0         l1[0][0]
                                                                 r3[0][0]
==================================================================================
Total params: 1,606
Trainable params: 1,606
Non-trainable params: 0
__________________________________________________________________________________
None
1606
```

```
FOLD 1
train value [48.594749450683594, 6.76880407333374]
value predict [53.25923156738281, 6.8834943771362305]
FOLD 2
train value [49.13568115234375, 6.759629249572754]
value predict [45.444210052490234, 6.764134883880615]
FOLD 3
train value [48.124298095703125, 6.758360385894775]
value predict [48.50204849243164, 6.764720916748047]
FOLD 4
train value [48.52439880371094, 6.744978904724121]
value predict [49.97163009643555, 6.779638290405273]
FOLD 5
train value [47.73243713378906, 6.729137420654297]
value predict [66.46054077148438, 6.973323345184326]
FOLD 6
train value [49.34126281738281, 6.792945861816406]
value predict [52.98618698120117, 6.776477336883545]
FOLD 7
train value [52.60466766357422, 6.8268280029296875]
value predict [52.99921798706055, 6.8393449783325195]
```

Figure 9 CNN model

Predict the result, using cross validation, K-fold cross validation. Split data into 7, which is the number of elements in previous array 'FE', recall it stores different types of 'Sex' and 'Smoking Status'. Basic idea is, in the first round, use the 1st fold as test data, 2rd to 7th folders as train data, etc… Then wrap the total 7 rounds predict process, get the mean value, to be the final predicted result.

Some future suggestions for the CNN model are adding skip connections that connect non-consecutive layers, which may help train deep models more efficiently; using blocks with this type of shortcut connections, in this way, can also create an additional path for the gradient to flow back more easily, which makes it easier to optimize the earlier layers. Better to go deeper than wider, however, a very tall and skinny network can be hard to optimize.

2.3.2 Quantile-Regression

This is the part where we use Quantile-Regression to predict the patient's severity of decline in lung function based on a CT scan of their lungs for all the possible weeks.

2.3.2.1 Standardization

From the Grouped data result, we can see FVC difference between a current smoker and non-smoker is around 1000, and FVC difference for female and male is also around 1000. Hence, we do not need to fix imbalanced datasets by oversampling instances of the minority or add weight. However, some of the maximum FVC result are way bigger than the mean value (e.g. Male, Ex-smoker), these bias results will affect the accuracy of the results. Hence, we manually

apply the mean removal method to centre data and remove the average value of each characteristic, and then scale it by dividing their standard deviation.

2.3.2.2 Performing Quantile-Regression

When we perform the analysis of these patients, we are also required to express how confident the prediction is. Unlike linear regression, Quantile-Regression can estimate the conditional medium and the standard deviation of the target result. We choose 3 models each with 15th quantile, 50th quantile, 85th quantile which indicate there are 15%, 50% and 85% chances of actual FVC is below the prediction. After training the model based on the Weeks, Percent, Age, Sex, Smoking Status, Fig 10 is the Quantile-Regression result. The percent and the Smoking Status have relative greater confidence, and Weeks has much lower standard error than other features.

QuantReg Regression Results

Dep. Variable:	FVC	Pseudo R-squared:	0.6104
Model:	QuantReg	Bandwidth:	120.8
Method:	Least Squares	Sparsity:	651.3
Date:	Fri, 07 Aug 2020	No. Observations:	1549
Time:	02:02:29	Df Residuals:	1543
		Df Model:	5

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2723.1519	14.084	193.349	0.000	2695.526	2750.778
Weeks	-0.7984	0.358	-2.232	0.026	-1.500	-0.097
Percent	643.2183	8.513	75.555	0.000	626.519	659.917
Age	-115.8680	8.340	-13.892	0.000	-132.228	-99.508
Sex	-429.8442	8.931	-48.127	0.000	-447.363	-412.325
SmokingStatus	20.3065	8.902	2.281	0.023	2.845	37.768

Figure 10 Quantile-Regression result

2.3.2.3 Predicting process

In order to predict any given test set. We need to have some pre-processing for the test dataset. The actual test set will contain any random patient with some random weeks. Given the test

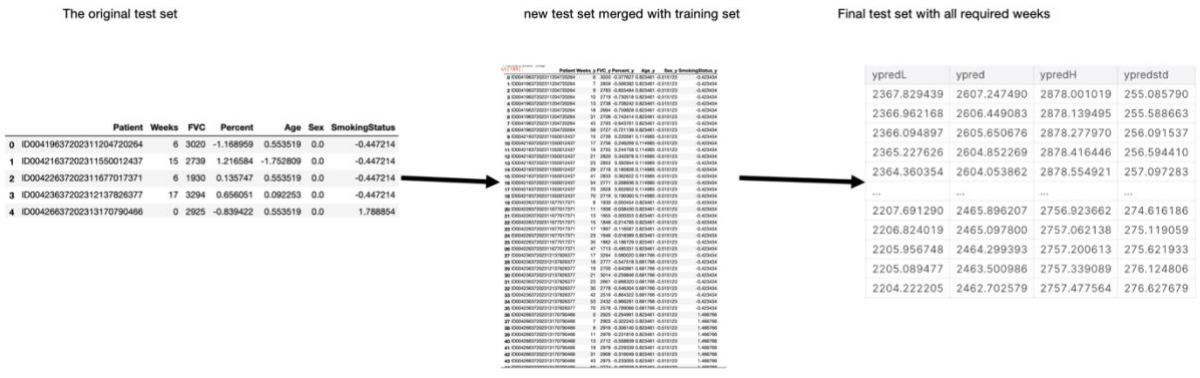

Figure 11 Some pre-processing for the test dataset

set, I first merge the test set with training that already finishes pre-processing, and form a new set, this set will contain more relevant and detailed information for each required patient. Finally, for each patient, we need to append the new test set for all the possible weeks, in this experiment, I used weeks from -12 to 134.

The first second third and fourth column each represent the result of quantile from 15%, 50% 85% and the predicted standard deviation with formula: *0.5*np.abs(result['85%']* -

*result['50%'])+0.5*np.abs(result['50%'] - result['5%'])*. And the predicted standard devotion will be used as the final confidence. Finally, Rearrange the data frame as required format, with first column be patient number + weeks, second column equal to the predicted FVC, and third column be the confidence.

2.3.3 Random Forest Regressor

The random forest regressor is adopted from the sklearn.ensemble module. After finishing the pre-processing and scaling, we extract the feature part and target part. For better accuracy and preciseness, we use k-fold cross validation to split the data into 5 to 10 folds (the results are pretty similar when cv = 5 or 10). Then we apply the random forest regressor to our data to make predictions. After that, we calculate the Laplace log likelihood of our prediction. One thing that needs notice is our prediction of FVC is scaled, therefore we need to scale it back to original magnitude then evaluate it using the Laplace log likelihood.

3. Evaluation

3.1 Laplace Log Likelihood

Specified in the evaluation page in the Kaggle competition, the prediction result is evaluated based on a modified version of the Laplace Log likelihood (OSIC Pulmonary Fibrosis Progression | Kaggle, 2020). The metric is designed to evaluate the confidence of the machine learning model by reflecting the accuracy and certainty of each FVC measurement. The metric is computed as following:

$$\sigma_{clipped} = (\sigma, 70)$$
$$\Delta = min\left(\left|FVC_{true} - FVC_{predicted}\right|, 1000\right)$$
$$metric = -\frac{\sqrt{2}\Delta}{\sigma_{clipped}} - ln\left(\sqrt{2}\sigma_{clipped}\right)$$

Where σ (standard deviation) is the confidence measure, which will be clipped at 70 ml, and the error Δ will be threshold at 1000 ml to avoid larger errors that affect the overall result. The metrics value will be negative and the higher the better. Although the FVC measurement of every possible week for the patient in the test set needs to be predicted, only the final three weeks will be used for scoring.

3.2 Comparing different ML models

Random forest is one supervised learning algorithm using ensemble learning methods to do the classification and regression. Compared with a single decision tree model, it improves the accuracy and controls overfitting by fitting different decision tree models on independent subsamples of the dataset and taking the average of the class prediction from each individual tree as the final decision. The advantages of random forest can be summed as 'the wisdom of crowds' where a large number of uncorrelated tree models form a committee to protect each other from making their individual mistakes (Understanding Random Forest, 2020).

Convolutional Neural Networks (CNN) model is flexible for image classification. The model will perform feature extraction on its own and does better when processing image data. As

there is no need to use traditional manual image processing methods, it is more likely to learn useful functions from the original data. Arden (2017) states that CNN models can automatically detect important functions without manual supervision. During the process of building CNN model, use image and filter convolution to generate invariant features and pass them to the next layer. In the next layer, the features are convolved with different filters to generate more constant and abstract features. Continue in each layer until the final output is obtained.

The Quantile-Regression model used for this project aims to estimate the linear relationship between FVC and other information of the patient with different quantile variables. The main goal of Quantile-Regression estimator is trying to minimize the residual square. By Comparing the Quantile-Regression model with the linear model, the linear model can work well if the data set is not complex, however, given there are several types of patient information plus some CT-scan image, the Quantile-Regression can explain conditional distribution of the different dataset more comprehensively, rather than just analysing the mean of the patient information data. Quantile-Regression also analyses how the different variables affect different quantiles of the explained variable. The three quartiles that we use 15%, 50%, 85% also help to explain how different patient information can affect the explained variable.

The scores for each model are: -6.9826 in CNN model; -6.88 ~ -7.48 in the random forest model; -7.571 for the Quantile-Regression model. The scores for the CNN and random forest model are obtained manfully where the model is tested on the data split from the original training data. The score for Quantile-Regression model is obtained by uploading the model on Kaggle where the model is tested on new data that has never seen by the model before. Due to a time limitation, the CNN and random forest model have not been uploaded on Kaggle, hence their real score should be worse than the score obtained. As a result, we select the Quantile-Regression model as our final decision. Our placing on the leader board is 645 at submission time.

4. Conclusion

In conclusion, we fulfill the requirements of the competition through utilising the machine learning technique to predict the patient's severity of decline in lung function based on the clinical data and a baseline CT scan. Clinical data of each patient are analysed and pre-processed. Exact features like lung tissue proportion and disease tissue proportion are extracted from the CT scan. Three machine learning models (CNN, Quantile-Regression, random forest) are implemented and evaluated based on their score. Their advantages and performance are discussed. Quantile-Regression model with a score -7.571 is selected as our final decision.

Reference

Arden, D. 2017. *Applied Deep Learning - Part 4: Convolutional Neural Networks, e-book,* accessed 8 August 2020, <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>.

EGRIBOZ, E., KAYNAR, F., VARLI, S., MUSELLIM, B. and SELCUK, T., 2019. Finding and Following of Honeycombing Regions in Computed Tomography Lung Images by Deep Learning. Computer Vision and Pattern Recognition.

Fernández Fabrellas, E., Peris Sánchez, R., Sabater Abad, C. and Juan Samper, G., 2018. Prognosis and Follow-Up of Idiopathic Pulmonary Fibrosis. Medical Sciences, 6(2), p.51.

Kaggle.com. 2020. OSIC Pulmonary Fibrosis Progression | Kaggle. [online] Available at: <https://www.kaggle.com/c/osic-pulmonary-fibrosis-progression/overview> [Accessed 5 August 2020].

Medium. 2020. Understanding Random Forest. [online] Available at: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2> [Accessed 8 August 2020].

Robbie, H., Daccord, C., Chua, F. and Devaraj, A., 2017. Evaluating disease severity in idiopathic pulmonary fibrosis. European Respiratory Review, 26(145), p.170051.

Appendix A

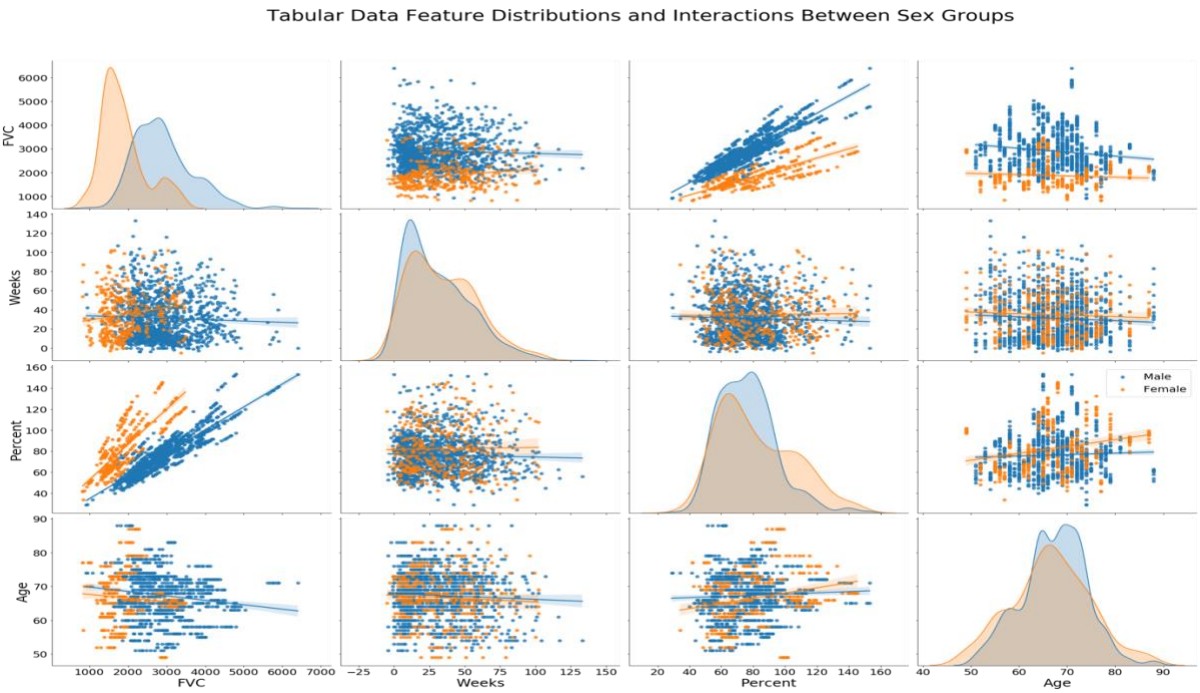

Figure 12 Tabular data attribute distributions and interactions between 'Sex' groups

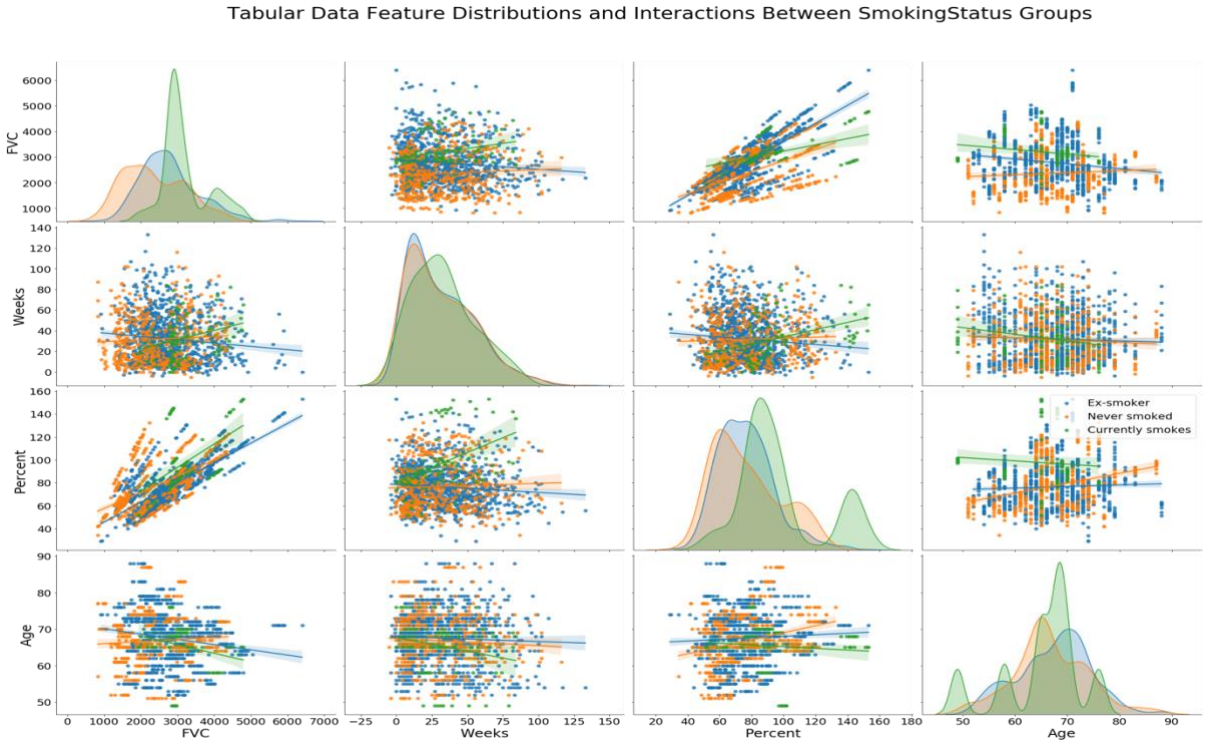

Figure 13 Tabular data attribute distributions and interactions between 'Smoking Status' groups