Project made by John : For projects click here <https://www.fiverr.com/john_alia>

**Bootstrap Method**

## The bootstrap method is a resampling technique used to estimate statistics on a population by sampling a dataset with replacement. It is a resampling method by independently sampling with replacement from an existing sample data with same sample size n, and performing inference among these resampled data.

## The client data is not large so that we could choose a subset of data to perform bootstrapping rather we shall perform bootstrap. We will perform bootstrap in order to retain a size with factor of 1.2 and the data will be split into three categories:

## Train ( 80 %)

## Test (10%)

## Validate (10 %)

## The client also required random taking of samples for training , validation and testing.

NOTE : CON variable is renamed as CON\_1 windows OS has issue declaring CON variable

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INPUT | | OUTPUT | COR Train | | | COR test | COR validate | MAE  train | MAE  test | | MAE  validate |
| A1-J16 | | IQ ( c ) | 0.95 | | | 0.85 | 0.81 | 4.05 | 5.10 | | 6.19 |
| A1-J16 | | SUICIDE ( b )-- | 0.76 | | | 0.85 | 0.90 | 0.35 | 0.25 | | 0.59 |
| A1-J16 | | AMBIG( c ) | 0.86 | | | 0.83 | 0.74 | 9.96 | 10.28 | | 13.09 |
| A1-J16 | | EGODYST( C ) | 0.85 | | | 0.81 | 0.64 | 9.45 | 10.62 | | 14.69 |
| A1-J16 | | AUTISM ( c ) | 0.86 | | | 0.44 | 0.71 | 1.64 | 2.75 | | 2.58 |
| A1-J16 | | SELF\_DIR(c) | 0.45 | | | 0.50 | 0.29 | 1.1 | 1.1 | | 1.4 |
| A1-j16 | | IDENTITY(c) | 0.87 | | | 0.36 | 0.78 | 0.87 | 1.3 | | 0.87 |
| A1-j16 | | EMPATHY (c) | 0.84 | | | 0.72 | 0.69 | 1.11 | 1.35 | | 1.31 |
| A1-J16 | | Intimacy ( c ) | 0.78 | | | 0.79 | 0.51 | 0.40 | 0.41 | | 0.57 |
| A1-J16 | | ACUTE  ( P ) | 0.60 | | | 0.32 | 0.39 | 1.78 | 1.73 | | 1.90 |
| A1-j16 | | CHRONIC  ( p ) | 0.74 | | | 0.65 | 0.54 | 1.16 | 1.34 | | 1.20 |
| A1-j16 | | TRAUMA\_PAST  ( C ) | 0.99 | | | 0.77 | 0.76 | 8.3 | 8.57 | | 8.69 |
| A1-j16 | | TRAUMA\_PRESENT  ( C ) | 0.93 | | | 0.78 | 0.93 | 3.18 | 3.78 | | 3.49 |
| A1-j16 | | Hope ( C ) | 0.99 | | | 0.60 | 0.61 | 0.29 | 0.56 | | 0.51 |
| A1-J16 | | SELFAFFICACY  ( p ) | 0.83 | | | 0.83 | 0.58 | 4.42 | 4.31 | | 4.21 |
| A1-J16 | | SELFESTEEM  ( c ) | 0.58 | | | 0.39 | 0.31 | 1.92 | 1.71 | | 2.1 |
| A1-j16 | | WELLBEING  ( c ) | 0.94 | | | 0.65 | 0.72 | 0.53 | 0.66 | | 0.69 |
| A1-J16 | | HYPER ( C ) | 0.89 | | | 0.60 | 0.77 | 9.3 | 9.5 | | 11.06 |
| A1-J16 | | AGGR ( C ) | 0.82 | | | 0.56 | 0.83 | 7.48 | 7.5 | | 9.15 |
| A1-J16 | | COND\_PR  ( C ) | 0.72 | | | 0.50 | 0.77 | 8.4 | 9.1 | | 9.3 |
| A1-j16 | | ANXIE  ( c ) | 0.78 | | | 0.49 | 0.75 | 9.07 | 9.61 | | 11.15 |
| A1-j16 | | DEPR  ( C ) | 0.49 | | | 0.47 | 0.71 | 10.16 | 9.44 | | 13.15 |
| A1-J16 | | SOMAT ( C ) | 0.59 | | | 0.31 | 0.65 | 10.08 | 11.07 | | 13.98 |
| A1-J16 | | ATYPI  ( C ) | 0.97 | | | 0.84 | 0.86 | 9.65 | 7.64 | | 13.37 |
| A1-j16 | | WITHD  ( c ) | 0.86 | | | 0.56 | 0.87 | 8.58 | 8.36 | | 10.51 |
| A1-j16 | | ATTE\_PR  ( c ) | 0.98 | | | 0.49 | 0.62 | 6.5 | 6.69 | | 6.27 |
| A1-j16 | | ANG\_CONT  ( c ) | 0.97 | | | 0.59 | 0.83 | 3.52 | 4.07 | | 4.99 |
| A1-J16 | | DEV\_SOC\_DIS  ( C ) | 0.91 | | | 0.73 | 0.78 | 5.17 | 5.19 | | 6.08 |
| A1-J16 | | BULLY  ( C ) | 0.95 | | | 0.82 | 0.86 | 6.44 | 6.71 | | 9.27 |
| A1-J16 | | EMOT\_SELF\_C  ( C ) | 0.50 | | | 0.33 | 0.60 | 9.2 | 8.89 | | 11.13 |
| A1-J16 | | EXE\_FUN  ( C ) | 0.52 | | | 0.36 | 0.62 | 6.59 | 6.9 | | 7.57 |
| A1-J16 | | MANIA  ( C ) | 0.90 | | | 0.89 | 0.83 | 2.3 | 2.93 | | 3.89 |
| A1-J16 | | NEG\_EMOT  ( C ) | 0.97 | | | 0.69 | 0.78 | 7.22 | 6.88 | | 7.62 |
| A1-J16 | | ADAPT  ( C ) | 0.96 | | | 0.63 | 0.86 | 7.09 | 7.68 | | 7.86 |
| A1-J16 | | SOC\_SK  ( C ) | 0.97 | | | 0.58 | 0.67 | 7.96 | 8.15 | | 9.13 |
| A1-j16 | | LEAD  ( c ) | 0.96 | | | 0.56 | 0.79 | 7.47 | 6.66 | | 7.64 |
| A1-j16 | | ACT\_DAIL  ( c ) | 0.96 | | | 0.62 | 0.71 | 6.8 | 7.12 | | 5.79 |
| A1-J16 | | FUN\_COM  ( c ) | 0.98 | | | 0.69 | 0.71 | 9.51 | 8.59 | | 10.33 |
| A1-j16 | | RESILI  ( c ) | 0.98 | | | 0.61 | 0.80 | 3.4 | 3.81 | | 3.88 |
| A1-j16 | | ANHEDONIA  ( c ) | 0.68 | | | 0.59 | 0.35 | 0.44 | 0.63 | | 0.76 |
| A1-j16 | | ANXIOUSNESS  ( p ) | 0.88 | | | 0.72 | 0.64 | 0.59 | 0.67 | | 0.75 |
| A1-j16 | | ATTENTION\_SEEKING  ( p ) | 0.91 | | | 0.68 | 0.64 | 0.50 | 0.67 | | 0.80 |
| A1-j16 | | CALLOUSNESS  ( p ) | 0.96 | | | 0.73 | 0.81 | 0.09 | 0.25 | | 0.18 |
| A1-j16 | | DECEITFULNESS  ( p ) | 0.87 | | | 0.79 | 0.58 | 0.28 | 0.36 | | 0.51 |
| A1-j16 | | DEPRESSIVITY  ( p ) | 0.97 | | | 0.89 | 0.69 | 0.085 | 0.200 | | 0.268 |
| A1-j16 | | ECCENTRICITY  ( p ) | 0.89 | | | 0.58 | 0.78 | 0.58 | 0.67 | | 0.80 |
|  | |  |  | | |  |  |  |  | |  |
| A1 – j16 | | EMOTIONAL\_LABILITY  (p) | 0.85 | | | 0.73 | 0.34 | 0.42 | 0.50 | | 0.75 |
| A1-j16 | | GRANDIOSITY  ( p ) | 0.76 | | | 0.68 | 0.48 | 0.41 | 0.50 | | 0.70 |
| A1 – j16 | | HOSTILITY  ( p ) | 0.99 | | | 0.82 | 0.85 | 2.77 | 2.78 | | 2.72 |
| A1-j16 | | IMPULSIVITY  ( p ) | 0.97 | | | 0.81 | 0.85 | 0.25 | 0.40 | | 0.35 |
| A1-j16 | | INTIMACY\_AVOIDANCE  ( p ) | 0.98 | | | 0.94 | 0.77 | 0.11 | 0.17 | | 0.33 |
| A1-j16 | | IRRESPONSIBILITY  ( p ) | 0.99 | | | 0.80 | 0.82 | 0.55 | 0.65 | | 0.64 |
| A1-J16 | | MANIPULATIVENESS  ( p ) | 0.82 | | | 0.75 | 0.56 | 0.40 | 0.44 | | 0.58 |
| A1-J16 | | PERCEPTUAL\_DYSREGULATION  ( P ) | 0.86 | | | 0.76 | 0.75 | 0.35 | 0.48 | | 0.40 |
| A1 –J16 | | PERSEVERATION  ( p ) | 0.97 | | | 0.831 | 0.835 | 0.94 | 0.95 | | 0.95 |
| A1-j16 | | RESTRICTED\_AFFECTIVITY  ( c ) | 0.82 | | | 0.72 | 0.61 | 0.60 | 0.70 | | 0.77 |
| A1-J16 | | RIGID\_PERFECTIONISM  ( P ) | 0.99 | | | 0.92 | 0.78 | 0.23 | 0.30 | | 0.42 |
| A1-J16 | | RISK\_TAKING  ( P ) | 0.98 | | | 0.86 | 0.80 | 0.52 | 0.67 | | 0.67 |
| A1-J16 | | SEPARATION\_INSECURITY  ( P ) | 0.98 | | | 0.85 | 0.79 | 0.68 | 0.75 | | 0.69 |
| A1-J16 | | SUBMISSIVENESS  ( P ) | 0.98 | | | 0.89 | 0.72 | 0.630 | 0.632 | | 0.67 |
| A1-J16 | | SUSPICIOUSNESS  ( P ) | 0.98 | | | 0.90 | 0.79 | 0.51 | 0.66 | | 0.54 |
| A1-J16 | | UNUSUAL\_BELIEFS  ( P ) | 0.91 | | | 0.87 | 0.53 | 0.28 | 0.34 | | 0.49 |
| A1-J16 | | WITHDRAWAL  ( P ) | 0.99 | | | 0.91 | 0.79 | 0.33 | 0.47 | | 0.53 |
| A1-J16 | | ACT  ( c ) | 0.99 | | | 0.88 | 0.91 | 1.22 | 6.13 | | 4.79 |
| A1-j16 | | DSP  ( c ) | 0.99 | | | 0.97 | 0.93 | 1.60 | 3.33 | | 4.22 |
| A1-J16 | | PAG  ( C ) | 0.99 | | | 0.89 | 0.89 | 1.16 | 5.25 | | 5.18 |
| A1-J16 | | PRO  ( C ) | 0.99 | | | 0.91 | 0.88 | 1.34 | 5.69 | | 5.59 |
|  | |  |  | | |  |  |  |  | |  |
| A1-J16 | | SPL  ( C ) | 0.99 | | | 0.80 | 0.90 | 1.39 | 8.98 | | 5.27 |
| A1-J16 | | WDR  ( C ) | 0.99 | | | 0.76 | 0.79 | 1.11 | 7.03 | | 5.93 |
| A1-J16 | | CON\_1  ( C ) | 0.99 | | | 0.90 | 0.92 | 2.06 | 9.23 | | 6.9 |
| A1-j16 | | FAN  ( C ) | 0.99 | | | 0.91 | 0.88 | 1.05 | 5.14 | | 4.9 |
| A1-J16 | | OMN  ( C ) | 0.99 | | | 0.83 | 0.91 | 1.32 | 7.84 | | 5.39 |
| A1-j16 | | REP  ( C ) | 0.99 | | | 0.94 | 0.90 | 1.08 | 3.49 | | 4.13 |
| A1-J16 | | SUB  ( C ) | 0.93 | | | 0.81 | 0.66 | 5.36 | 10.57 | | 9.17 |
| A1-J16 | | UND  ( C ) | 0.99 | | | 0.86 | 0.92 | 1.4 | 7.87 | | 4.7 |
| A1-j16 | | ALT  ( c ) | 0.99 | | | 0.64 | 0.80 | 1.00 | 8.6 | | 5.67 |
| A1-J16 | | HUM  ( C )` | 0.97 | | | 0.71 | 0.78 | 3.22 | 11.9 | | 8.56 |
| A1-J16 | | IDL  ( C ) | 0.96 | | | 0.82 | 0.81 | 2.68 | 7.013 | | 5.62 |
| A1-J16 | | RFM  ( C ) | 0.96 | | | 0.87 | 0.83 | 4.49 | 9.66 | | 8.00 |
| A1-J16 | | DN  ( C ) | 0.97 | | | 0.84 | 0.78 | 3.29 | 9.48 | | 8.81 |
| A1-J16 | | INT  ( C ) | 0.97 | | | 0.81 | 0.78 | 2.99 | 9.22 | | 8.8 |
| A1-J16 | | SUP  ( C ) | 0.97 | | | 0.80 | 0.84 | 2.99 | 10.11 | | 8.60 |
| A1-J16 | | ANT  ( C ) | 0.97 | | | 0.88 | 0.74 | 2.8 | 5.9 | | 8.7 |
| A1-J16 | | ISO  ( C ) | 0.98 | | | 0.89 | 0.89 | 2.3 | 5.53 | | 6.3 |
| A1-j16 | | DISTRACTIBILITY  ( p ) | 0.99 | | | 0.86 | 0.76 | 0.25 | 0.39 | | 0.52 |
| Predicted 25 var | | SEL  ( b ) | 0.83 | | | 0.66 | 0.61 | 6.16 | 12.7 | | 7.88 |
| Predicted 25 var  A1-j16 | | SEV  ( b )  DSC ( C ) | 0.96  0.98 | | | 0.75  0.81 | 0.82  0.88 | 5.6  2.93 | 11.4  8.6 | | 10.77  \_\_\_\_\_\_\_  7.1 |
| A1-j16 | SOM ( C ) | | | 0.95 | 0.70 | | 0.87 | 4.9 | | 12.3 | 8.25 |

**It was observed that machine learning models designed for only Increasing of Pearson correlation let to very large MAE and Pearson up to 0.99 but MAE was observed up to 100. So the balance of both worlds was adapted in order to both minimize the loss in MAE and maximize Pearson correlation**

Objective Function =[ maximize ( Pearson ) + minimize ( MAE ) ] \_\_\_\_ C

Objective function = [ maximize (pearson)]\_\_\_\_p

Objective function = MAE\_\_\_\_b

Hundreds of experiments were performed on variables some variable showed higher accuracy on EQUATION A, Some B and other a single suicide on C.

The variables with there corresponding objective have written brackets in brackets in front of them i.e (p) ( c ) or ( b )

Regularization Technique implemented in Algorithms:

What is regularization?

Regularization is a technique which makes slight modifications to the learning algorithm such that the model generalizes better. This in turn improves the model’s performance on the unseen data as well.

## Different Regularization Techniques in Deep Learning

Now that we have an understanding of how regularization helps in reducing overfitting, we’ll learn a few different techniques in order to apply regularization in deep learning.

### L2 & L1 regularization

L1 and L2 are the most common types of regularization. These update the general cost function by adding another term known as the regularization term.

Cost function = Loss (say, binary cross entropy) + Regularization term

Due to the addition of this regularization term, the values of weight matrices decrease because it assumes that a neural network with smaller weight matrices leads to simpler models. Therefore, it will also reduce overfitting to quite an extent.

However, this regularization term differs in L1 and L2.

In L2, we have:

https://cdn.analyticsvidhya.com/wp-content/uploads/2018/04/Screen-Shot-2018-04-04-at-1.59.54-AM.png

Here, **lambda** is the regularization parameter. It is the hyperparameter whose value is optimized for better results. L2 regularization is also known as weight decay as it forces the weights to decay towards zero (but not exactly zero).

In L1, we have:

https://cdn.analyticsvidhya.com/wp-content/uploads/2018/04/Screen-Shot-2018-04-04-at-1.59.57-AM.png

In this, we penalize the absolute value of the weights. Unlike L2, the weights may be reduced to zero here. **Hence, it is very useful when we are trying to compress our model. Otherwise, we usually prefer L2 over it.**

Regularization technique was applied to avoid over fitting the data so it performs good on the testing data.

The following below are statistics of regularization technique:

Kernel regularization = L1\_L2 (L1=1e-5,L2=1e-4),

Bias regularization = L2 (1e-4),

Activity regularization = L2 (1e-5)

These regularization were applied to neurons in the second layer in the variable terms these could be seen in formulas as L2NX where x were all the neurons mostly 0-2047.

**Clients Request for .632 method for sampling from data set .**

**What is 0.632 method ?**

In other words, our sample of n items will contain approx. n\*0.632 (63.2%) of the items in the dataset (since we sampled with replacement from a uniform distribution). This means that we are resampling the original data with only 63 % of the data with the same distribution as the original population. Client’s data was already small if only 63 % of the data was sampled 38 % of the data would be lost resulting in poor results . It was tried a severe over fitting was taking place. We could never down sample the data as it was already so small .

**What did we do ?**

We resampled the whole dataset with replacement by a factor of 1.2 . There was no way that the data could be down sampled . The distributions of data before sampling and after sampling was same . If you re run the training file . The distribution before and after resampling is displayed in results that is same as stated in 0.632 method. The testing ,validating and training data was chosen randomly as demanded by client.

**What is relu ?**

In a neural network, the activation function is responsible for transforming the summed weighted input from the node into the activation of the node or output for that input.

The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

**How relu was implemented in your project ?**

If you see the python scripts and the formulas for functions producing output variables you may observe that some variables are wrapped in brackets with relu written outside i.e. relu( L2N76) it means that the term inside the brackets will become negative if the value inside the brackets is negative otherwise it will output the same value .

**FAQ:**

**What is training data ?**

The training data is an initial set of data used to help a program understand how to apply technologies like neural networks to learn and produce sophisticated results

**What is validation data ?**

A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's parameters. There is much confusion in applied machine learning about what a validation dataset is exactly and how it differs from a test dataset.

**What is test data ?**

A test dataset is a dataset that is independent of the training dataset, In a scenario where both validation and test datasets are used, the test dataset is typically used to assess the final model that is selected during the validation process. In the case where the original dataset is partitioned into two subsets (training and test datasets), the test dataset might assess the model only once

**What is difference in validation and test data?**

Validation set is different from test set. Validation set actually can be regarded as a part of training set, because it is used to build your model, neural networks or others. It is usually used for parameter selection and to avoid over fitting. Test set is used for performance evaluation.

Basic difference between Training Validation and test sets are as follows:

[1. Training](https://www.researchgate.net/deref/http%3A%2F%2F1.Training) Set: This is the data that used by the training algorithm to adjust the weights of the network.

2. Validation set (Development set): This data is used during training to assess how well the Neural Network is currently performing — the performance of the network on this data may

be used to guide the training in some way (e.g. controlling the learning rate, deciding

when to stop training, choosing between several trained networks).

3. Test set: This is the genuine test data — and, ideally, should be used once

Only , after training is complete.

NOTE:

The testing validation data taken random after bootstrapping is saved in relevant folder.

Variablename\_testing.xlsx

Variablename\_validating.xlsx

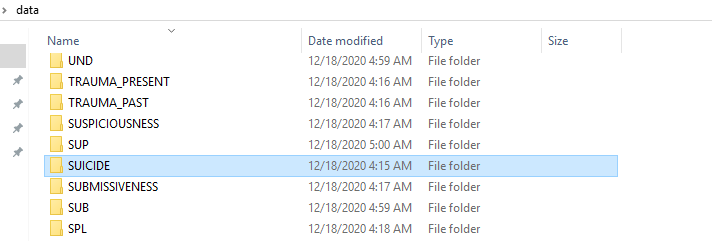
Variablename\_training.xlsx

Project made by John : For projects click here <https://www.fiverr.com/john_alia>

LETS TAKE EXAMPLE OF VARIABLE SUICIDE :

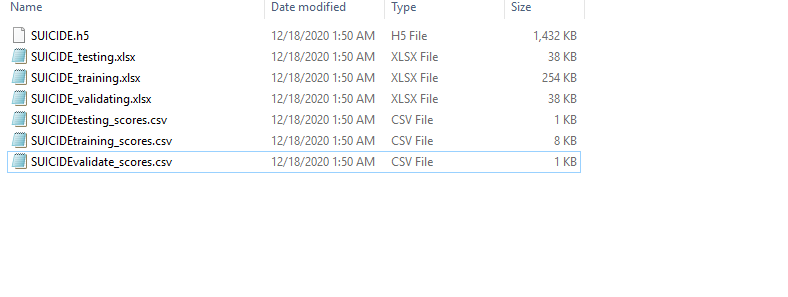
In the data folder all the variables have a folder with their name :

Lets open the suicide variable



In the folder

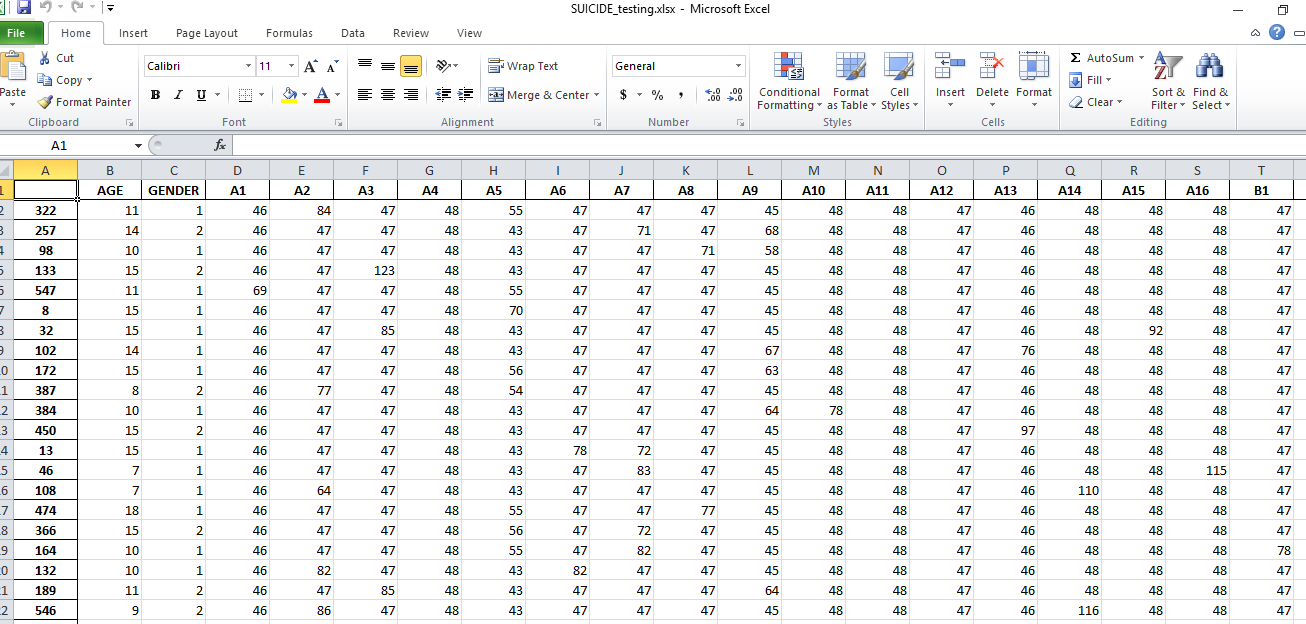
1. Training testing validating input files along with scores.
2. Deep learning .h5 model saved



Project made by John : For projects click here <https://www.fiverr.com/john_alia>

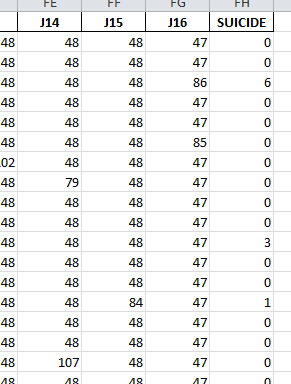
If we open testing, validating , training .xlsx file we see as following:

Lets open suicide\_testing.xlsx file. Same symmetry applies with training and validating

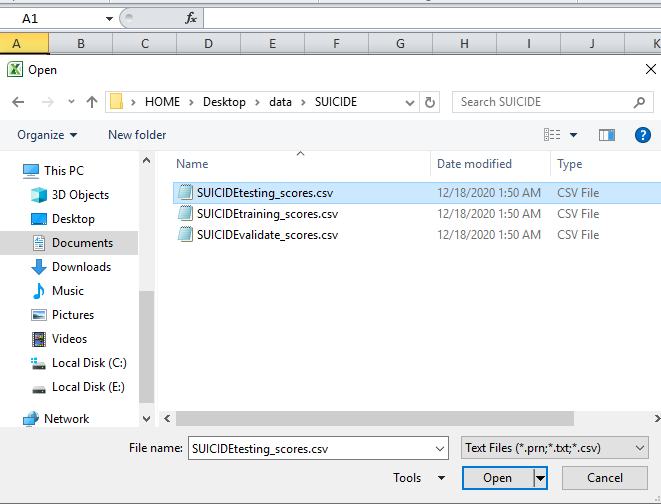


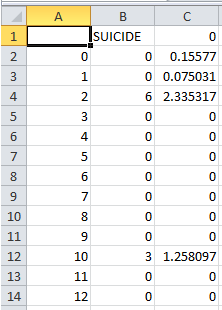
In the first column you can see the random sample that was chosen from the original dataset for testing.xlsx

This is the last column of the Excel file is the ground truth.



Lets open SUICIDEtesting\_score.csv file for the testing data predicted by the model.



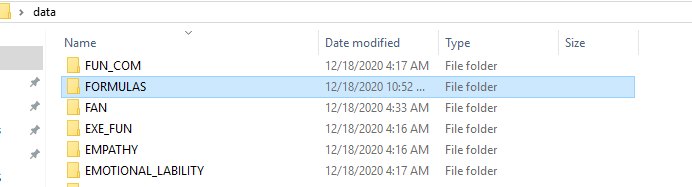


DON’T WORRY ABOUT ACCURACY OVERALL ACCURACY IS GOOD IT CAN NOT BE JUDGED BY VIEWING THE FILE BUT RATHER APPLYING FORMULA ON WHOLE

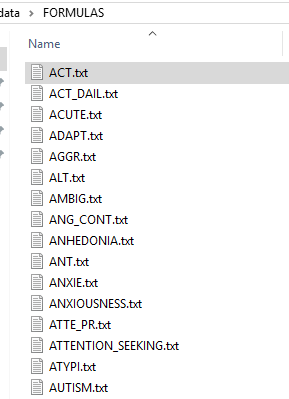
The 2nd column is the real data whereas the third column is predicted column . The rows no of testingscores.csv have number correspondence with the suicide\_testing.xlsx file .

Now lets move to formulas:

There is a formulas folder in the data folder . It may be provided you separate because it is 1.4 GB fiverr allows only to upload 1 GB in project

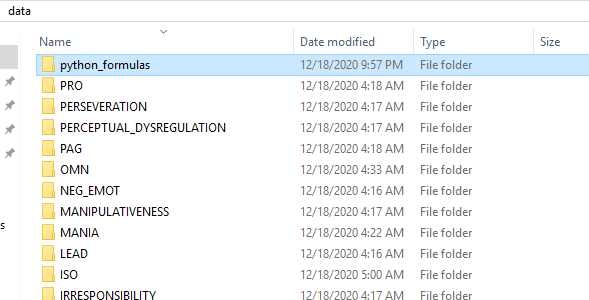


Every folder has its separate formula file :



Project made by John : For projects click here <https://www.fiverr.com/john_alia>

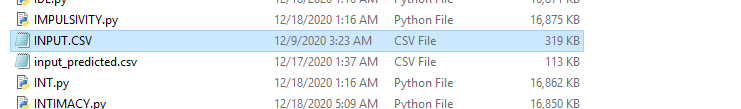
Lets see the python scripts now and run them of formulas :



You can see the python\_formulas folder it contains formula script for each file lets run a script :

The input in the script is INPUT0N[X] where x ranges from 1-162. When running the script it reads the input.csv for all variables except sel and sev for them it reads the 25 variable predicted input then predicts the equation on it.

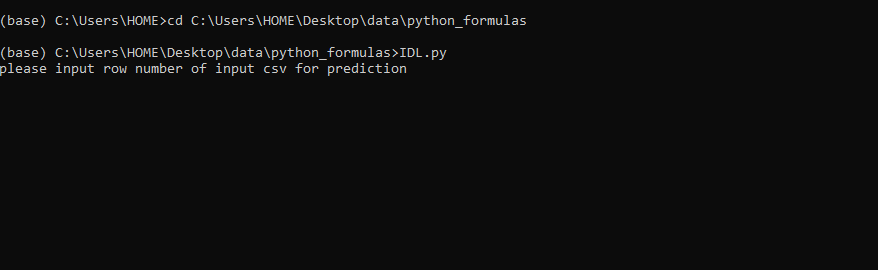
The INPUT. Csv is the same data provided by user it has only the A1-J16 separated for input.By reading the input.csv it ask user to select the row of the csv containing to A1-J16 to predict the script .

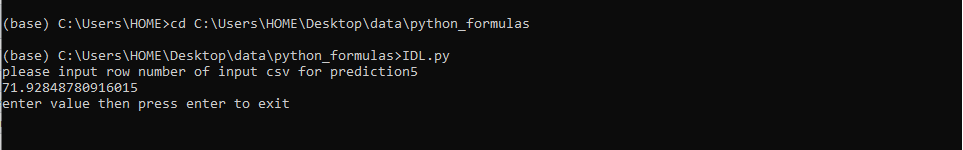


For SEL and SEV it is the input predicted.csv as desired by the client.

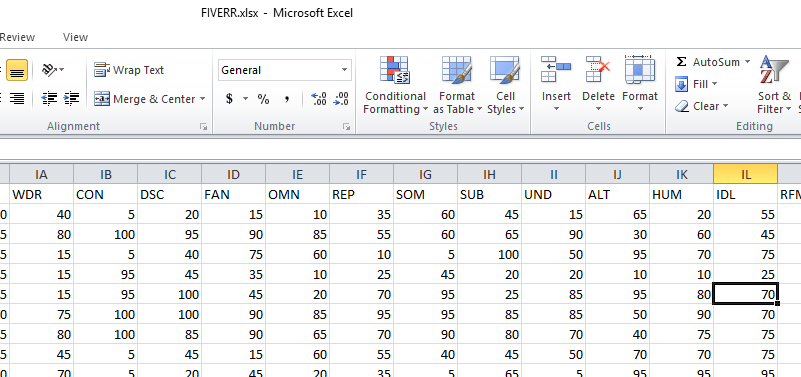
Project made by John : For projects click here <https://www.fiverr.com/john_alia>

LETS RUN A RANDOM SCRIPT IDL.py





It predicts on the input row of 162 inputs A1-J16 the input portion of fiverr.xlsx as provided by the user. You cam math the outputs from formula and .h5 files they are same with some error but the overall error is small and correlation is established. Lets see the IDL 5th row value

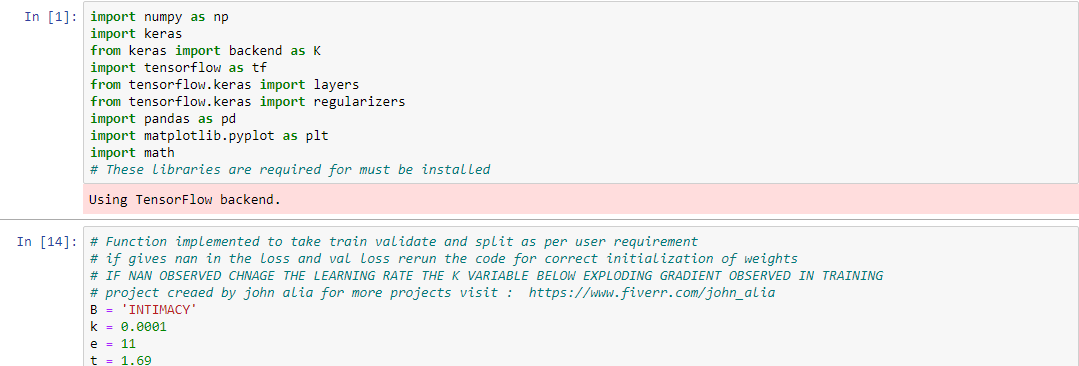


71 predicted and real value was 70.

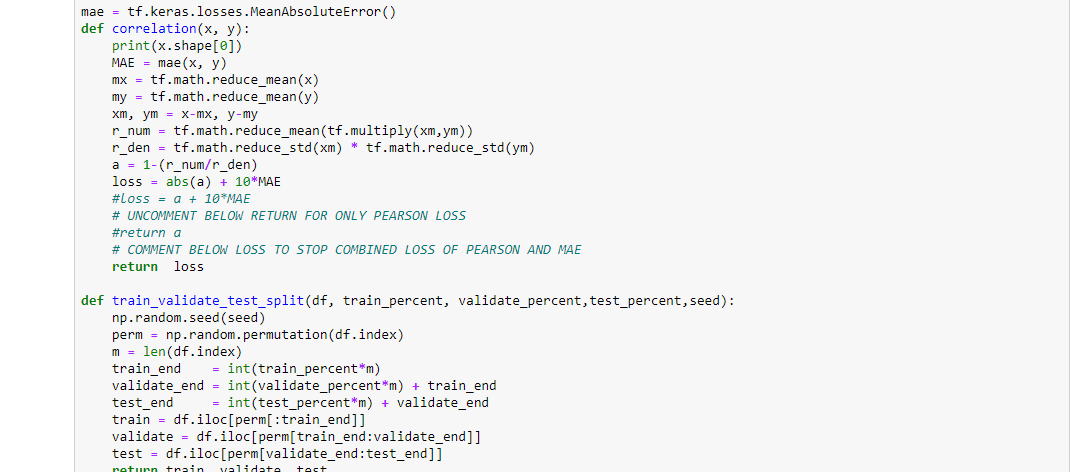
Project made by John : For projects click here <https://www.fiverr.com/john_alia>

LETS SEE THE CODING NOW .

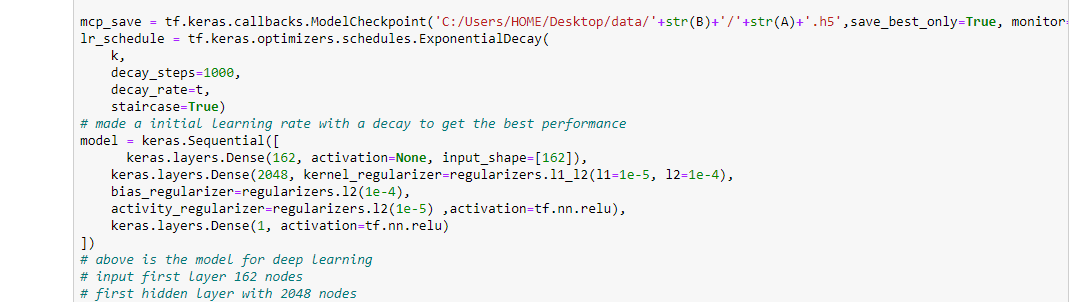
THE TRAINING FILE FOR ALL EXCEPT SEL AND SEV .



IMPORTING libraries and all the stuff.

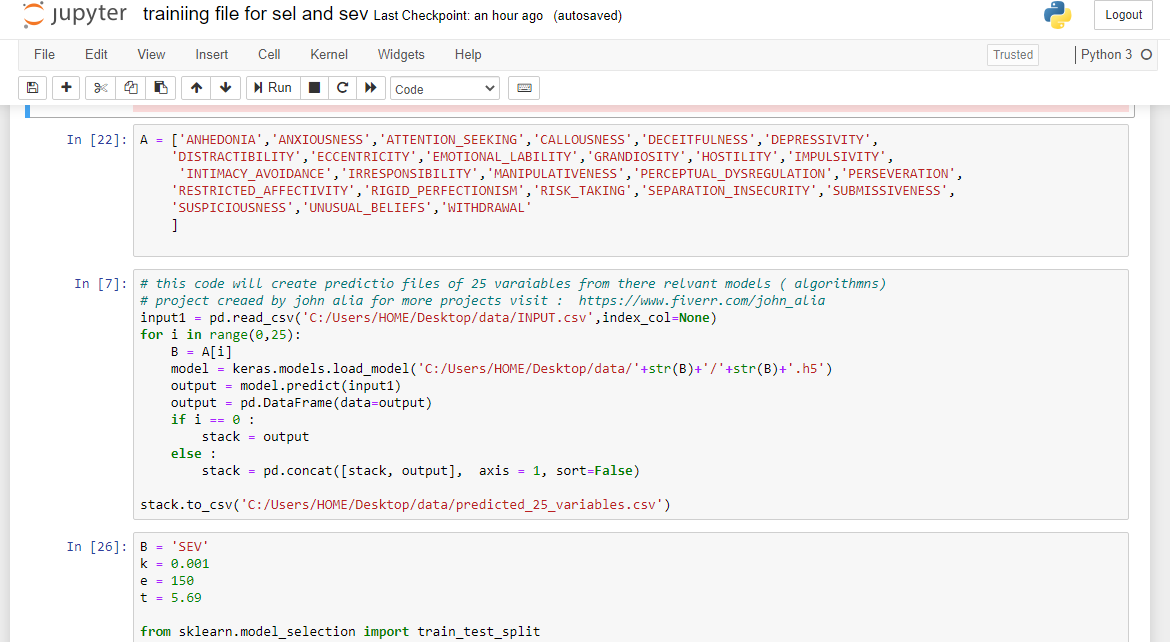


Defining correlation for training and functions as discussed in documentation. Randomly taking test train split.

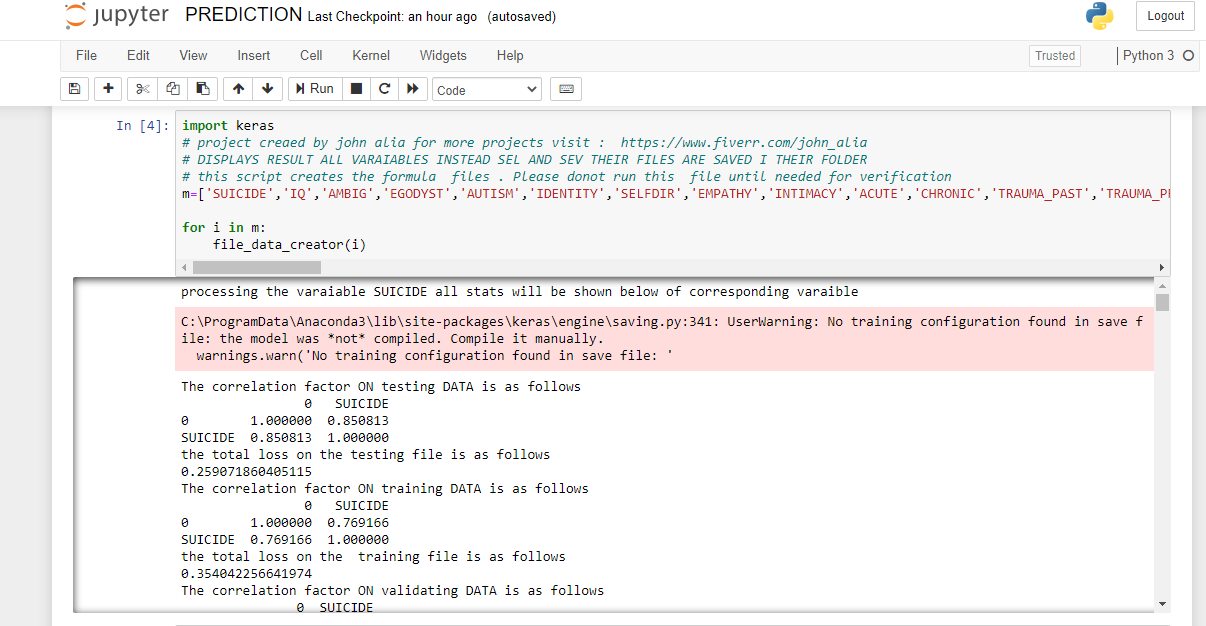
Architecture of the deep learning model with all the regularization implemented in python

Project made by John : For projects click here <https://www.fiverr.com/john_alia>

Training file for sel and sev



First taking prediction from 25 elements then training the model.



This prediction.ipynb takes in the machine learning model outputs all the stats correlation and MAE on test, validate and training . It’s output is saved in RESULT\_OUTPUT\_SCRIPT.txt in data folder.