**Final Project - Fundamentals of AI**

**Course Number:** EAI 6000

**Academic Term:** Fall 2020 CPS Analytics  
**Instructor’s Name:** Kasun Samarasinghe

**Assignment Completion Date:** 12-06-2020

**Submitted By:**

Dhiren Vasudev Pagrani

Sunil Raj Thota

Shivani Sharma



**INTRODUCTION**

This is our group assignment and we have selected the data set from Kaggle based on the medical research conducted by MIT and Harvard which is serving the purpose of developing a drug advancement through enhancements to “Mechanisms of Action (MoA)” prediction algorithms.

As the discovery of new drugs has evolved, the technology of innovation has also improvised. As of now, researchers look to recognize a protein target related to disease and build up a molecule that can regulate that protein target. As a shorthand to depict the biological movement of a given molecule, scientists allot a mark alluded to as Mechanism of Action or MoA for short.

Previously, researchers got drugs from normal items or were propelled by customary cures. Extremely basic medications, for example, paracetamol, referred to in the US as acetaminophen, were placed into clinical use a very long time before the organic instruments driving their pharmacological exercises were perceived. Today, with the appearance of all the more impressive advancements, drug revelation has transformed from the fortunate methodologies of the past to a more focused on model dependent on a comprehension of the hidden organic component of a sickness. In this new structure, researchers try to recognize a protein target related with an illness and build up a particle that can adjust that protein target. As a shorthand to depict the organic action of a given atom, researchers allot a name alluded to as instrument of-activity or MoA for short.

**BUSINESS PROBLEM**

Our goal is to improvise the algorithm already implemented by MIT and Harvard which provides the classification of drugs based on the biological activities happening as a result of medicines with the body cells and predict molecular targets from that data using an integrated preprocessing and machine learning pipeline.

WE have decided a approach with this dataset that joins quality articulation and cell feasibility information. The information depends on another innovation that measures at the same time (inside similar examples) human cells' reactions to drugs in a pool of 100 diverse cell types (hence taking care of the issue of distinguishing ex-bet, which cell types are more qualified for a given medication). Likewise, you will approach MoA comments for in excess of 5,000 medications in this dataset.

As is standard, the dataset has been part into testing and preparing subsets. Consequently, your errand is to utilize the preparation dataset to build up a calculation that naturally marks each case in the test set as at least one MoA classes. Note that since medications can have different MoA comments, the assignment is officially a multi-mark grouping issue.

This report summarizes all the models and techniques we learned during the class in the last 6 weeks and its implications on the model.

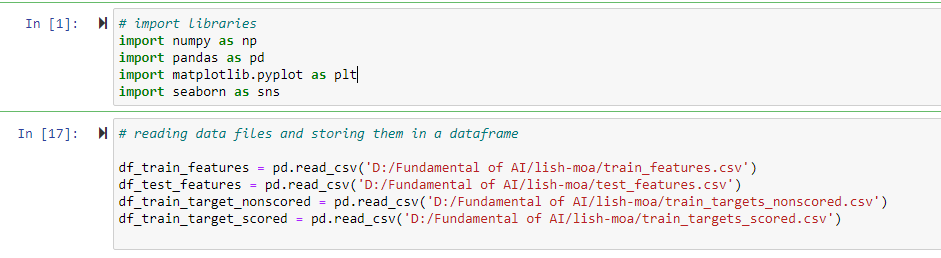
**ANALYSIS**

We would like to start our analysis part with EDA and then we will explain all the algorithms with analysis individually in each section. In light of the MoA comments, the accuracy of solutions will be evaluated on the average value of the logarithmic loss function applied to each drug-MoA annotation pair.

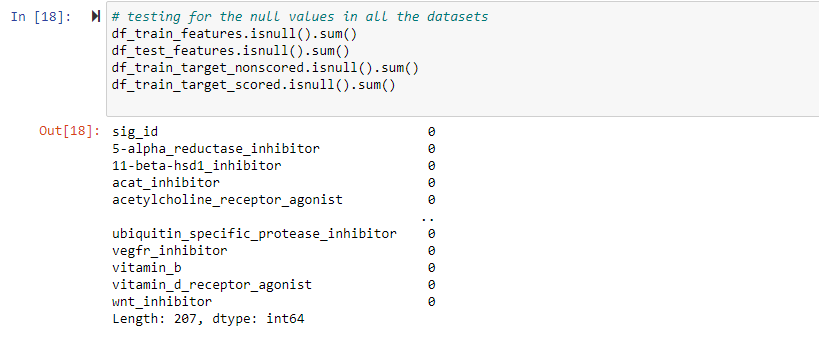
In nut shell we have to design a algorithm to anticipate a compound's MoA given its cell signature, accordingly assisting researchers with propelling the drug discovery process.

**EDA**

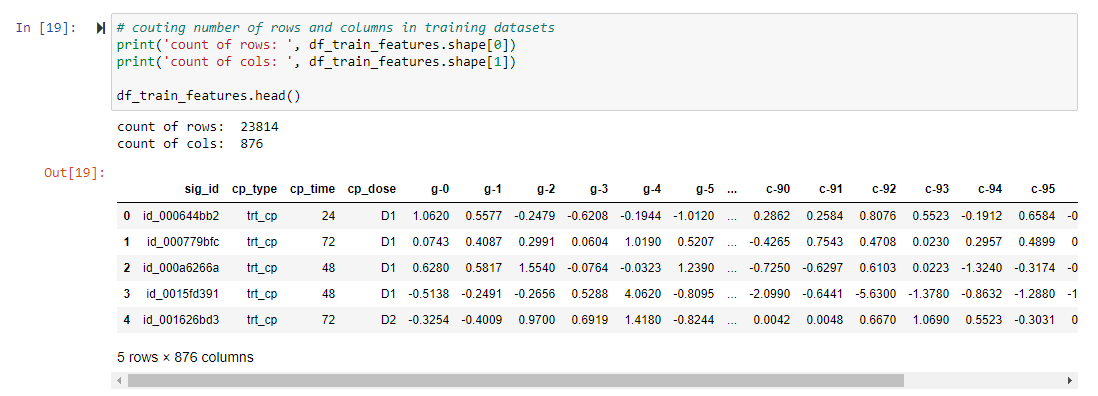
In order to do a kick off we decided to start from the very basic step i.e finding the null values and fixing them . Below is the detailed analysis we did as a part of our exploratory data anlaysis.

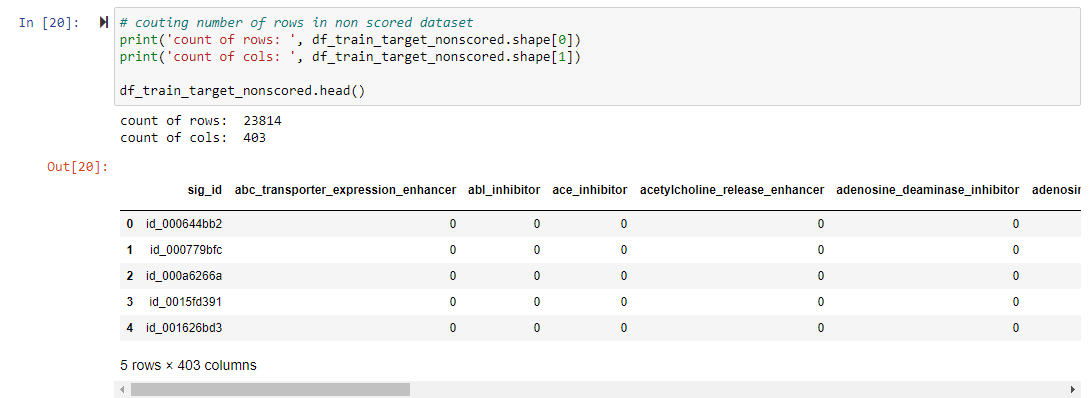


Checking the dataset for the presence of null values:



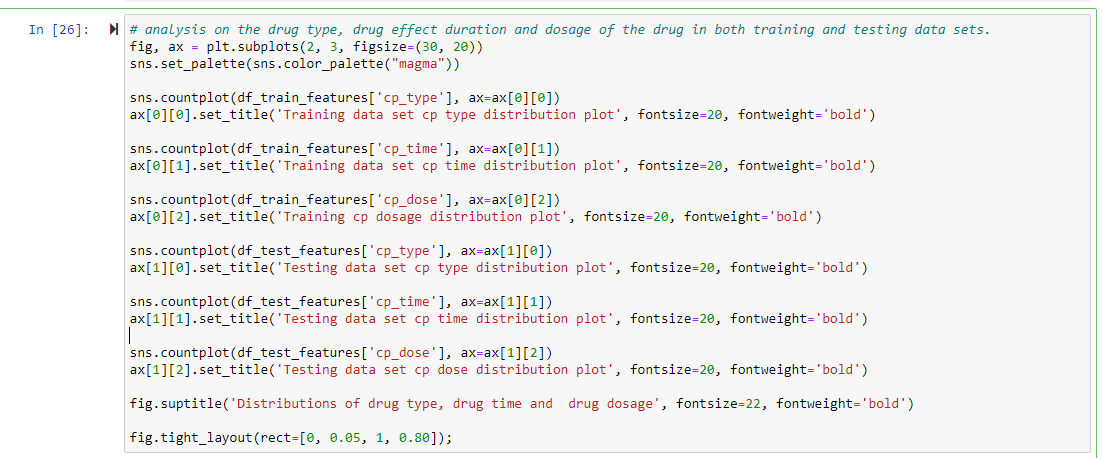
Counting the number of rows and columns in the dataset before starting the analysis part:

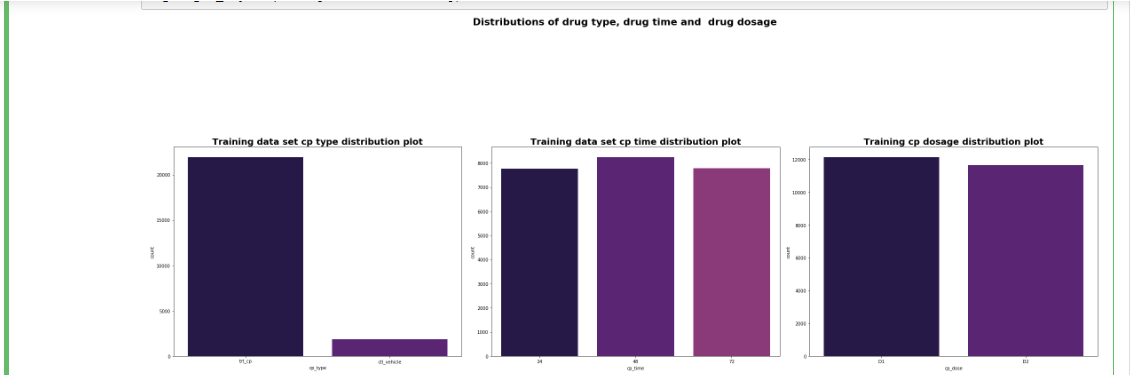


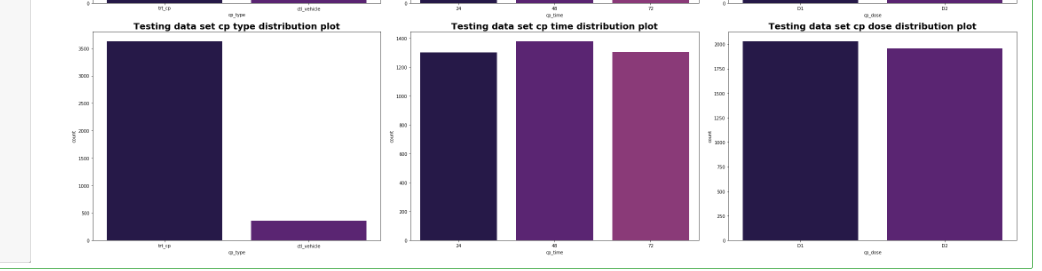




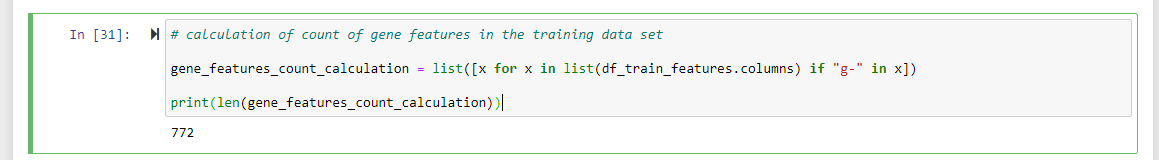
Now we have analyzed with the help of visualization we created to understand the pattern and relationship between drug type, duration, and dosage of drugs.

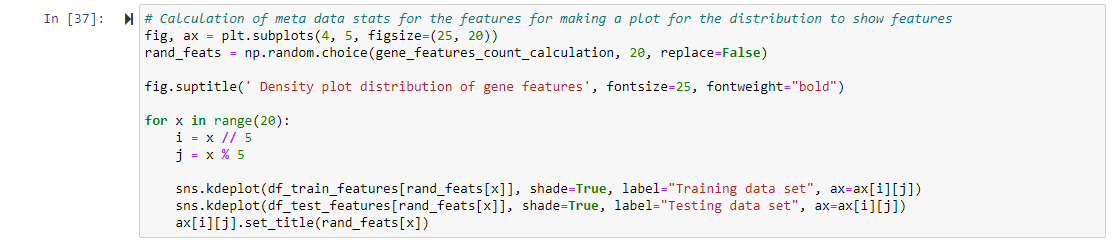


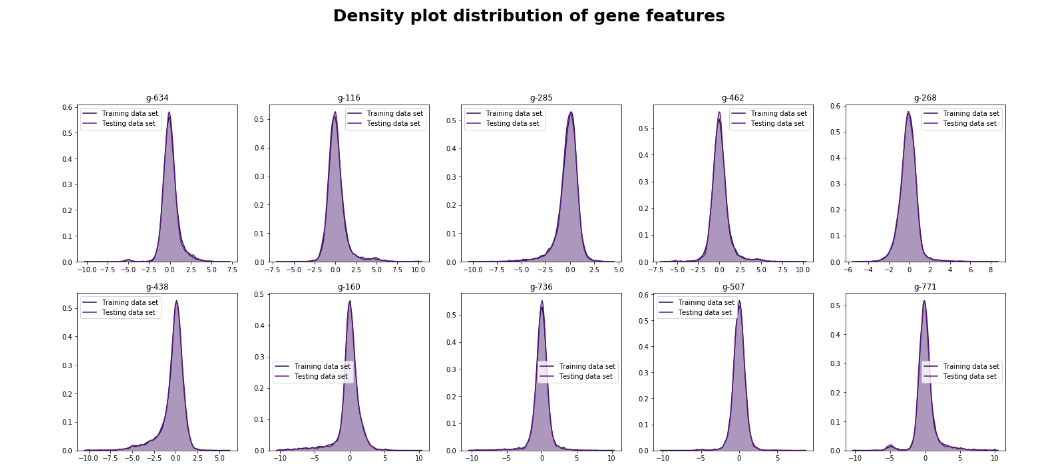


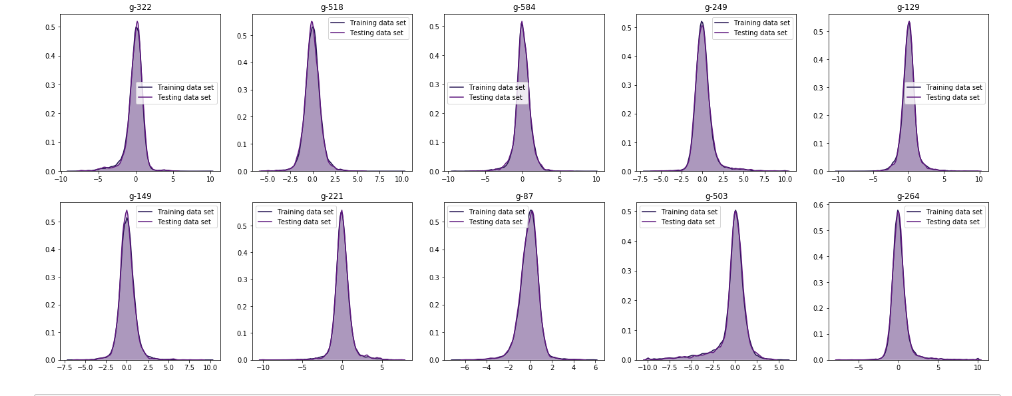


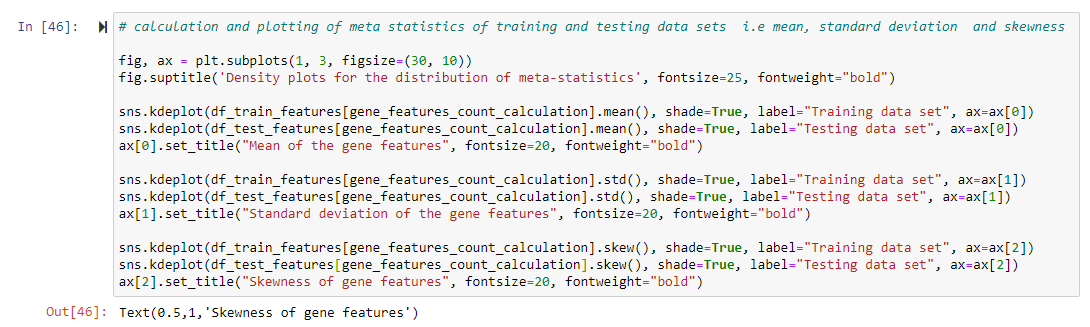
Now we decided to deep dive into the gene and cell data factors provided in the data set and here is the analysis we did for that:

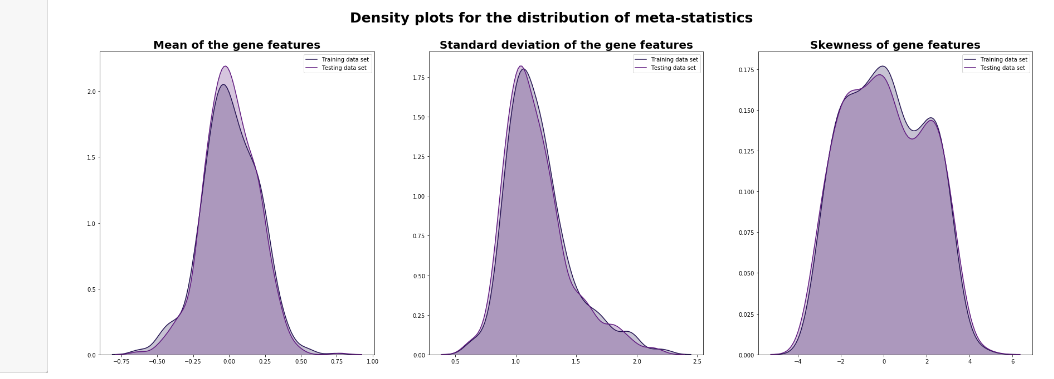






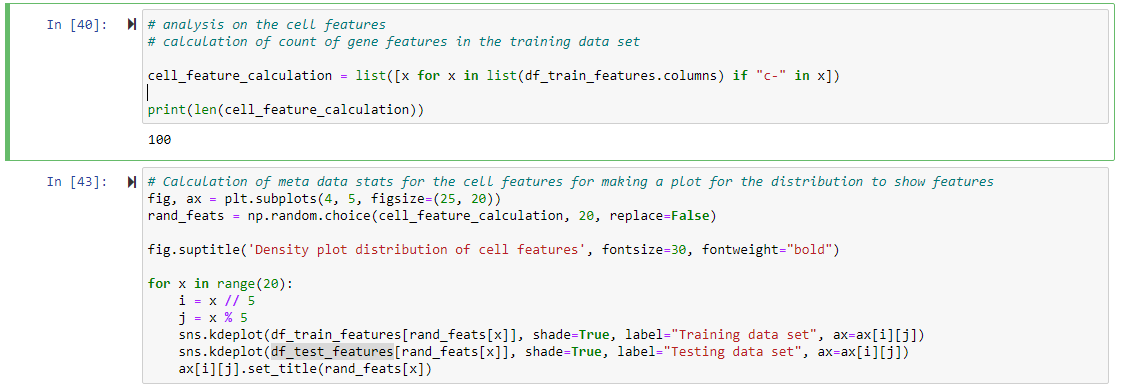


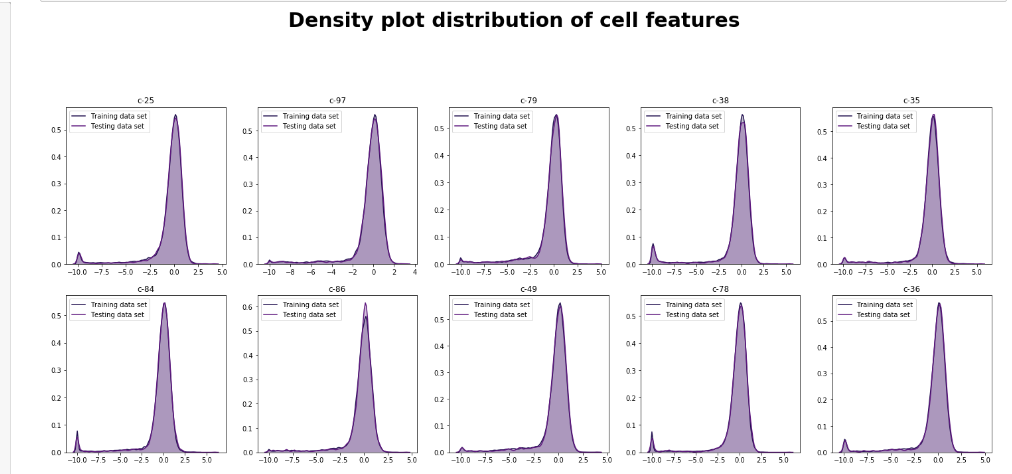


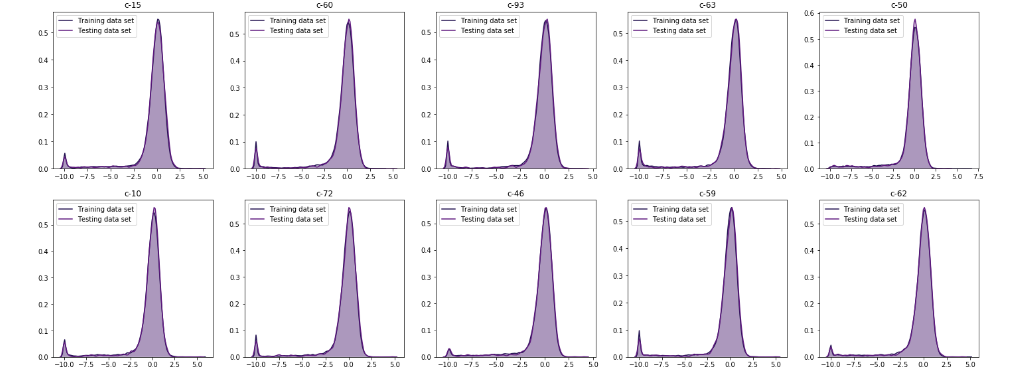


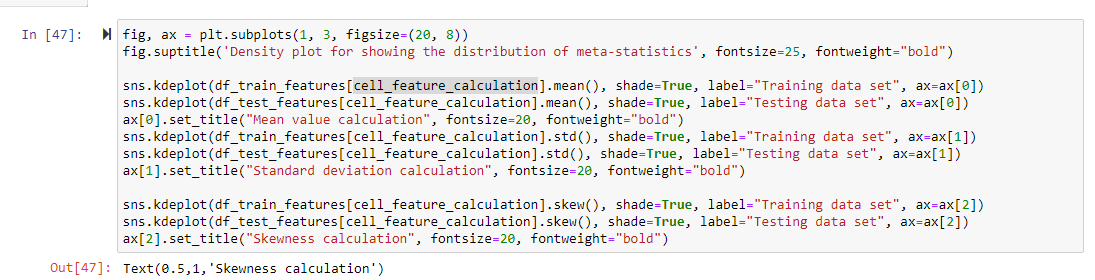
From the above plots, we can infer that mostly feature distributions are centered at 0 and mostly are symmetric, and very few are right or left-skewed.

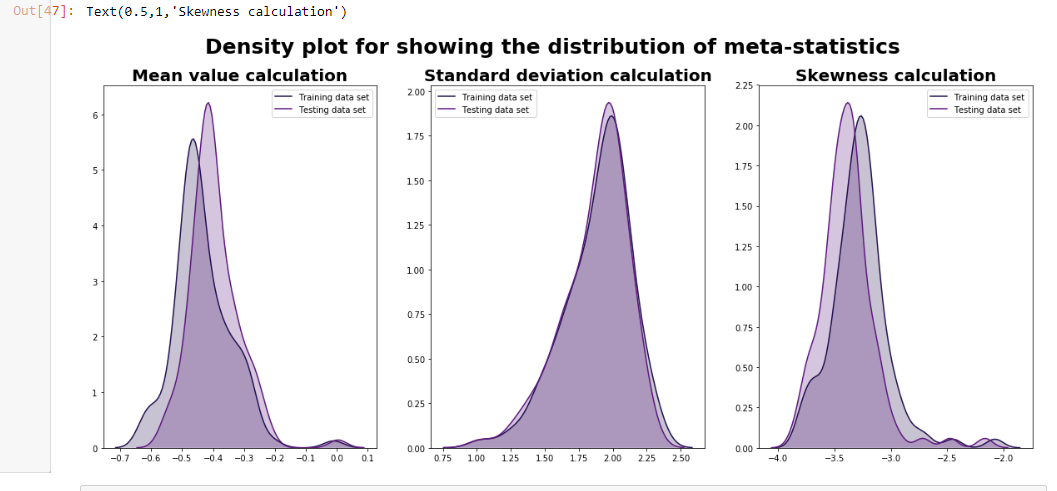
Now following is the analysis we did for the data provided for the gene and cell features collection for each test conducted.

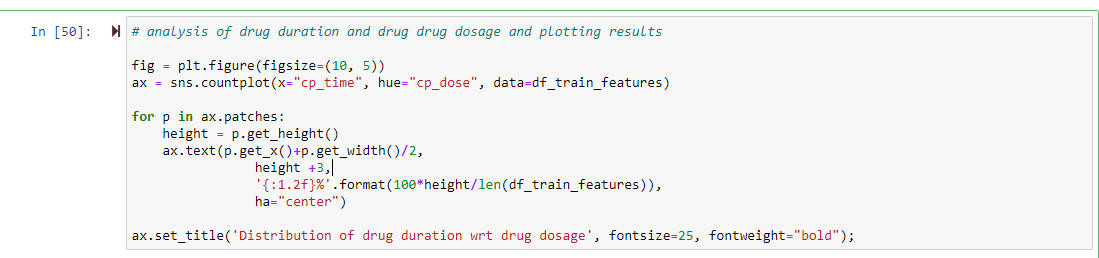


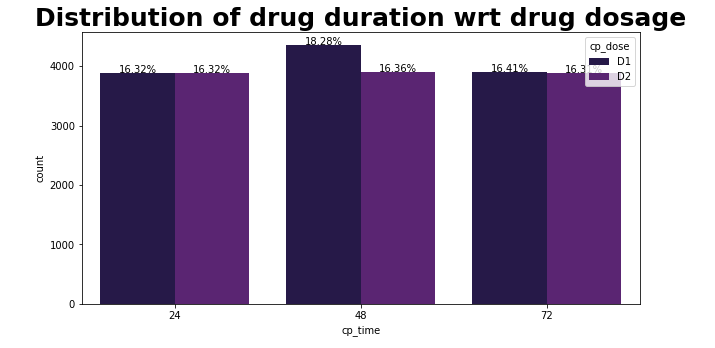


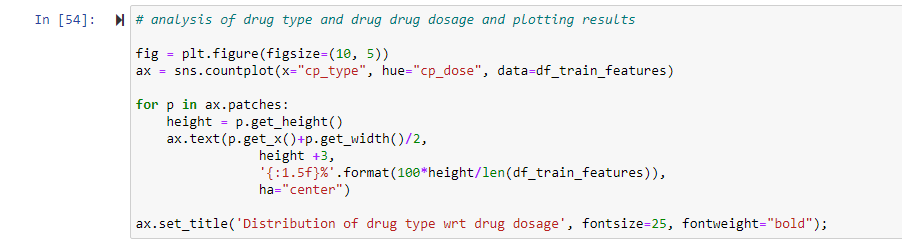


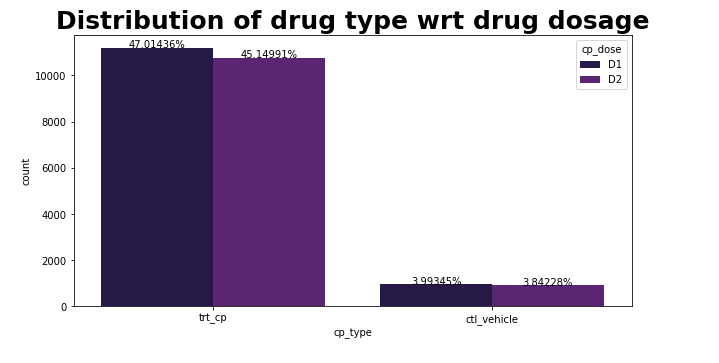


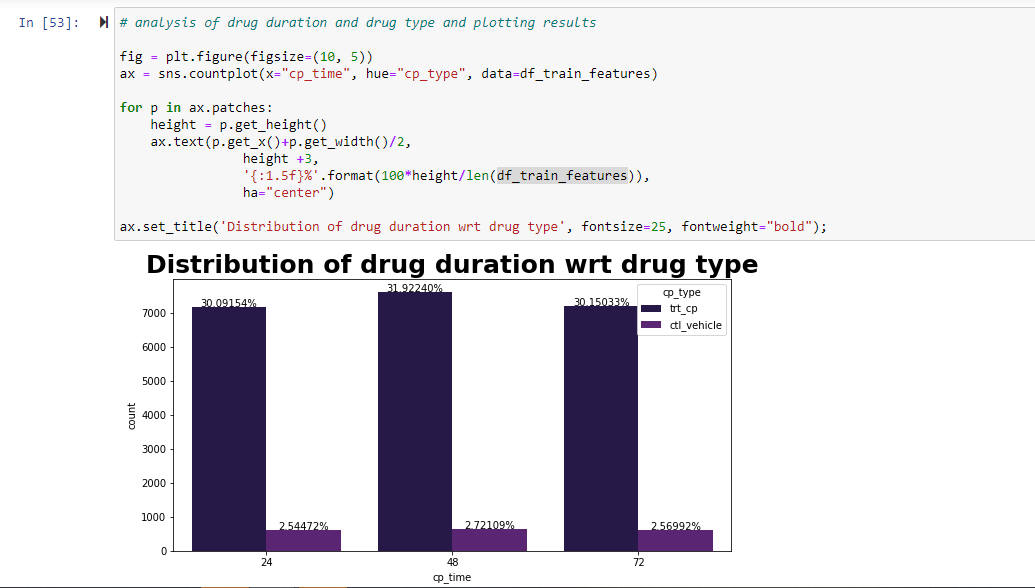






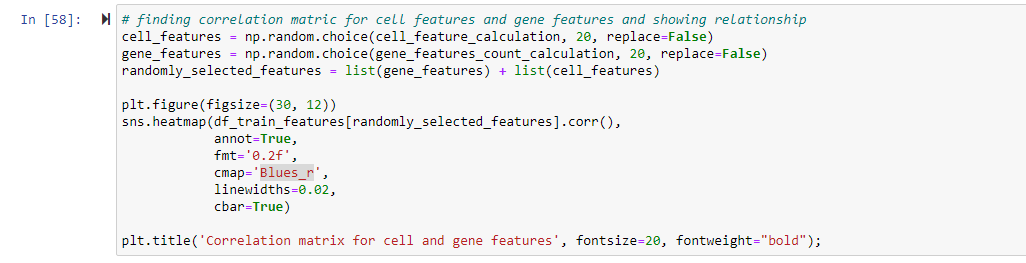


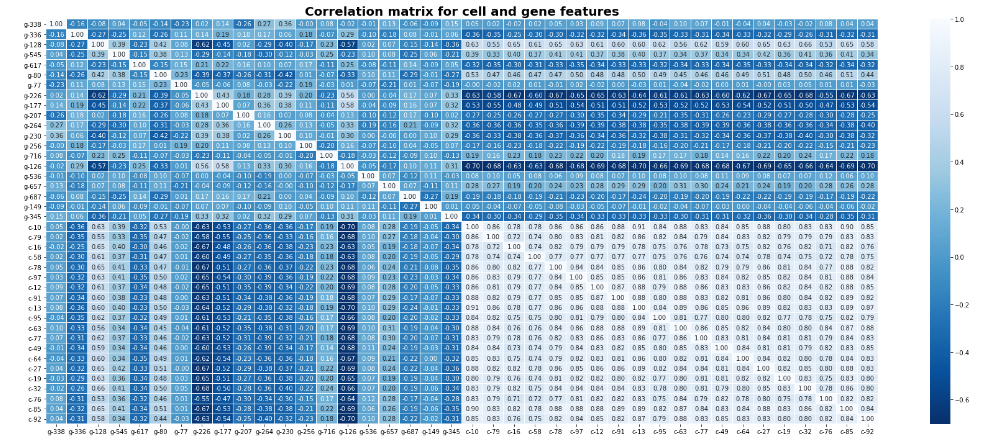


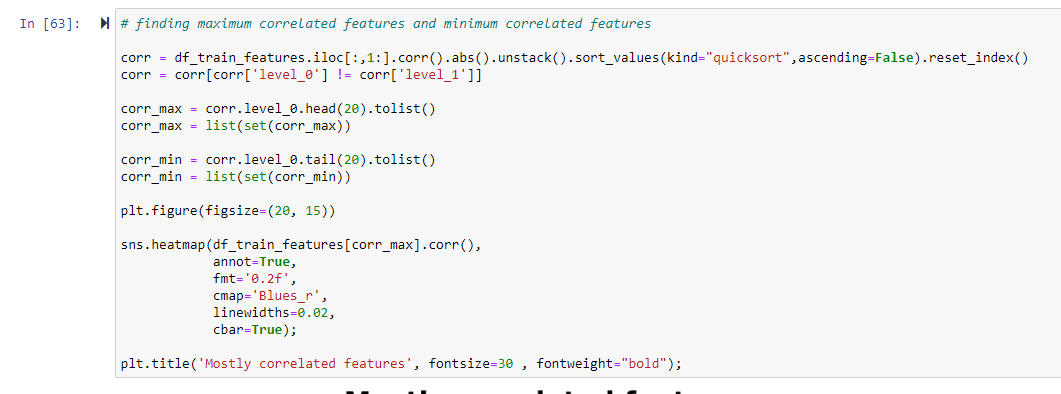


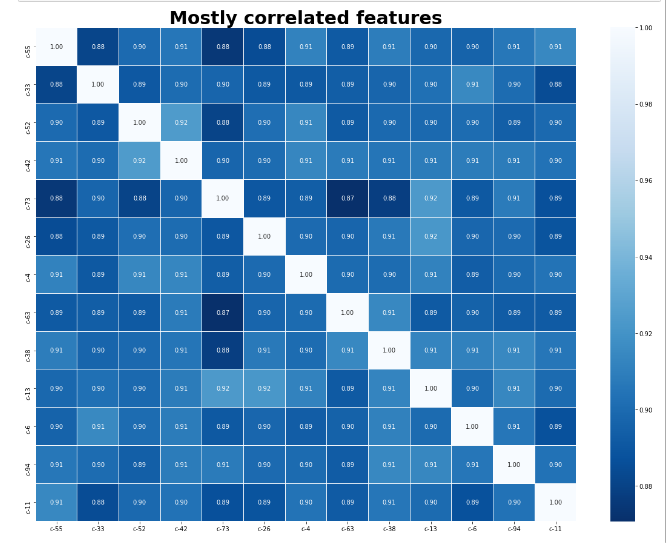
From the above plots, we can infer that the drug type and its effect of duration are having a significant difference between each other.

Now we decided to find out the correlation for all the attributes and then we have filtered out the most significant features by using the correlation matrix









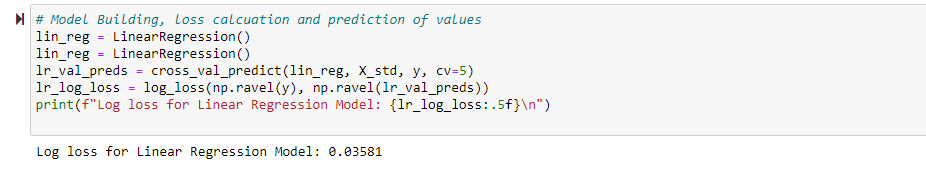
**Linear Regression**

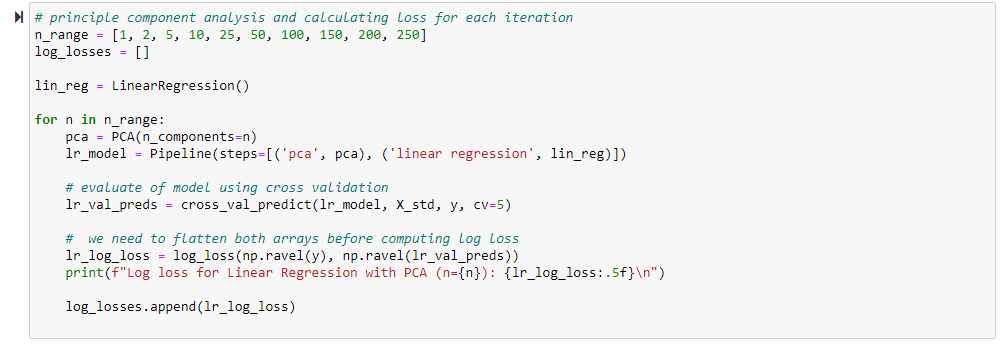
Linear Regression is an essential and ordinarily utilized kind of prescient examination. The general thought of relapse is to look at two things: (1) does a bunch of indicator factors work superbly in anticipating a result (subordinate) variable? (2) Which factors specifically are critical indicators of the result variable, and how would they–showed by the extent and indication of the beta assessments sway the result variable? These relapse gauges are utilized to clarify the connection between one ward variable and at least one free factors. The least difficult type of the relapse condition with one reliant and one autonomous variable is characterized by the equation y = c + b\*x, where y = assessed subordinate variable score, c = consistent, b = relapse coefficient, and x = score on the free factor.

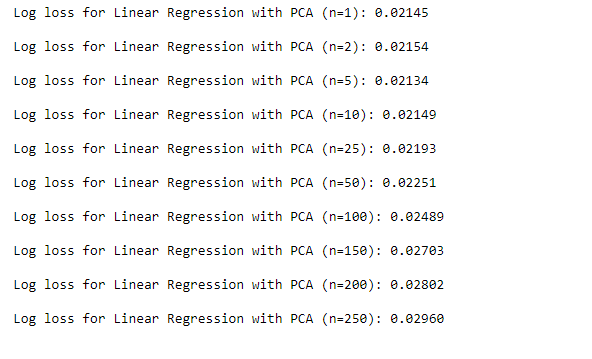
Naming the Variables. There are numerous names for a relapse's reliant variable. It could be called a result variable, basis variable, endogenous variable, or regressand. The free factors can be called exogenous factors, indicator factors, or regressors.

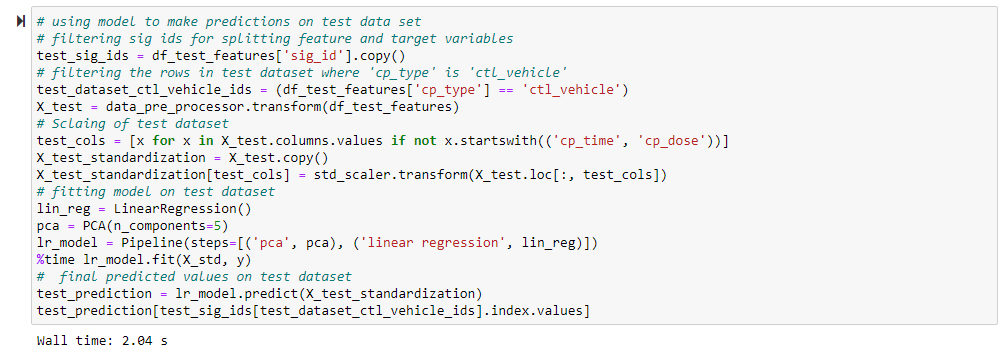
Three significant uses for relapse examination are (1) deciding the strength of indicators, (2) anticipating an impact, and (3) pattern estimating.

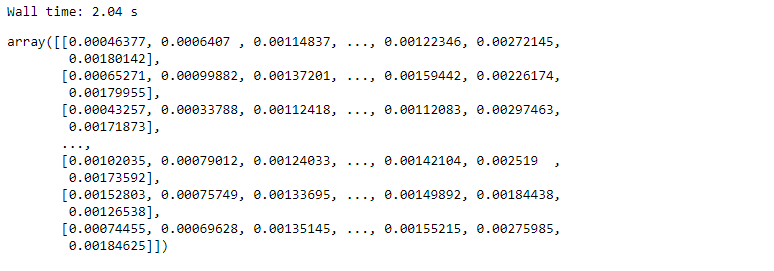
Here is the analysis we did as a part of linear regression model on our dataset:









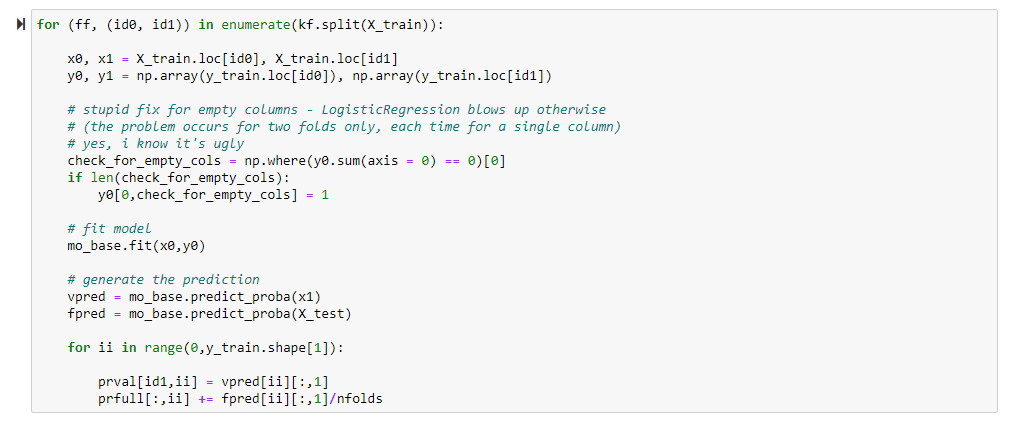


**Logistic Regression**

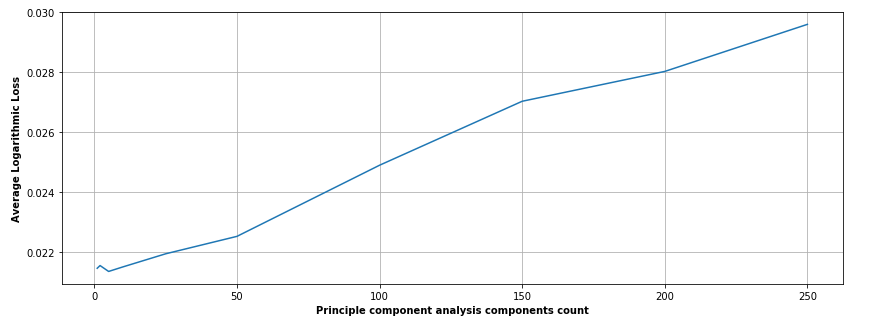
It is a method to examine an informational collection which has a needy variable and at least one autonomous factors to foresee the result in a paired variable, which means it will have just two results. The reliant variable is unmitigated in nature. Subordinate variable is additionally alluded as target variable and the autonomous factors are known as the indicators. Logistic Regression is a unique instance of direct relapse where we just anticipate the result in an absolute variable. It predicts the likelihood of the occasion utilizing the log work. We utilize the Sigmoid capacity/bend to foresee the straight out worth. The edge esteem chooses the outcome(win/lose).

Now coming to analysis part of logistic model:



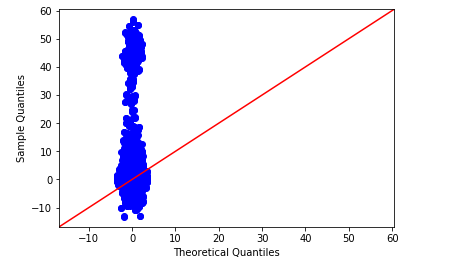


Following are the visualizations created for the linear and logistic regression models.

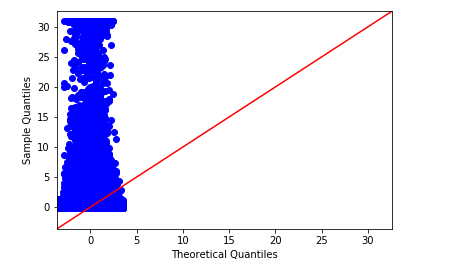


From the above curve we can see that there is little non linearity in the data set and hence we build logistic on top of this model in order to reduce it.

Below is the qq plot representation for the linear model.



QQ plot for logistic regression model is as follows:



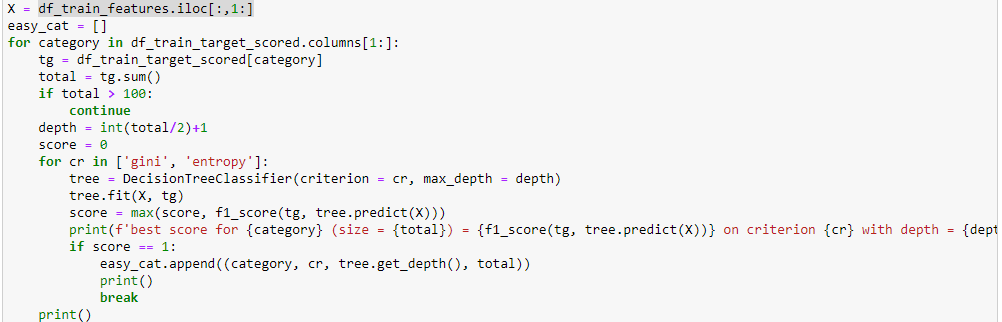
**Decision Tree**

Decision Trees are made by an algorithmic strategy that arranges approaches to part an informational index dependent on different conditions. It is one of the most broadly utilized and true methodologies for managed learning.

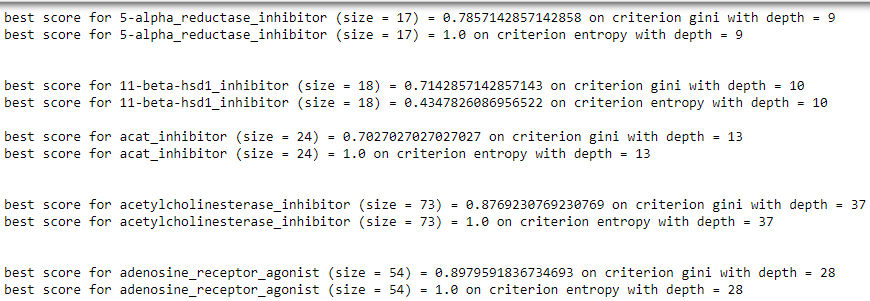
These are a non-parametric regulated learning technique utilized for both order and relapse errands. The objective is to create a model that visualizes the estimation of an objective variable by taking in straightforward choice standards gathered from the information highlights. The standards are commonly in type of on the off chance that else explanations. More profound the tree, the more mind boggling the guidelines and fitter the model.

The 2 most popular decision tree’s decisions are Gini Index and Information Entropy. The gini score is always the same because they always add to 0 in the formula above. A gini score of 0 is the purest score possible.

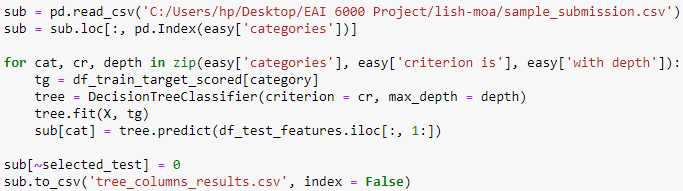
Entropy is more computationally heavy due to the log in the equation. Like Gini, the basic idea is to measure the condition of a grouping by the target variable. Instead of utilizing simple probabilities, this method takes the log base2 of the probabilities.



Decision Trees are created by an algorithmic method that classifies ways to split a data set based on various conditions. It is one of the most extensively used and real-world approaches for supervised learning.

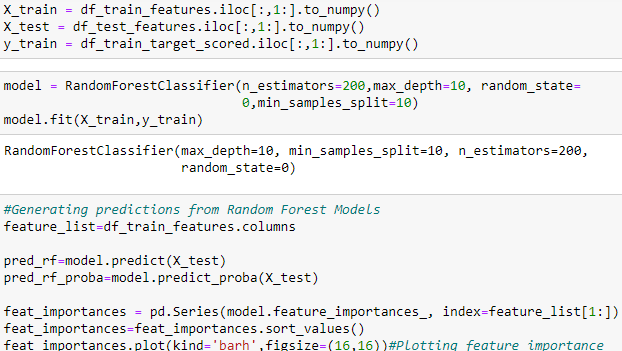


Categories, positive samples in dataset, criterion is, and with depth are called as ct, s, cr, and d.



These are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to produce a model that envisages the value of a target variable by learning simple decision rules inferred from the data features. The rules are generally in form of if-then-else statements. Deeper the tree, the more complex the rules and fitter the model.

From this we have obtained the predicted values and stored as tree\_columns\_results.csv file



**Random Forest**

Random Forest is a tree-based machine learning algorithm that leverages the facility of multiple decision trees for creating decisions. It's a forest of randomly created decision trees. Each node within the decision tree works on a random subset of features to calculate the output.

Random Forest is a method used in modeling predictions and behavior analysis and is built on decision trees. It contains many decision trees that signify a discrete instance of the classification of data input into the random forest. This technique takes consideration of the instances individually, taking the one with the majority of votes as the selected prediction. Each tree in the classifications takes input from samples in the initial dataset. Features are then randomly selected. In such a way, it enables any classifiers with weak correlations to create a strong classifier.

The random forest associates hundreds or thousands of decision trees, trains each one on a somewhat diverse set of the observations, segregating nodes in each tree considering a limited number of the features. The final outcomes of the random forest are made by averaging the predictions of each distinct tree.

Here’s the plot between list of features and following things are observable from the plot above - The first principal components for c- and g- variables have the highest feature importance.



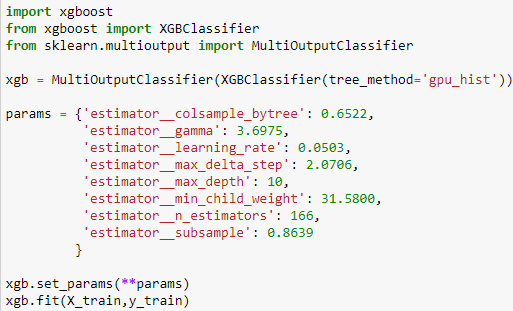
As observed in our EDA - dosage, treatment type and duration of treatment have the lowest feature importance in the dataset. c- and g- PCs solely contribute to the performance of the model.

**XG Boost Algorithm**

XGBoost is a calculation that has as of late been overwhelming applied AI and Kaggle rivalries for organized or even information. XGBoost is an execution of inclination supported choice trees intended for speed and execution. The execution of the calculation was designed for productivity of register time and memory assets. A plan objective was to utilize accessible assets to prepare the model. Some key calculation usage highlights include:

1. Sparse Aware execution with programmed treatment of missing information esteems.
2. Block Structure to help the parallelization of tree development.
3. Continued Training with the goal that you can additionally help a generally fitted model on new information.

In XGBoost, we fit a model on the gradient of loss generated from the previous step. In XGBoost, we just modified our gradient boosting algorithm so that it works with any differentiable loss function.



**Advantages of XG Boosting:**

* Fast to interpret and Outliers have minimal impact.
* Handles large sized datasets well and Less prone to overfitting
* Good Execution speed and Good model performance

**Disadvantages of XG Boosting:**

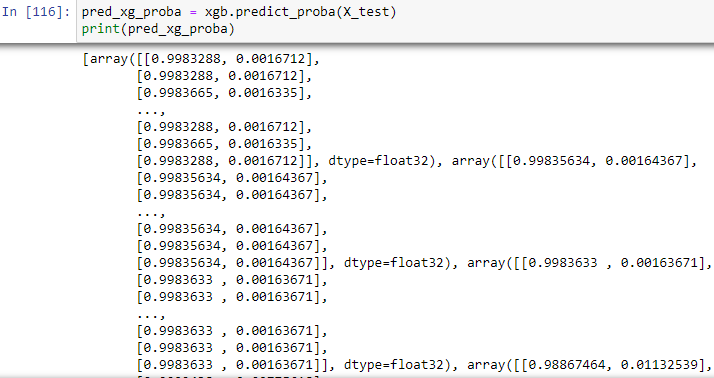
* Difficult interpretation
* Overfitting possible if parameters not tuned properly.
* Visualization is tough
* Harder to tune as there are too many hyperparameters.



Bagging and boosting are two widely used ensemble learners. Though these two techniques can be used with several statistical models, the most predominant usage has been with decision trees.

Here we can see the predicted values by the XG Boosting model.

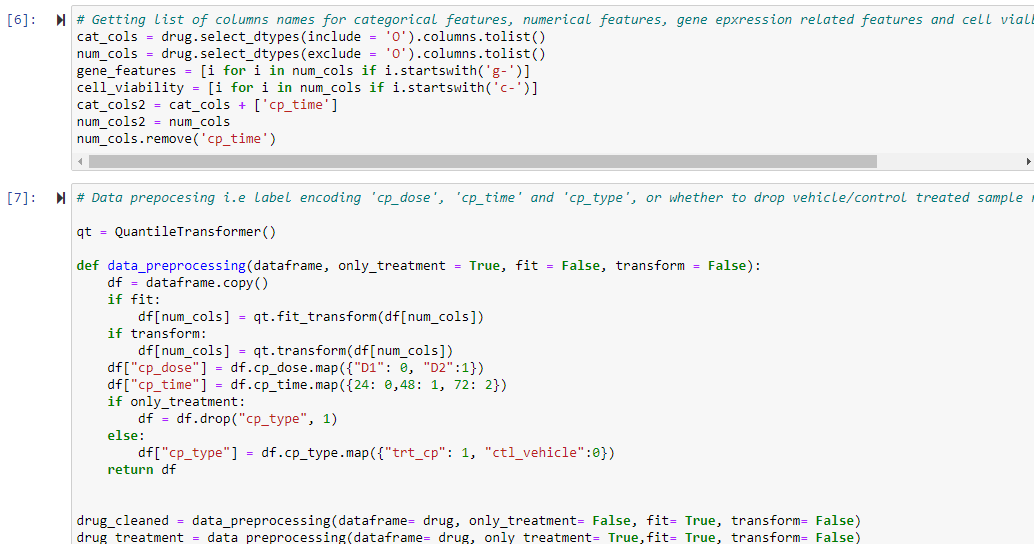
Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

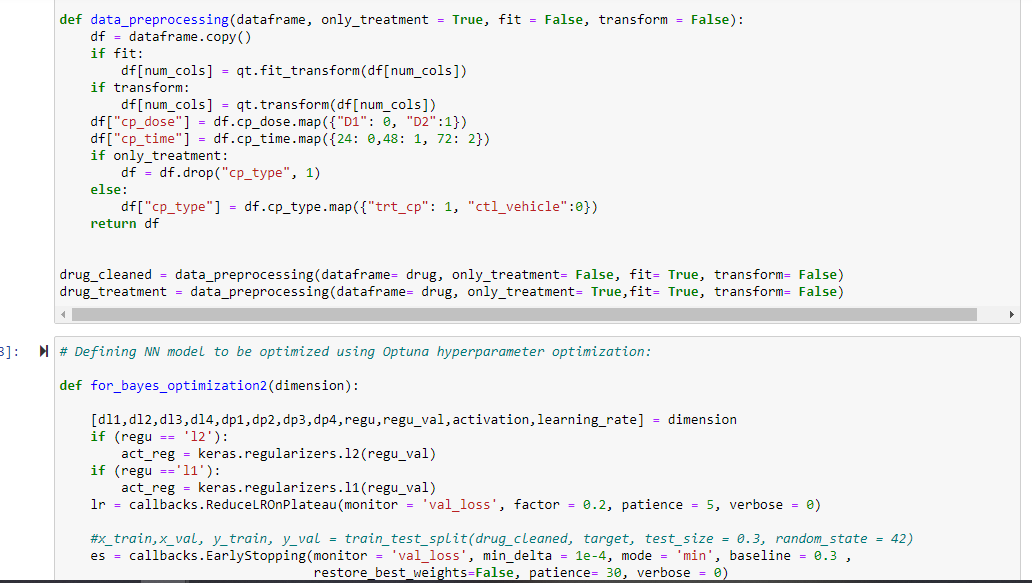


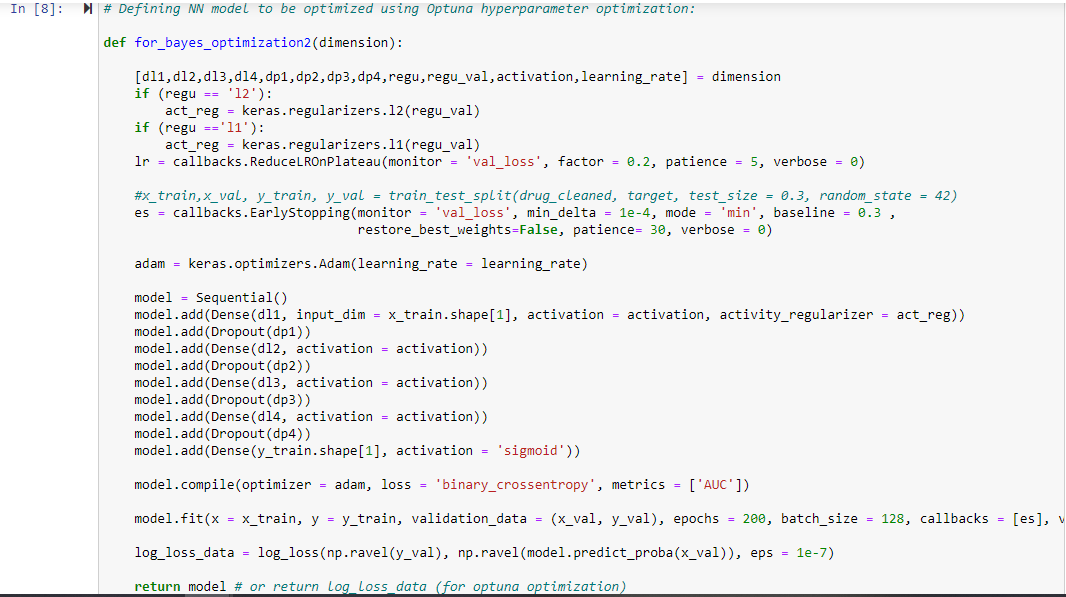
We first model data with simple models and analyze data for errors. These errors signify data points that are difficult to fit by a simple model. Then for later models, we particularly focus on those hard to fit data to get them right. In the end, we combine all the predictors by giving some weights to each predictor.

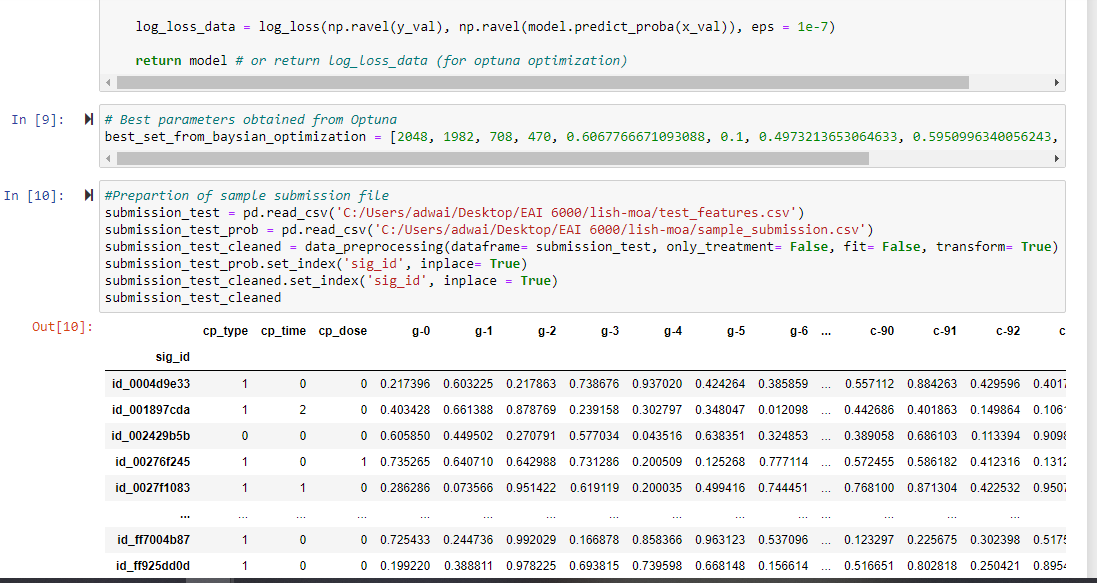
Since from the above approach we were not getting good results for the final file so we did hyper parameter tuning and applied **Bayesian Optimization for the same.**

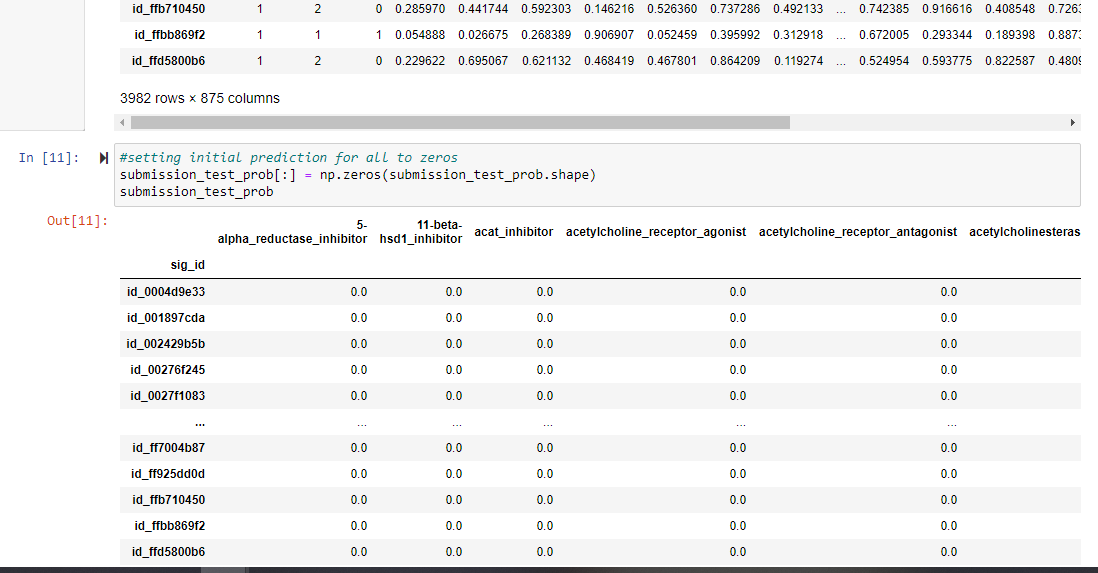


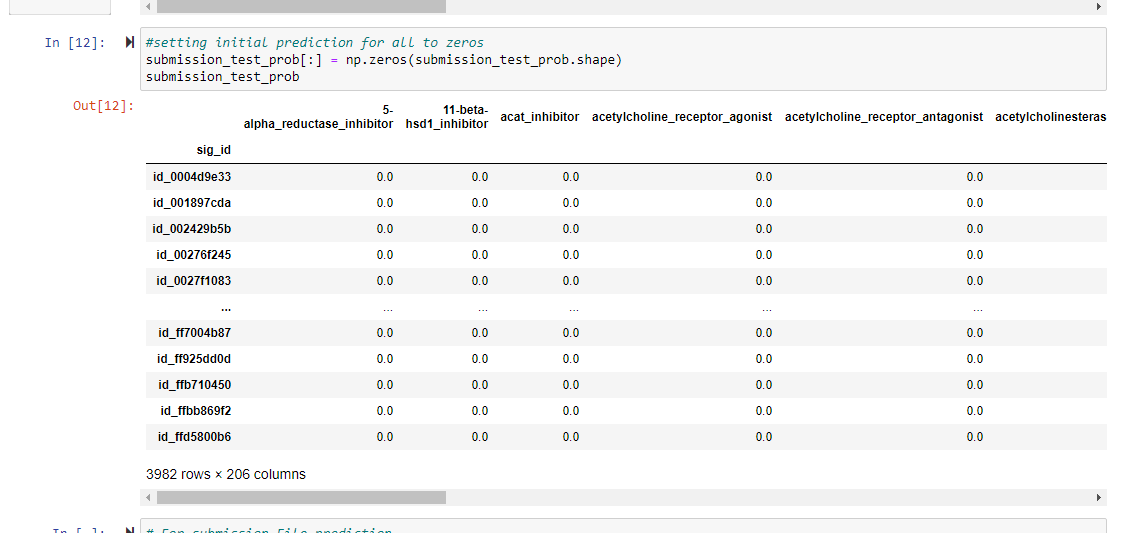




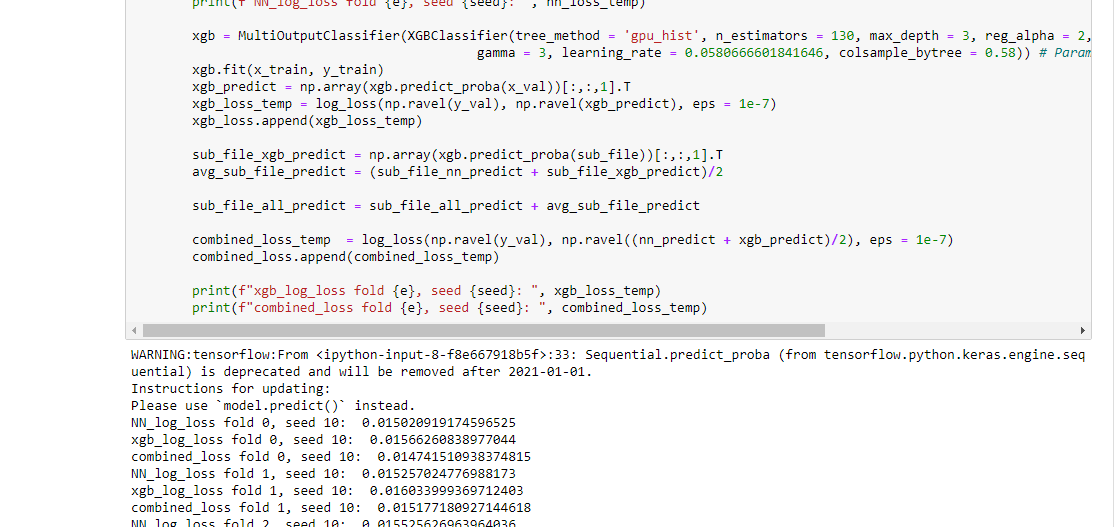


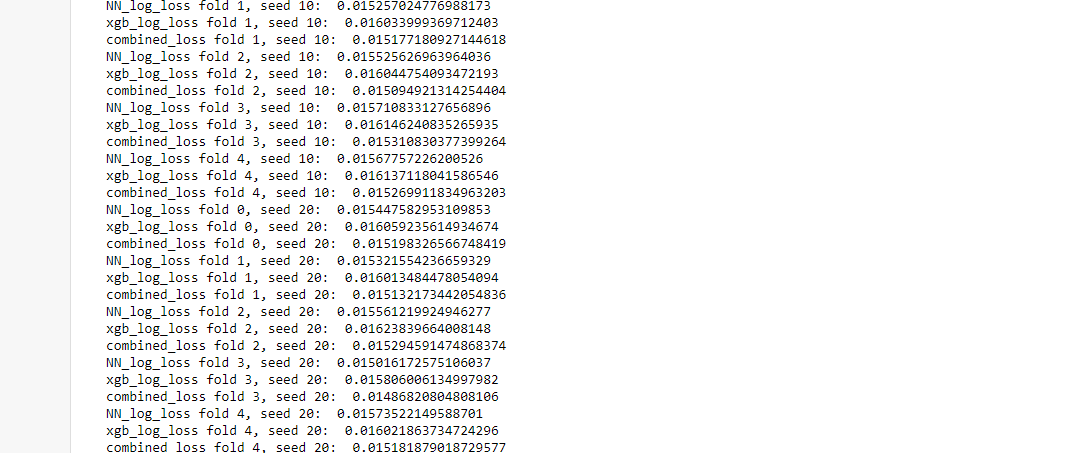












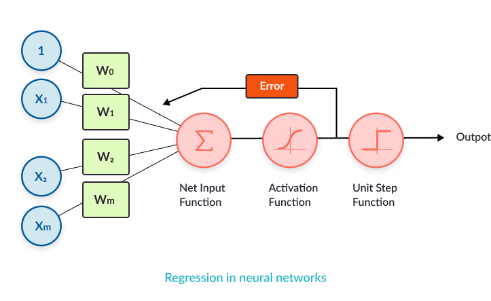






**Neural Networks**

Neural organizations are reducible to regression models - a neural organization can "pretend" to be any sort of regression model. It takes several dependent factors = input boundaries, multiplies them by their coefficients = weights, and runs them through a sigmoid initiation work and a unit step function, which intently looks like the calculated relapse work with its mistake term.

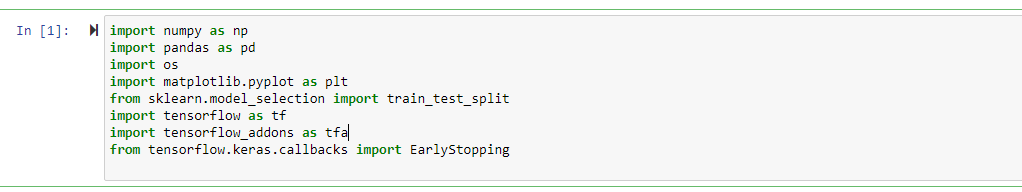


The logistic regression we modeled above is reasonable for binary classification. Imagine a scenario where we have to demonstrate multi-class grouping. We can build the intricacy of the model by utilizing various neurons in the hidden layer, to accomplish one-versus all order. Every order choice can be encoded utilizing three parallel digits.

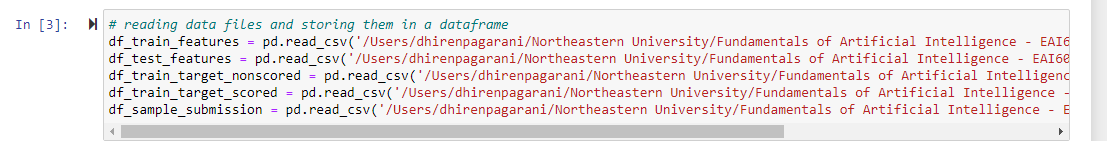
**Pros and Cons using Neural Networks as opposed to other ML models**

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| * Neural organizations are adaptable and can be utilized for both regression and classification issues * Neural networks are acceptable to show with nonlinear information with an enormous number of contributions; for example, pictures. * When prepared, the expectations are pretty quick. * Neural organizations can be prepared with quite a few sources of info and layers * Neural organizations work best with more information focuses | * Neural networks are black boxes, which means we can't realize how much every autonomous variable is impacting the dependent variables. * It is computationally over the top expensive and tedious to prepare with customary CPUs * These rely a ton upon preparing information. * This prompts the issue of over-fitting and speculation. * The mode depends more on the preparation information and might be tuned to the information. |

Here is the analysis we did for implementing neural network on our dataset:



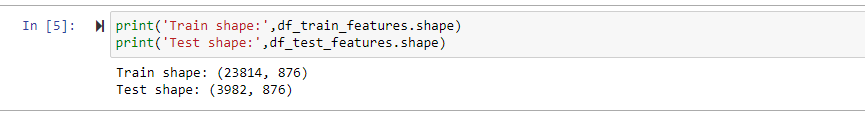
Following is the logic for reading the csv files and storing them in a data frame.



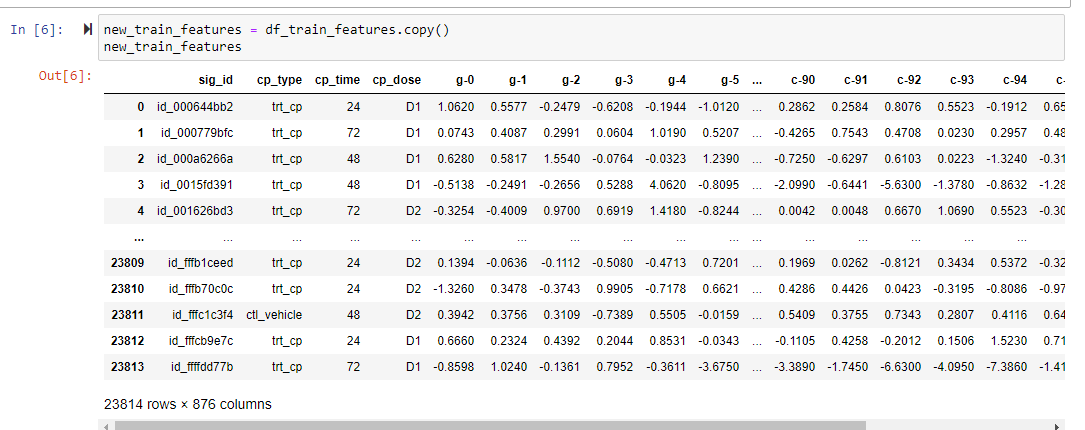
Following is the logic for defining the seeds



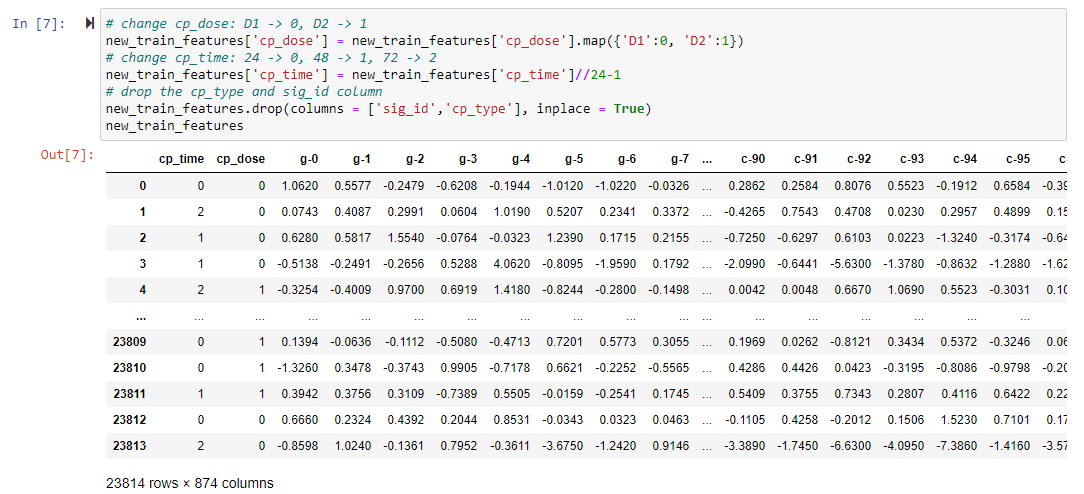
Following is the check for verifying the training and testing datasets size:



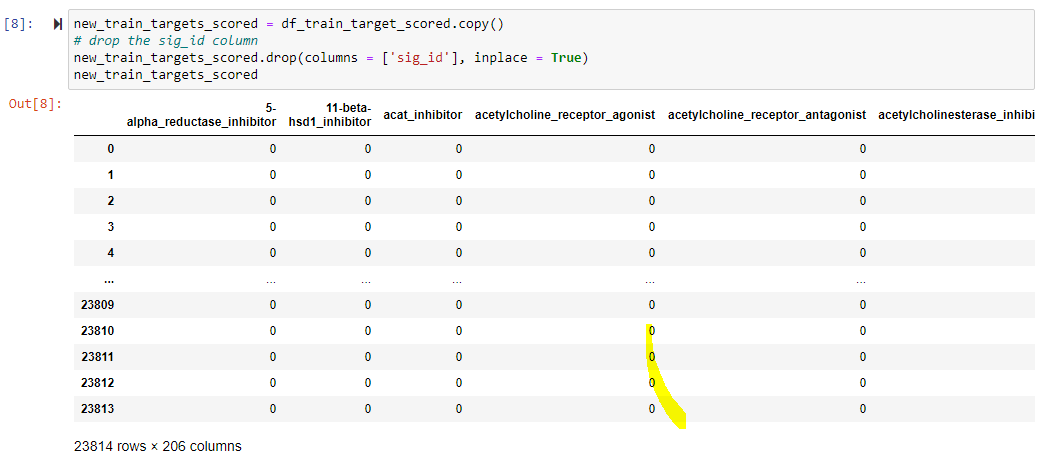
For building the model we decided to copy the data frame rather than modifying the dataset in order to feed the model with it and here is the logic for the same:



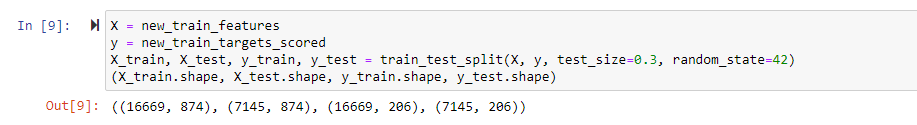
Now pre-processing the dataset:



Copying the target scored dataset and then processing the target scored dataset:



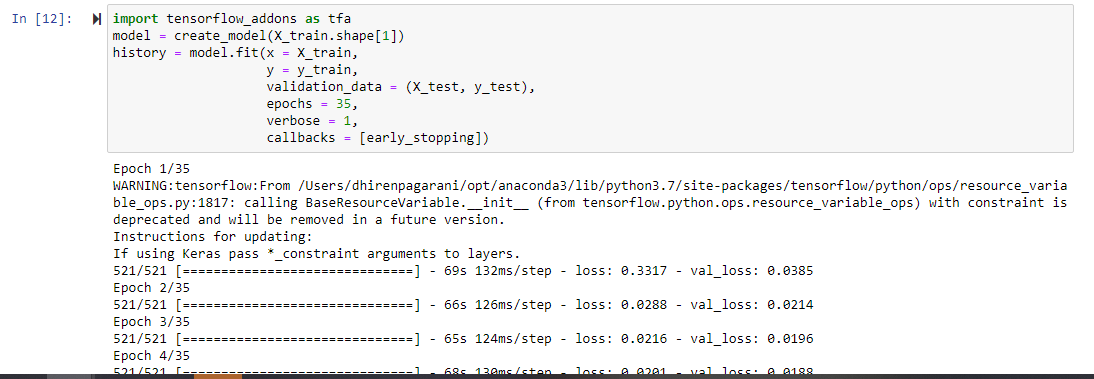
Storing the data into X and y variables in order to read target variables and label values.



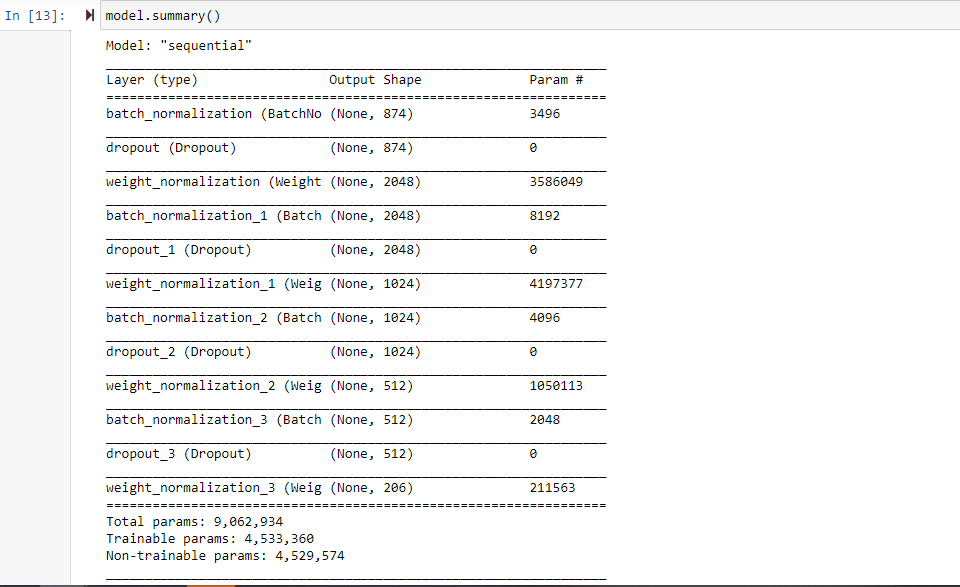
Now we have to build a model and here is the logic for the same: We decided to use early stopping callback approach as we were not getting good results with other approaches and then build sequential model on top of it with 3 layers on top of sigmoid layer. Compiling it using adam optimizer and ‘binary cross entropy’ loss.



Now we will finally build the model and fit it one the dataset for epoch count 35.



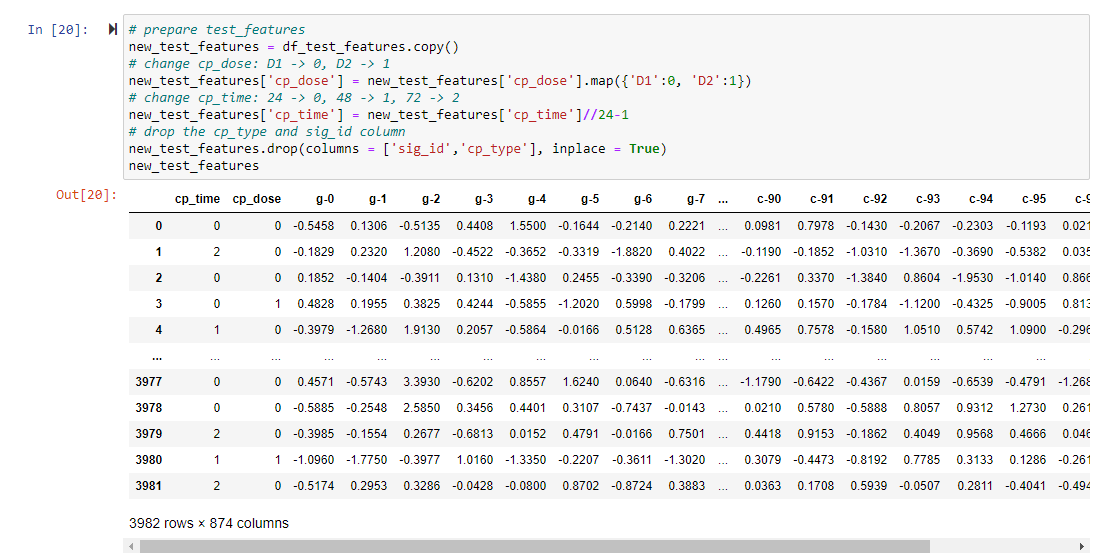
Here is the summary of our model:



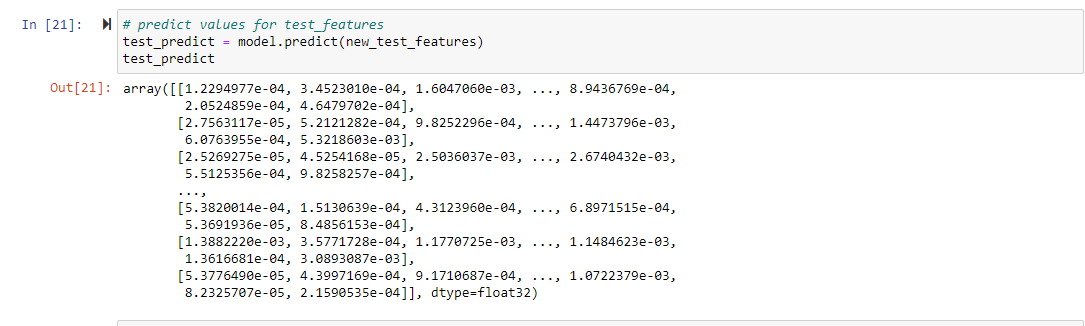
Here is the visualization we did for our model for displaying validation loss and training loss



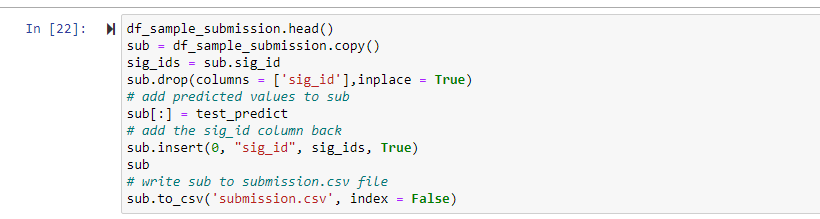
Now before making predictions we will again make copy of test dataset and do the preprocessing before feeding it into the model:

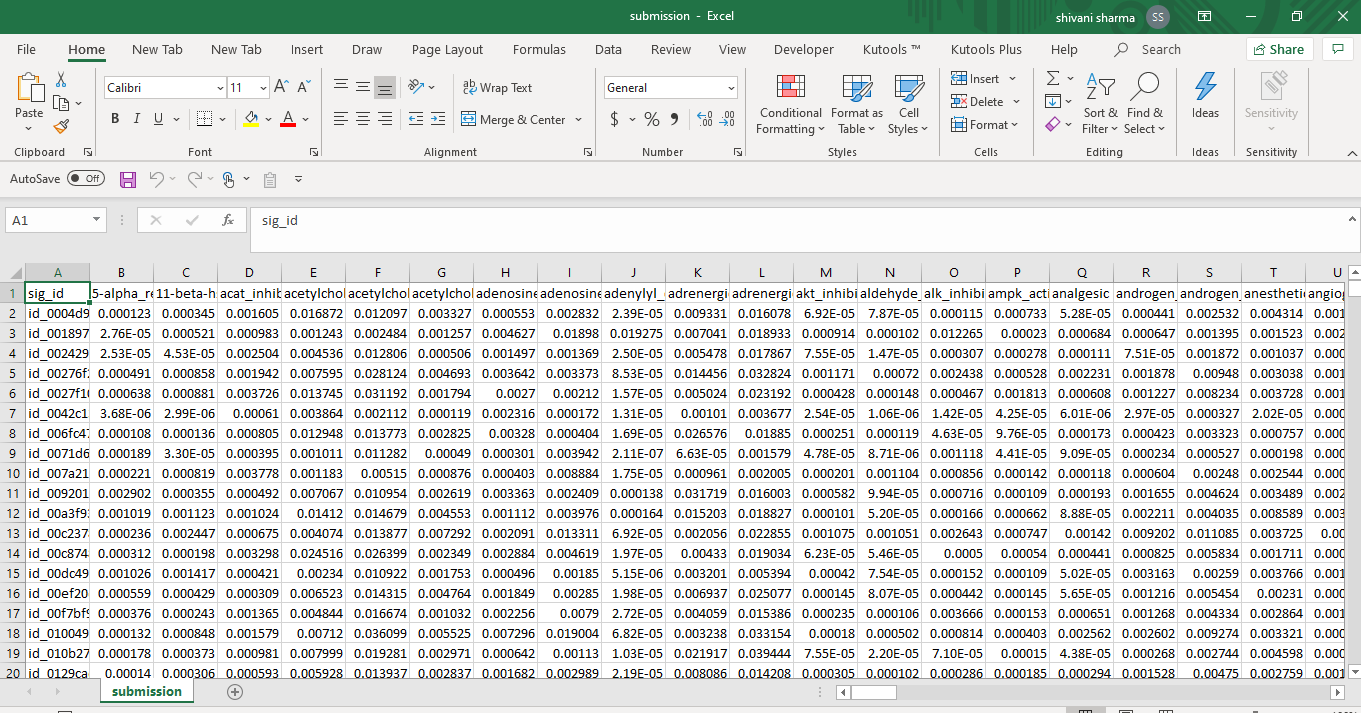


Now finally making predictions :



Finally, here is the predicted file that we received in the form of probabilities and the results are pretty good as compared to previously build algorithms.





**CONCLUSION**

After applying all the models i.e linear, logistic, random forest, decision tree classifier, XG Boost and Neural network from scratch we concluded that the log loss value is less for neural network as compared to other models. Also, the value obtained with XG Boost after applying hyper parameter tuning is competitive to neural network. We reached to this conclusion for neural network based on the elbow curve we plotted which is showing the value of 0.01 and with xg boost it is 0.015. Also, the accuracy of neural network is coming out as 79.54% which is more as compared to other models. Accuracy with XG boost is also competitive i.e 76.38%. Also, we compared the resulted output file obtained after applying each algorithm with the already existing model and its pretty good and the predicted values are close to actual values.

**REFERENCES**

1. Koehrsen, W. (2018, March 23). Histograms and Density Plots in Python. Retrieved November 01, 2020, from https://towardsdatascience.com/histograms-and-density-plots-in-python-f6bda88f5ac0
2. Zaric, D. (2020, April 12). Better Heatmaps and Correlation Matrix Plots in Python. Retrieved November 01, 2020, from <https://towardsdatascience.com/better-heatmaps-and-correlation-matrix-plots-in-python-41445d0f2bec>
3. BioSymetrics, (n.d), Mechanism of Action Prediction Platform, Retrieved from <https://www.biosymetrics.com/products/moa-prediction/>
4. Gaurav Singh, Joelostblom, Rafaelvalle, Apogentus, Phanindra Varma, SM Sivaprakaash (1964, June 01). Plot correlation matrix using pandas. Retrieved November 01, 2020, from https://stackoverflow.com/questions/29432629/plot-correlation-matrix-using-pandas
5. DataVizGuys, Sergey Bushmanov, Niraj, Shayan Shafiq, Predmod, Connor Wilmers (1965, January 01). How to plot a histogram using Matplotlib in Python with a list of data? Retrieved November 01, 2020, from https://stackoverflow.com/questions/33203645/how-to-plot-a-histogram-using-matplotlib-in-python-with-a-list-of-data
6. Packt, (n.d), Pros and cons of neural networks, Retrieved from <https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788397872/1/ch01lvl1sec27/pros-and-cons-of-neural-networks>
7. Real Python. “Linear Regression in Python.” Real Python, Real Python, 7 Nov. 2020, realpython.com/linear-regression-in-python/.
8. Jaadi, Z. (n.d.). A Step by Step Explanation of Principal Component Analysis. Retrieved November 09, 2020, from <https://builtin.com/data-science/step-step-explanation-principal-component-analysis>
9. Varshney, P. (2020, October 17). Q-Q Plots Explained. Retrieved November 09, 2020, from <https://towardsdatascience.com/q-q-plots-explained-5aa8495426c0>
10. Model-Building Strategies and Methods for Logistic Regression. (2005). Applied Logistic Regression, 91-142. doi:10.1002/0471722146.ch4
11. Dawson, C. (2019, September 26). Gradient Boosting in Python from Scratch. Retrieved October 06, 2020, from https://towardsdatascience.com/gradient-boosting-in-python-from-scratch-4a3d9077367
12. Megha Mishra (May 26, 2018), REGULARIZATION: An important concept in Machine Learning, https://towardsdatascience.com/regularization-an-important-concept-in-machine-learning-5891628907ea