Introduction

Background :-

At Santander, mission is to help people and businesses prosper. We are always looking for ways to help our customers understand their financial health and identify which products and services might help them achieve their monetary goals.

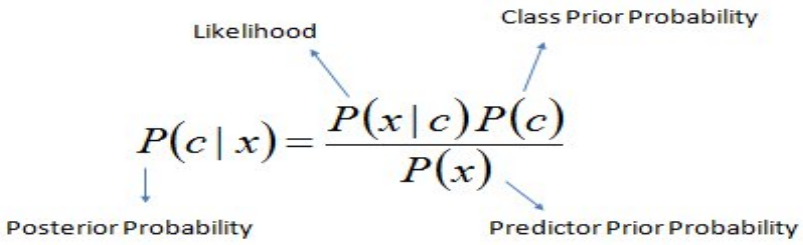
Problem Statement :-

in this challenge, we need to identify which customers will make a specific transaction in the future, irrespective of the amount of money transacted.

Number of attributes:

We are provided with an anonymized dataset containing numeric feature variables, the binary target column, and a string ID_code column. The task is to predict the value of target column in the test set. For Classification the Accuracy metrics need to be considered are AUC, Precision & Recall.

In this kernel, we will apply Bayesian inference on Santander Customer Transaction data, which has a binary target and 200 continuous features.



$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- $P(c|x)$ is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

We have been provided with two datasets, one of train and another is of test. Both the dataset have 200000 rows and 202 columns and 201 columns in train and test repectively.

Exploring Dataset

We have been provided with two datasets. One is of trainning dataset which includes target variable and another one is test dataset which dosent include any target dataset.

Four head rows of training dataset is mentioned below:

	ID_code	target	var_0	var_1	var_2	var_3	var_4	var_5	var_6	var_7	var_8	var_9	var_10	var_11
0	train_0	0	8.9255	-6.7863	11.9081	5.0930	11.4607	-9.2834	5.1187	18.6266	-4.9200	5.7470	2.9252	3.1821
1	train_1	0	11.5006	-4.1473	13.8588	5.3890	12.3622	7.0433	5.6208	16.5338	3.1468	8.0851	-0.4032	8.0585
2	train_2	0	8.6093	-2.7457	12.0805	7.8928	10.5825	-9.0837	6.9427	14.6155	-4.9193	5.9525	-0.3249	-11.2648
3	train_3	0	11.0604	-2.1518	8.9522	7.1957	12.5846	-1.8361	5.8428	14.9250	-5.8609	8.2450	2.3061	2.8102
4	train_4	0	9.8369	-1.4834	12.8746	6.6375	12.2772	2.4486	5.9405	19.2514	6.2654	7.6784	-9.4458	-12.1419

Both the training and test data have ‘ID_code’ column. Excluding ‘target’ and ‘ID_code’ column, both the dataset has same number and name of the variables ranging from ‘var_0’ to ‘var_199’, which denotes different variable set.

Both the datasets have no NA value. Min, Max description of few variables from train dataset is as follows:

Index	target	var_0	var_1	var_2	var_3	var_4	var_5	var_6	var_7
count	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000
mean	0.100490	10.679914	-1.627622	10.715192	6.796529	11.078333	-5.065317	5.408949	16.545850
std	0.300653	3.040051	4.050044	2.640894	2.043319	1.623150	7.863267	0.866607	3.418076
min	0.000000	0.408400	-15.043400	2.117100	-0.040200	5.074800	-32.562600	2.347300	5.349700
25%	0.000000	8.453850	-4.740025	8.722475	5.254075	9.883175	-11.200350	4.767700	13.943800
50%	0.000000	10.524750	-1.608050	10.580000	6.825000	11.108250	-4.833150	5.385100	16.456800
75%	0.000000	12.758200	1.358625	12.516700	8.324100	12.261125	0.924800	6.003000	19.102900
max	1.000000	20.315000	10.376800	19.353000	13.188300	16.671400	17.251600	8.447700	27.691800

The above image is itself descriptive.

1. The target variable is a factor of 0,1 only.
2. The standard deviation of all the variables is approx same. Though the min and max of all the variables is quite different.
3. Mean of few of the variables is approximately same for example ‘var_0’ and ‘var_2’. Also the min and max of these variables is approximately same.
4. More than 75% of the target variable is consist of 0 values only. That means our dataset is biased towards target value 0.

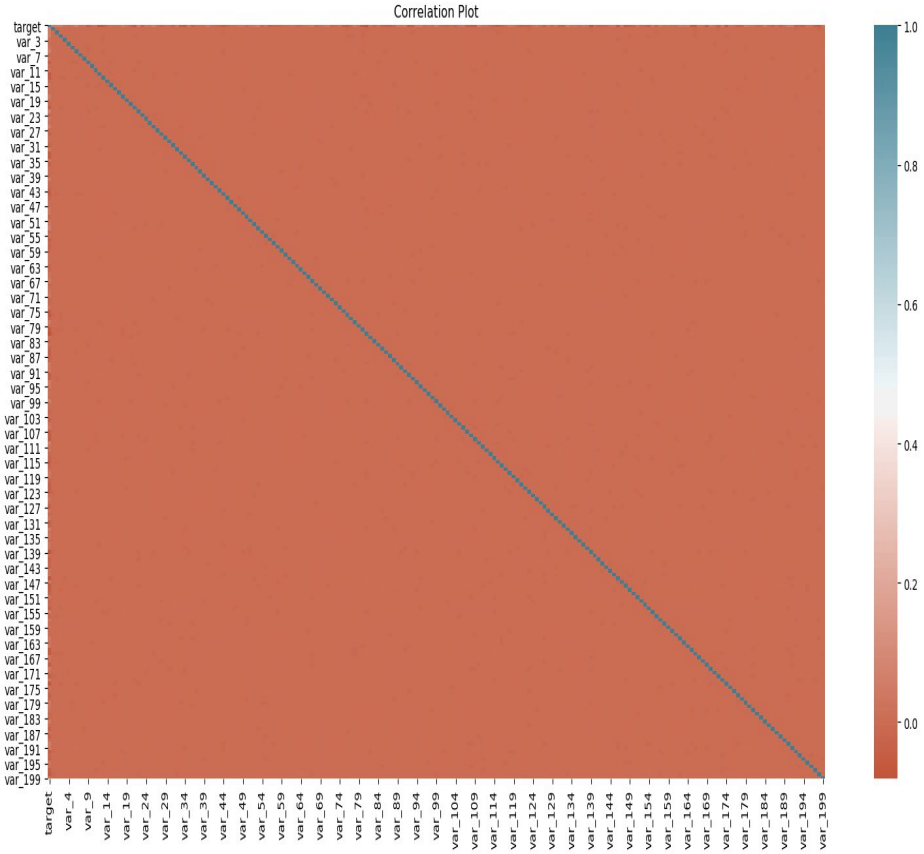
The below image is of test dataset:

Index	var_0	var_1	var_2	var_3	var_4	var_5	var_6	var_7
count	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000
mean	10.658737	-1.624244	10.707452	6.788214	11.076399	-5.050558	5.415164	16.529143
std	3.036716	4.040509	2.633888	2.052724	1.616456	7.869293	0.864686	3.424482
min	0.188700	-15.043400	2.355200	-0.022400	5.484400	-27.767000	2.216400	5.713700
25%	8.442975	-4.700125	8.735600	5.230500	9.891075	-11.201400	4.772600	13.933900
50%	10.513800	-1.590500	10.560700	6.822350	11.099750	-4.834100	5.391600	16.422700
75%	12.739600	1.343400	12.495025	8.327600	12.253400	0.942575	6.005800	19.094550
max	22.323400	9.385100	18.714100	13.142000	16.037100	17.253700	8.302500	28.292800

Description of the test dataset:

1. As it is a test dataset, so it dosent include any target variable.
2. This dataset also doesnot include any NA value.
3. The standard deviation of all the variables is quite different. But most of the variables have approx same standard deviation.
4. Mean of few of the variables is approximately same for example ‘var_0’ and ‘var_2’. Also the min and max of these variables is approximately same.
5. There is not much differenc ein the mean and standard deviation of the variables of test and train dataset.
6. Both test and train dataset contains same number of rows.

Now, in oder to use Naive Bayes algorithm , we need to the likelihood distributions are normal and independent. To check that variables are mutually independent or not, we will now look at the correlation between different variables of the train dataset and will draw a heat map of the correlation in order to visualize the correlation more precisely. We will use Pearson correlation.



From the above correlation heatmap plot we can see that all the variables are weakly correlated with each other. So till now we dont need to remove any variable.

Shapiro-Wilk test for the normal distribution of the data can be performed as follow:

Null hypothesis: the data are normally distributed

Alternative hypothesis: the data are not normally distributed

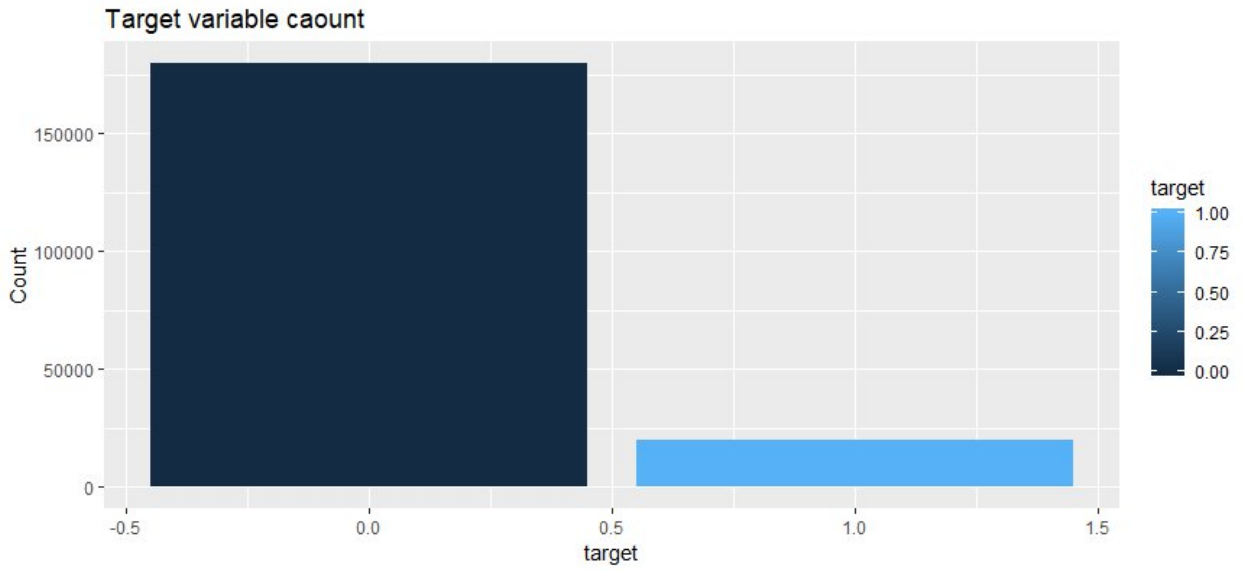
In the test the p-value comes to be less than 0.05 which mean null hypothesis is statistically significant. This means we fail to reject the null hypothesis and cannot accept the alternative hypothesis.

Data Visualizations

Count of Target variable

We will now visualise the data on the basis of few things. We hope that we can get best of the knowledge and secrets from both the dataset.

First is grouping by target variable to get the exact count of 0 & 1.



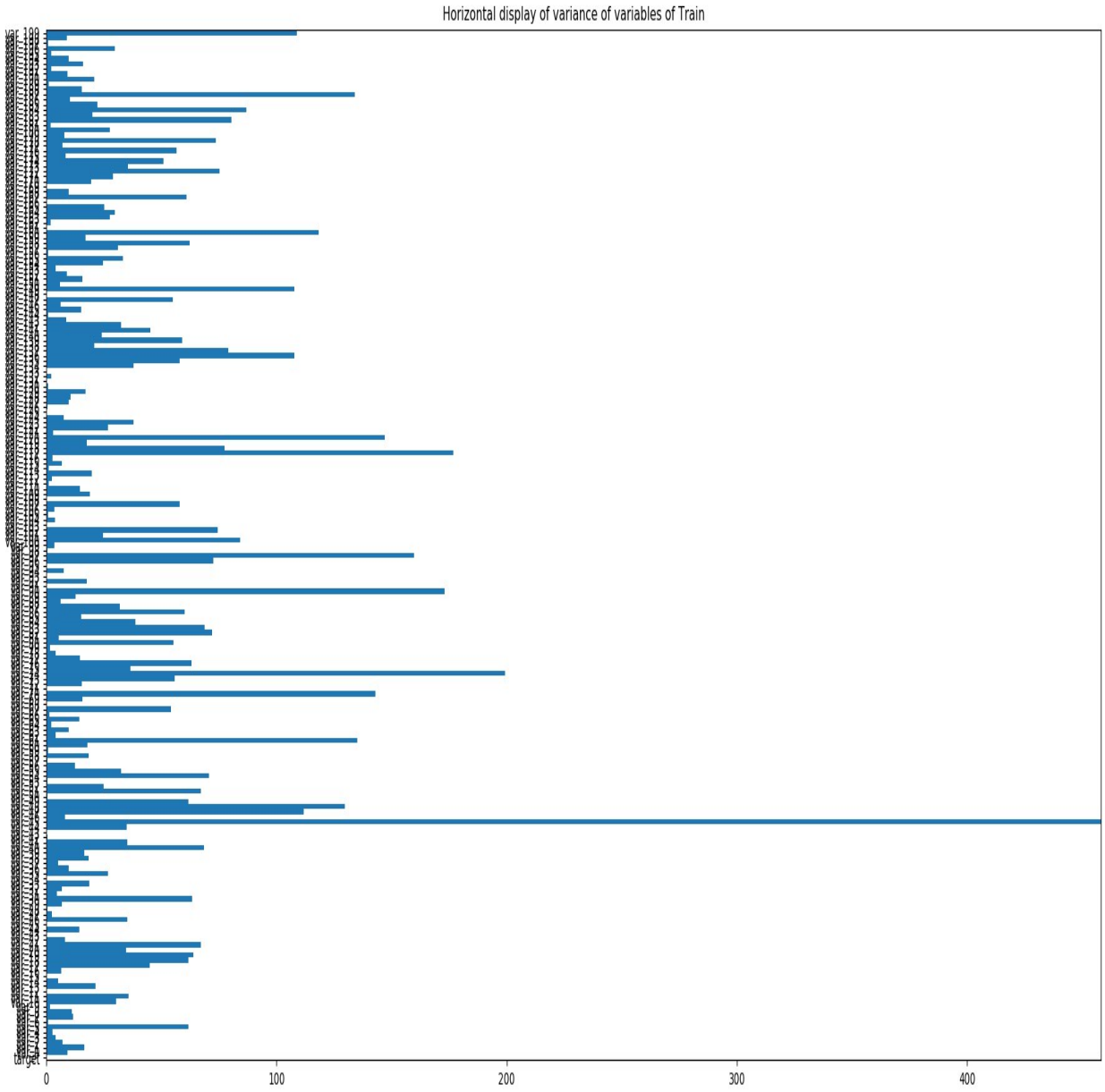
From this bar graph we can clearly see that our data contains more of 0 as target than 1. Hence our data is biased towards 0 target variable. The exact count of both the target values is, 179902 of 0 and 20098 of 1. That means approximately only 10% of the data contains 1 as target variable.We need to keep this in mind and need to use any sampling technique if required.

Visualization of Variance and Standard Deviation

Now, we will visualize the Variance and Standard deviation of all columns of both the datasets separetly and will see if get any pattern in that.

Train Dataset:-

Firstly will see the values of variance per variable of train dataset

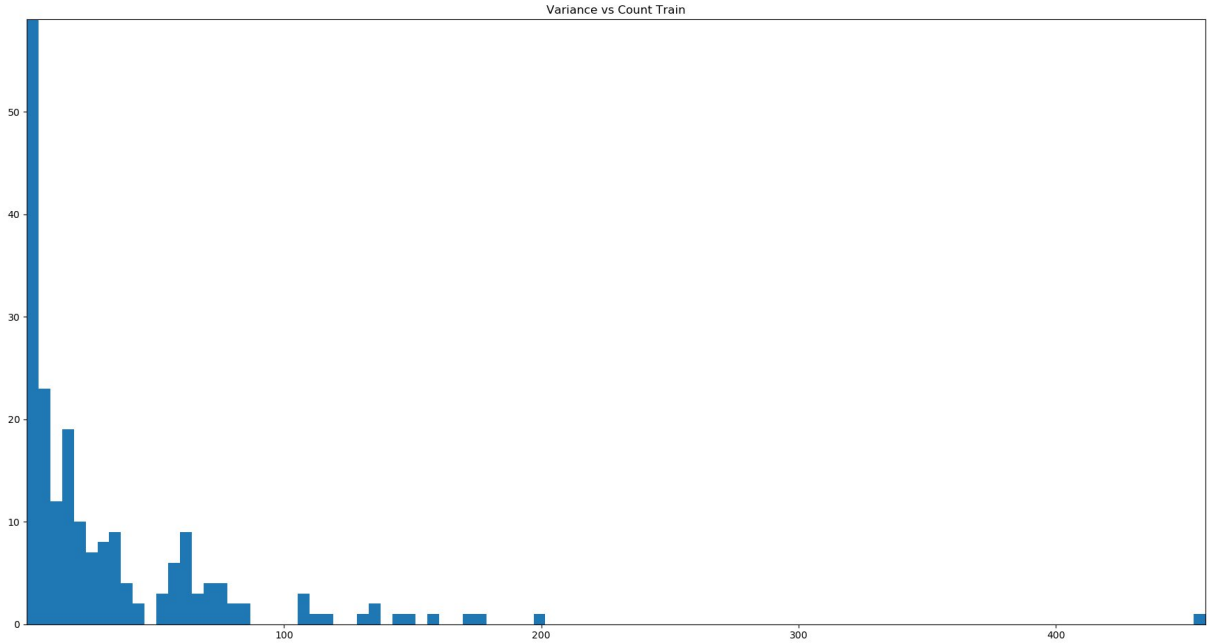


Formula for variance :-

$$\sigma^2 = \sum (X_i - \bar{X})^2 / N$$

σ^2 = *variance*
X_i = *the value of the ith element*
$\bar{X}$ = *the mean of X*
N = *the number of elements*

We can see the variance of a variable ‘var_45’ has a variance of 458. Some other also has high variance. So it would be better to plot Standard Deviation plot.

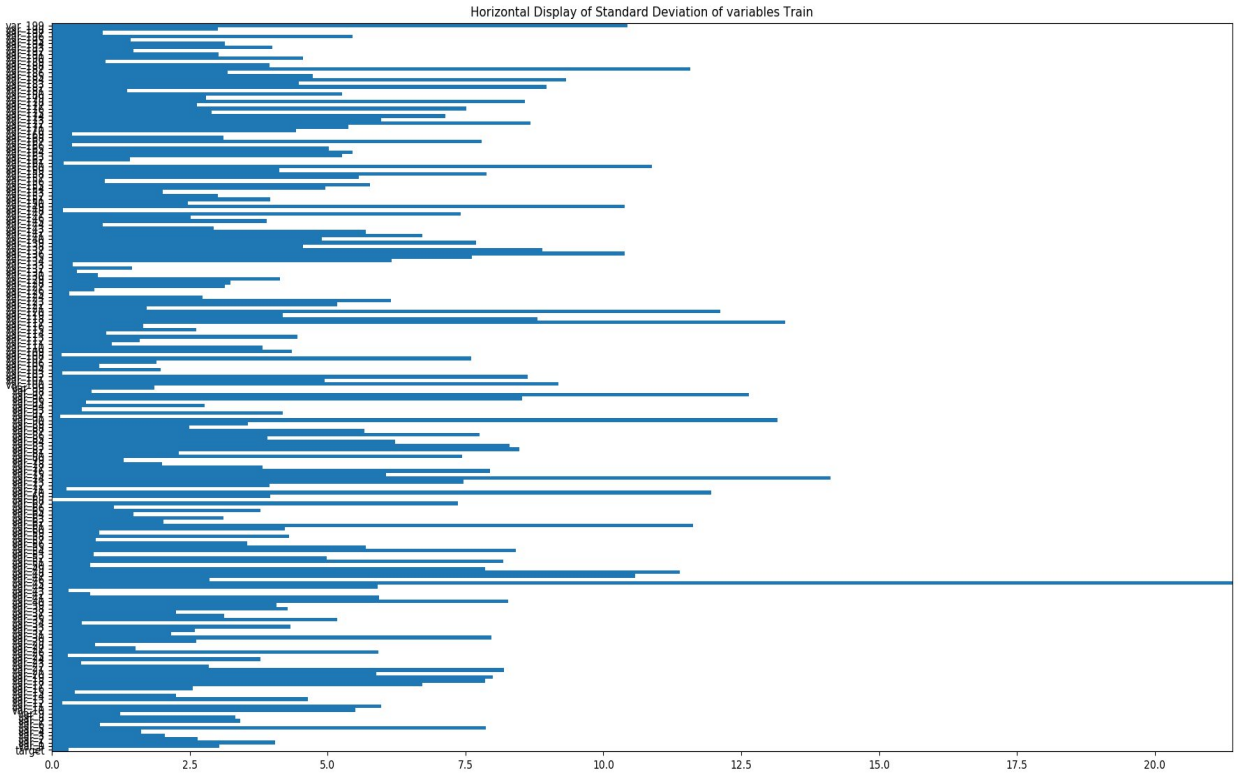


From this above graph we can see that most of the variables has variance less than 50.

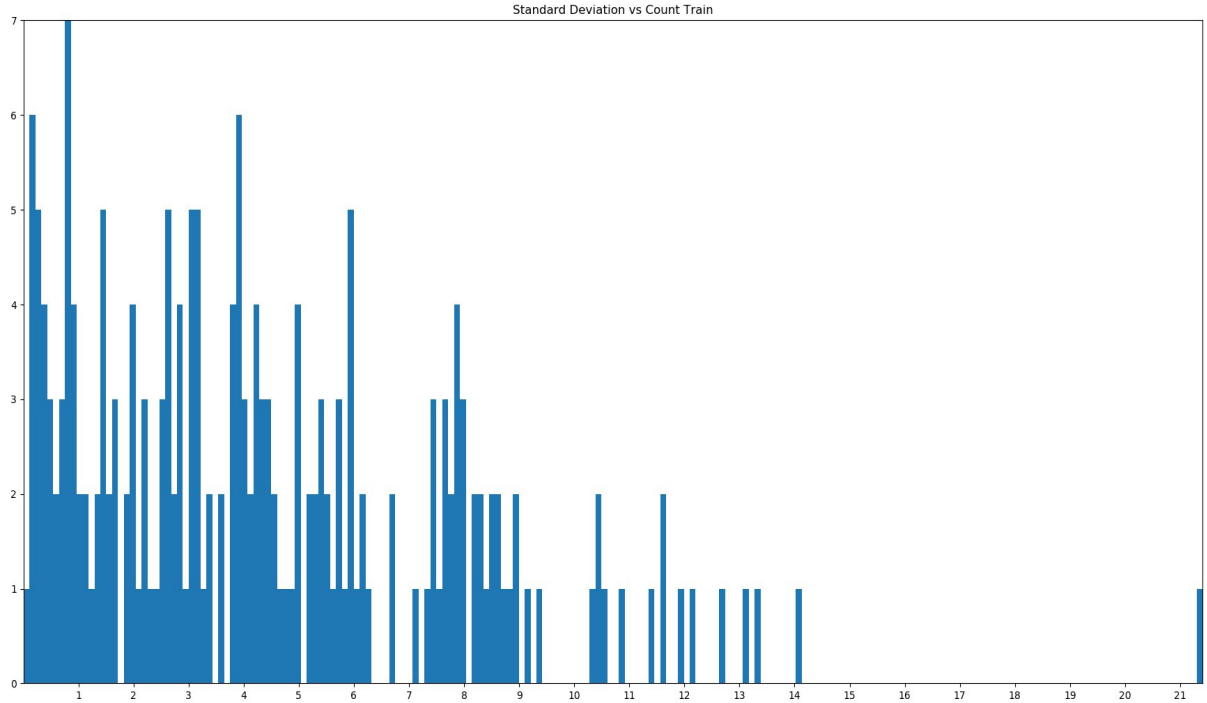
Now checking the Standard deviation of all the columns.

$$\sigma = \sqrt{\frac{\sum (x - \bar{x})^2}{n}}$$

σ = **lower case sigma**
$\sum$ = **capital sigma**
$\bar{x}$ = **x bar**



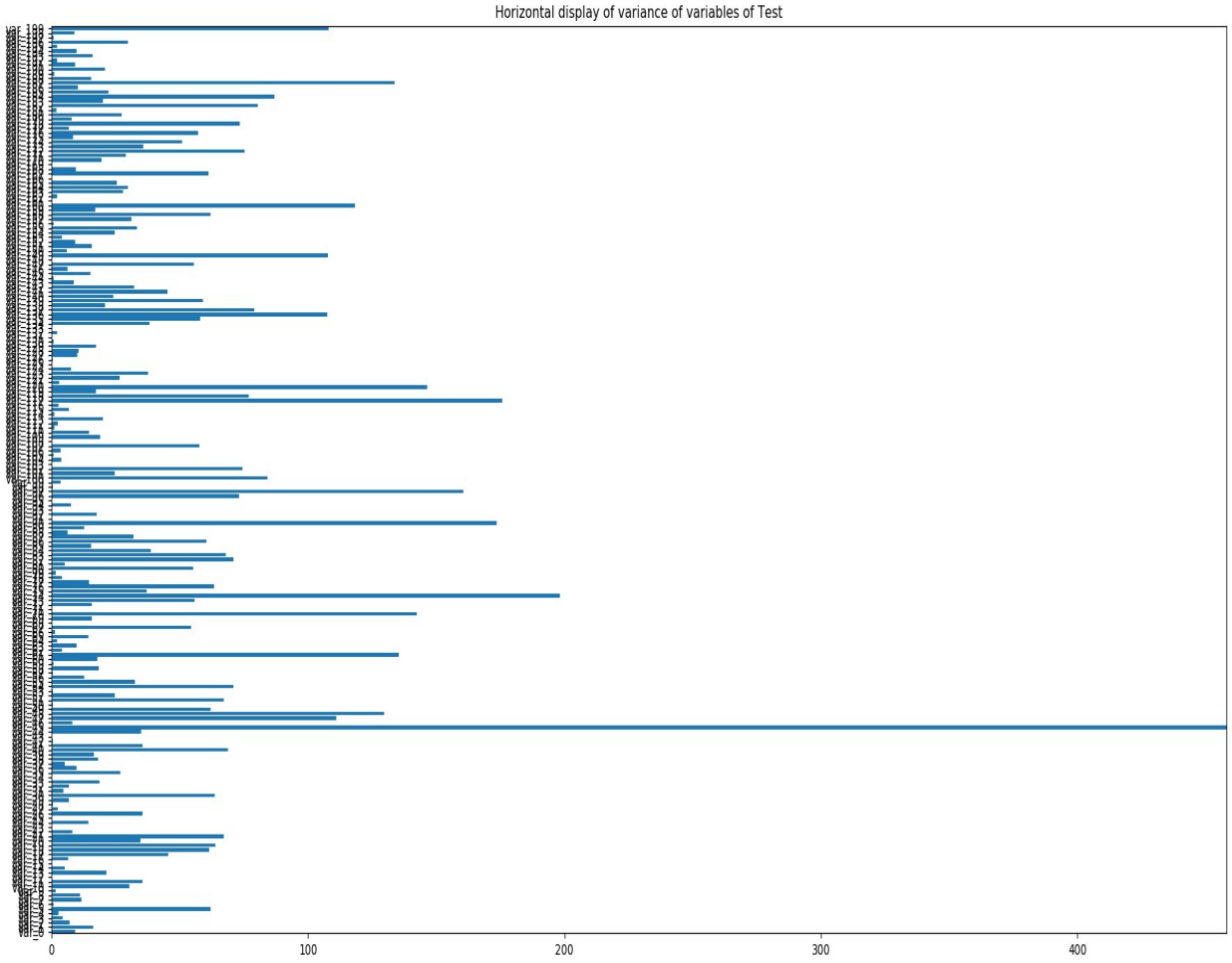
From this horizontal barplot of the standard deviation of all the columns of train we can see that the standard deviation of values of each column is very less. This can be more illustrative the count vs standard deviation plot



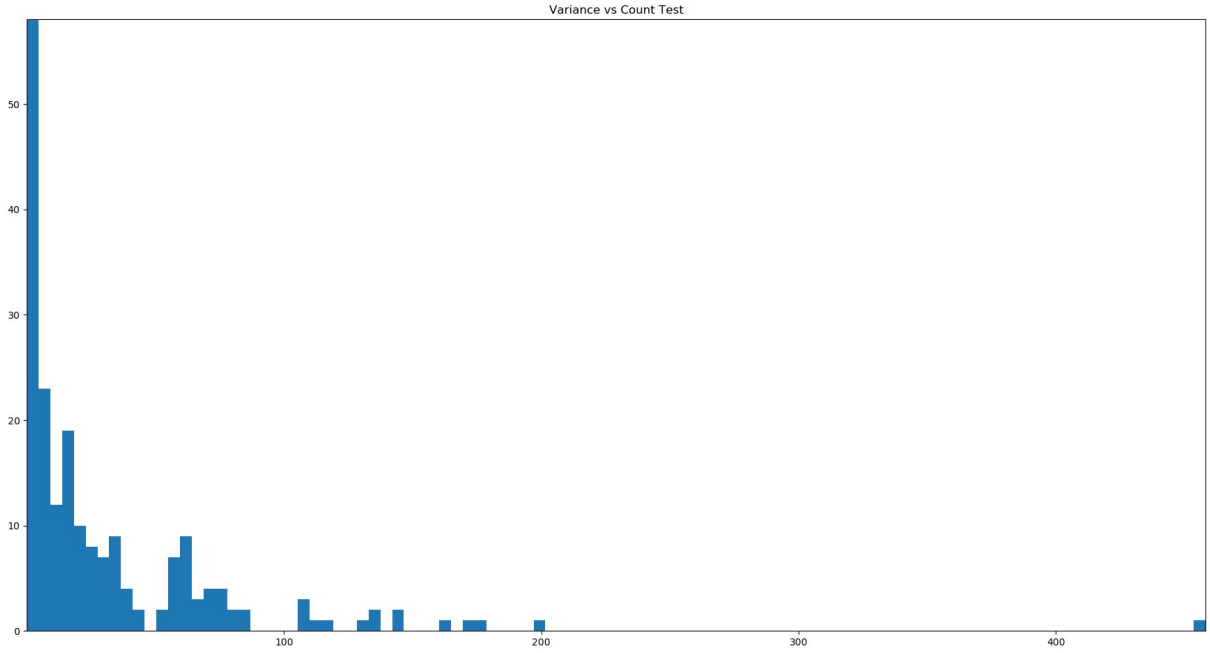
This barplot illustrates us better that most of variables has standard deviation less than 9. This is the deviation of the values from their mean. Hence we can say except from some of the variables in the train dataset, rest all have less standard deviation. Hence the data is not broadly devided.

Test Dataset:-

Firstly will see the values of variance per variable of test dataset

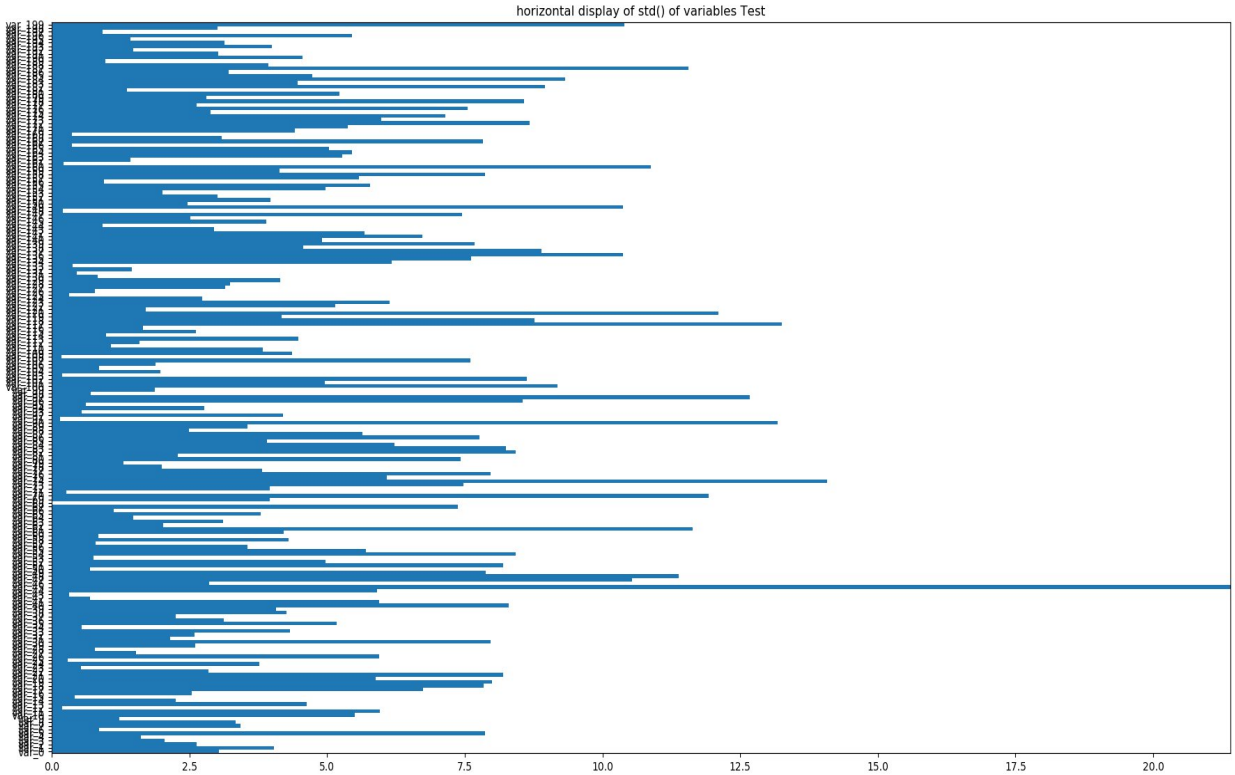


We can see the variance of a variable ‘var_45’ has a variance of 458 almost same as that of train dataset. Some other also has high variance. So it would be better to plot Standard Deviation plot.

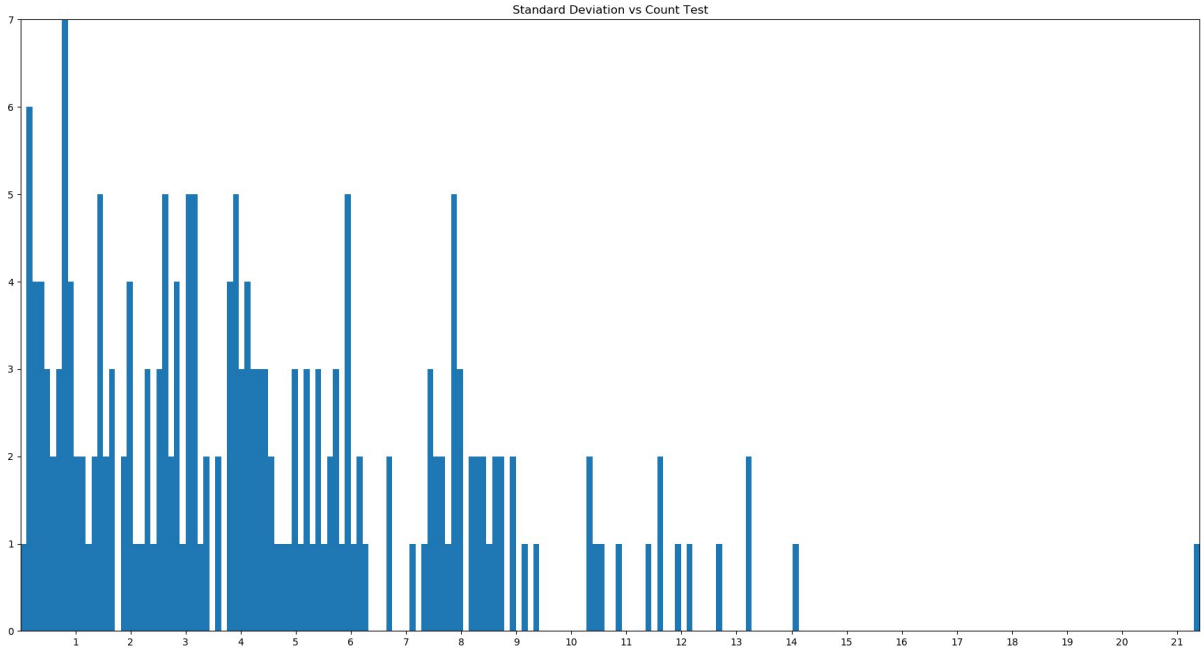


From this above graph we can see that most of the variables has variance less than 50.

Now checking the Standard deviation of all the columns.



From this horizontal barplot of the standard deviation of all the columns of test we can see that the standard deviation of values of each column is very less. This can be more illustrative the count vs standard deviation plot.

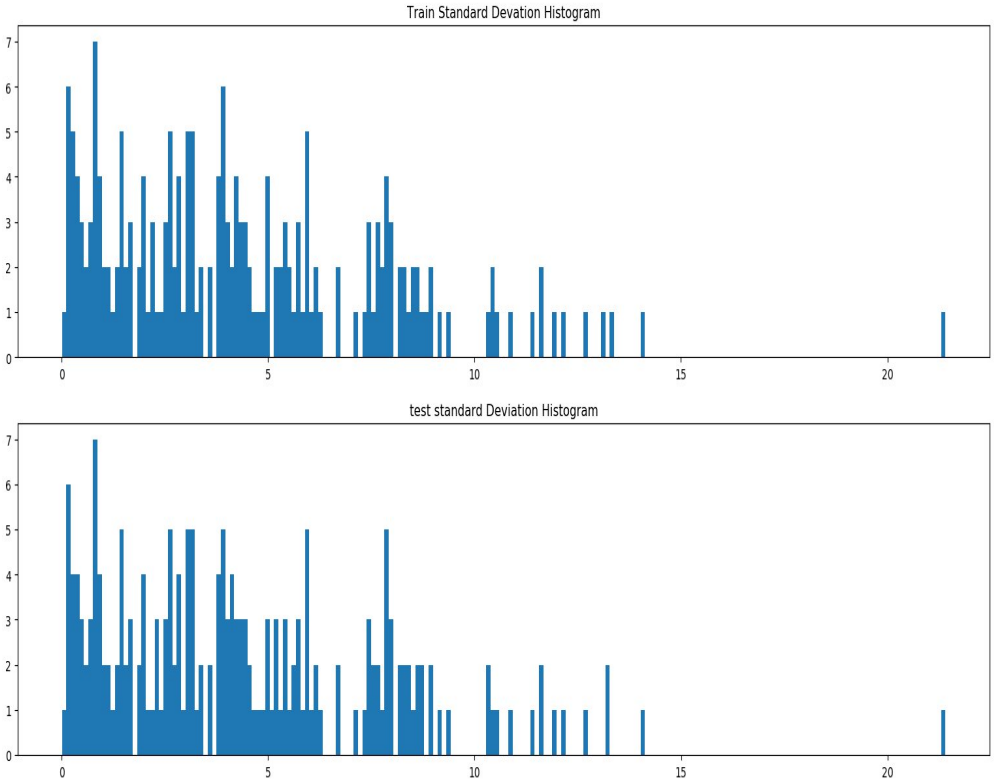


This barplot illustrates us better that most of variables has standard deviation less than 9. This is the deviation of the values from their mean. Hence we can say except from some of the variables in the test dataset, rest all have less standard deviation. Hence the data is not broadly devided.

From the above illustration we can see that there is not much difference in the std and var of both the datasets. Both of them has approximately same standard deviation and variance and which means that the data is not diversified much. This may include in difficulties to predict the target variable as values are very close to each other.

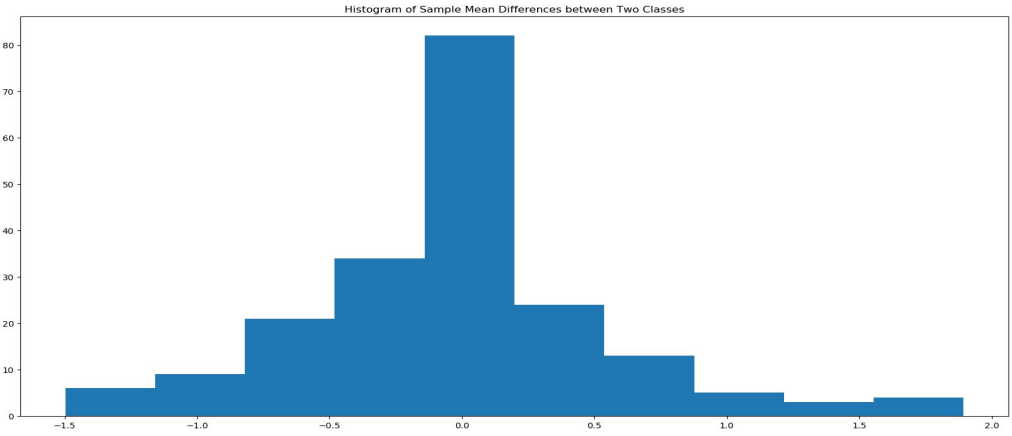
Comparing Standard Deviation of Both Test and Train Dataset

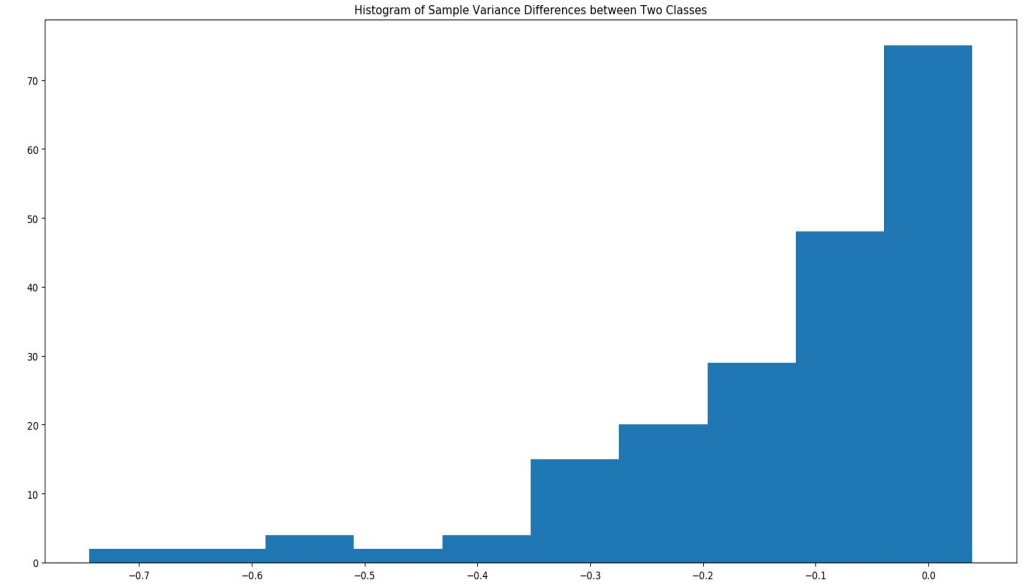
Now in order to get proper comaprison of standard deviation of both the datasets we will plot the barplot together for comparison.

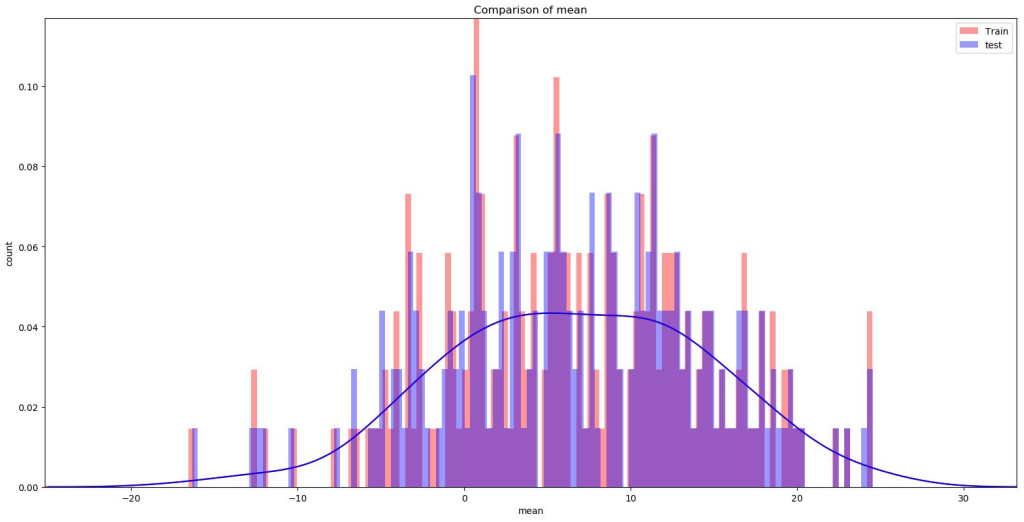


We can see that both the datasets have almost same pattern of bargraph. hence we can assume that both the datasets are in the same range.

Difference in Mean and Standard Deviation of two Target classes



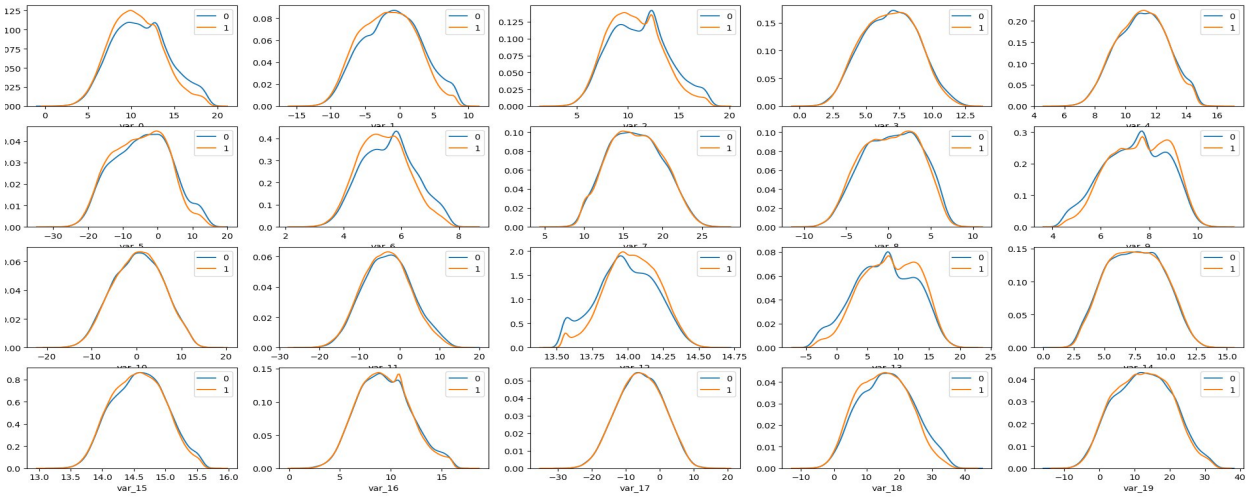


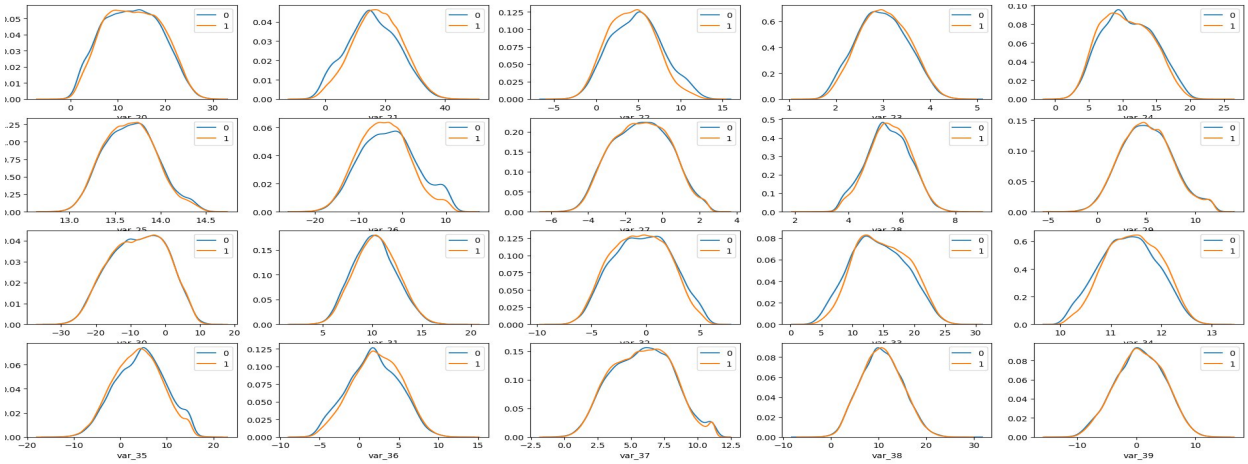


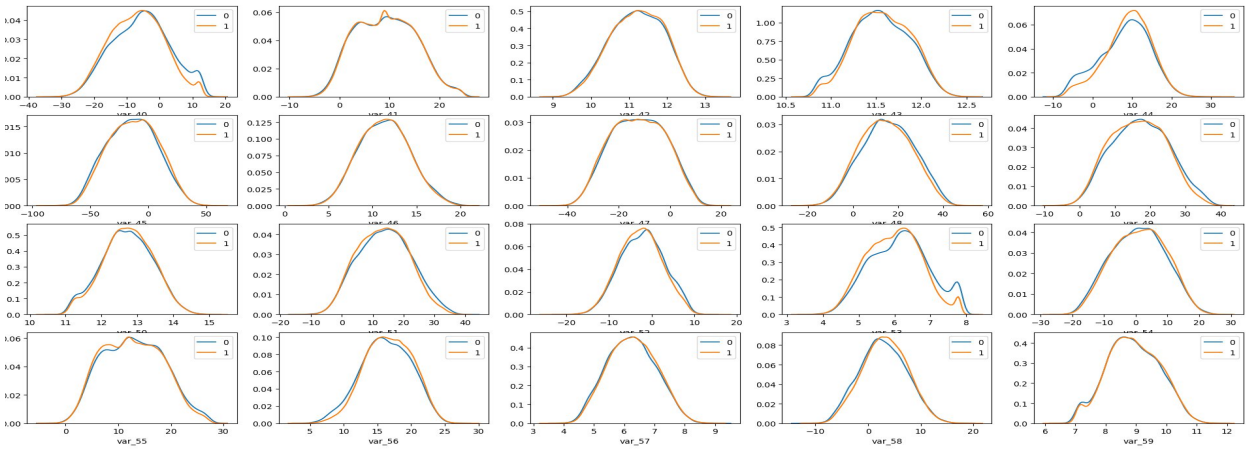
While the mean differences are more or less balanced around zero, the sample standard deviation differences are almost entirely on the negative side. This means that the negative likelihood distributions are more concentrated around their means than the positive ones. These differences add to the discriminative power of the model. The further away the centers of the distributions or the greater the difference in the spread of the distributions, the more it can tell about which class the point is coming from.

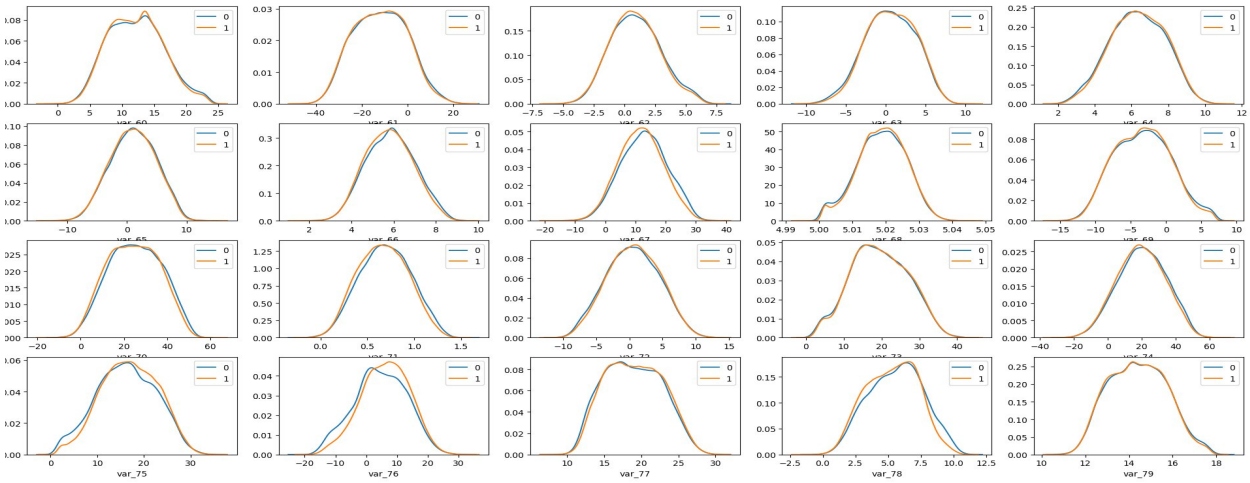
Plotting Graph of Variables with respect to Target for comparison

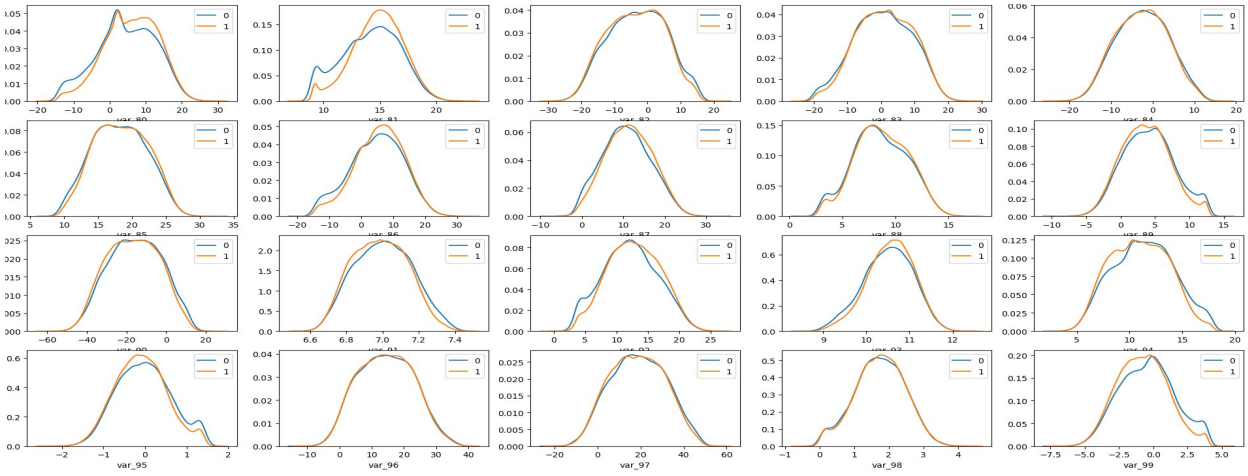
We will now plot the graph of each variable with respect to target variable in order to see if we can see any pattern or secret in the values that yield 0 and 1.

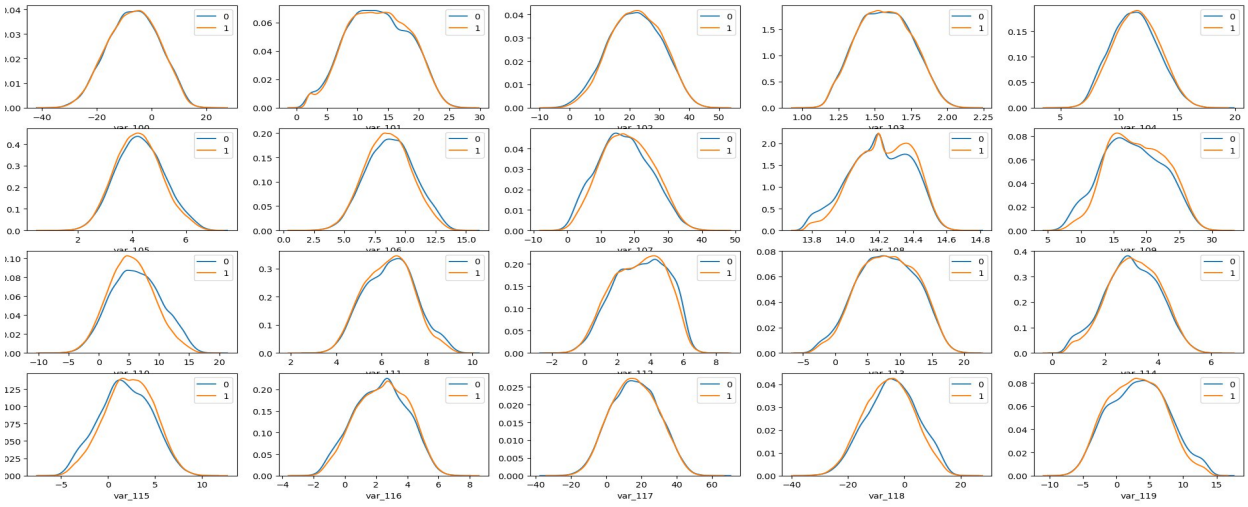


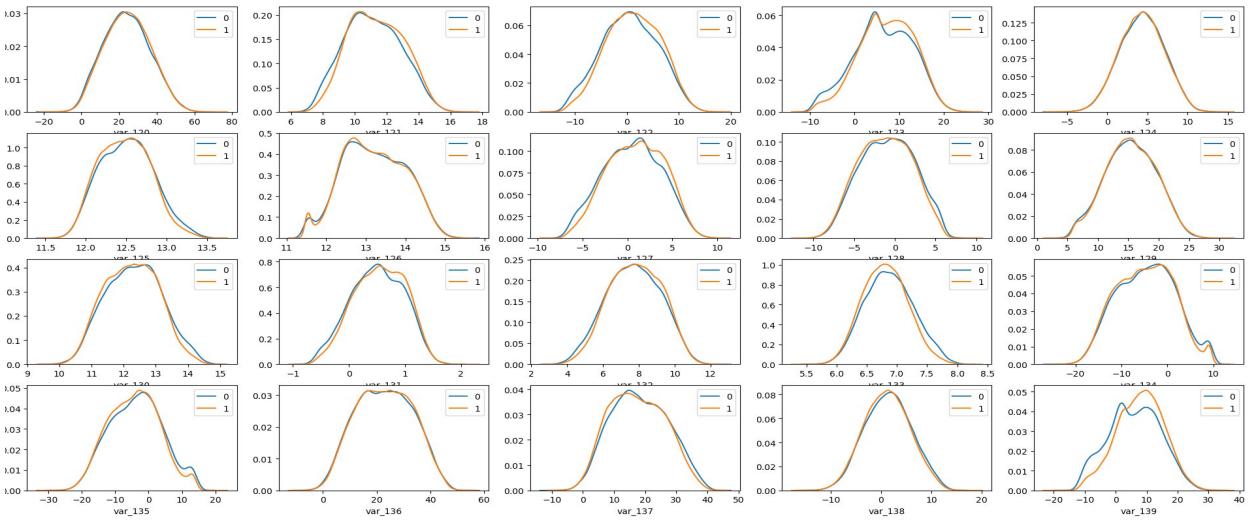


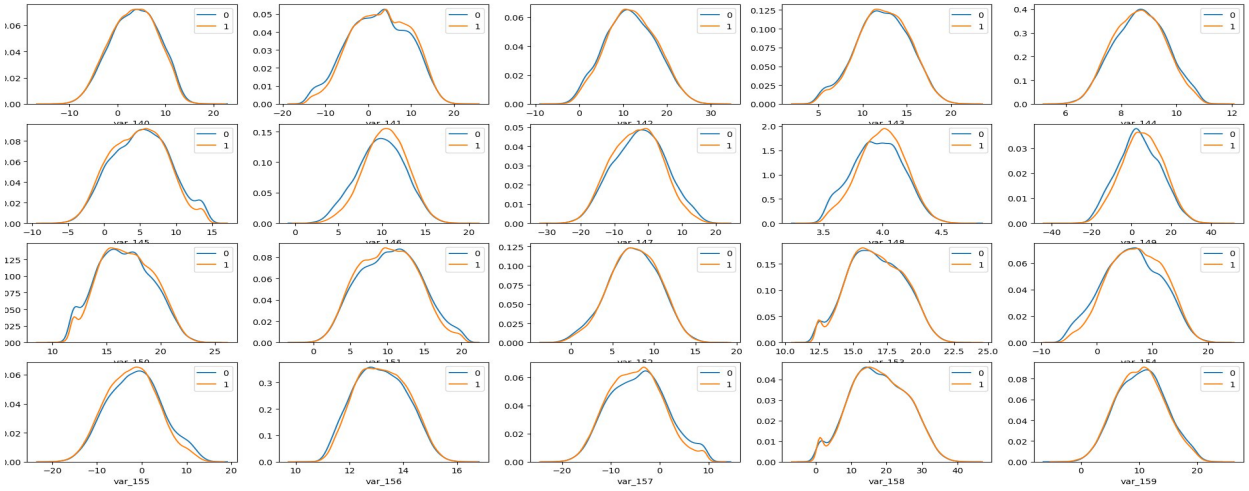


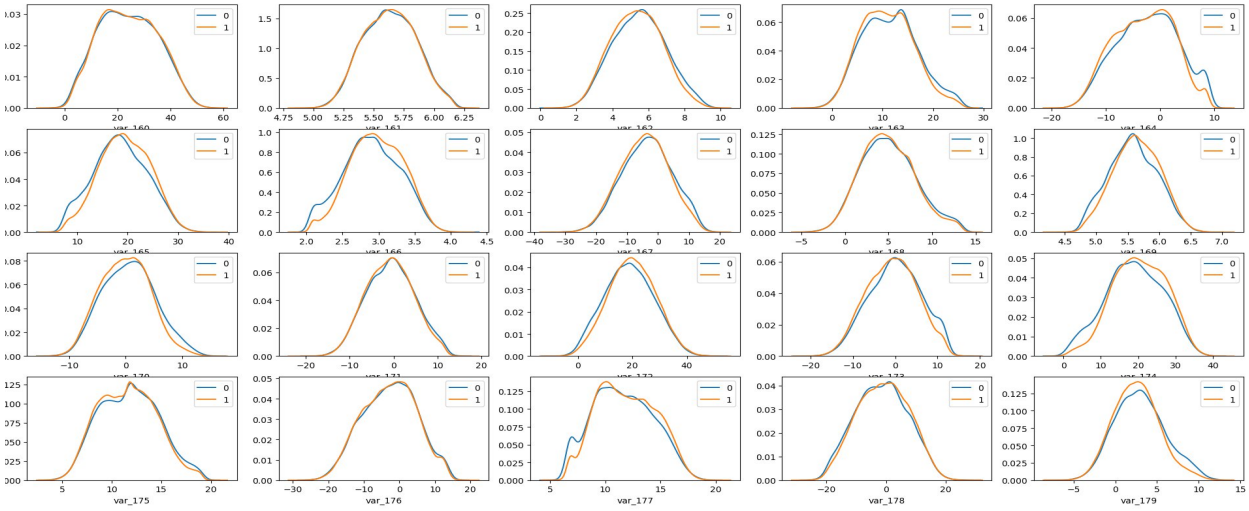

0
1
var_75
var_76
var_77
var_78
var_79

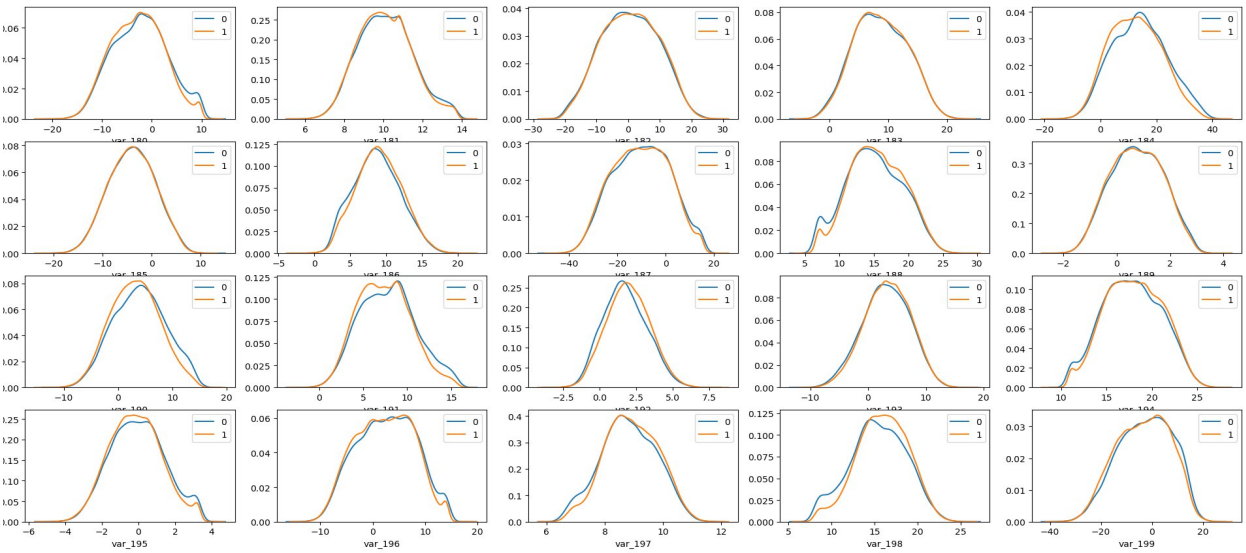

0
1
var_95
var_96
var_97
var_98
var_99


0
1
var_115
var_116
var_117
var_118
var_119


0
1
var_135
var_136
var_137
var_138
var_139


0
1
var_155
var_156
var_157
var_158
var_159


0
1
var_175
var_176
var_177
var_178
var_179

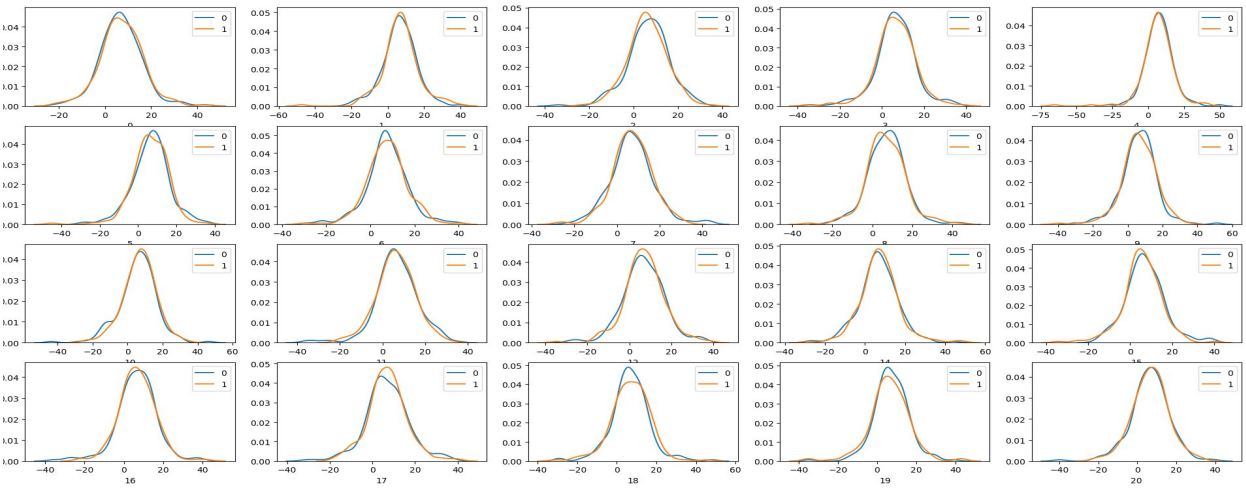


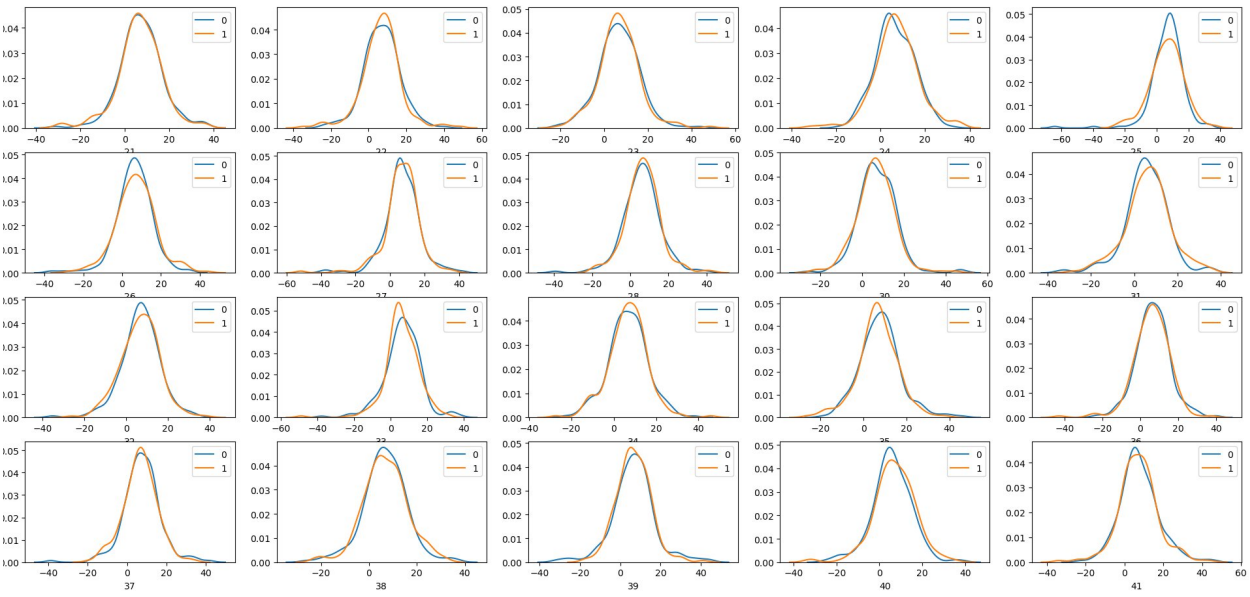
We can see that for few of the variables 'var_17','var_38','var_46','var_59' the values are almost same. That means their graph overlaps. That means they contain same information for target 0 s for target 1. Omitting them won't change anything.

Note:- I have tried to develop model by omitting the above mentioned variables, the model yield approximatey same results as when they were included in the model. Though I have included these variables in other models as these variables also countain some information regarding target 1 and might be descriptive while training a model.

Ploting Graph by Row, to Observe Any Pattern in Variables Combining to yield 1 and 0

I have plotted the graph for few rows so that we can observe any pattern from all the values of variables that combine to give 0 or 1. There mught be any difference in the values or any peak in variable that can predict that the trasaction.





Again, though there are certain peaks in the data but are very small to be evaluated. And those peaks are only observed in certain variables only. Hence the range of data is very small and would be difficult for the model to predict the values as the difference in the variables values that yeild that transaction is going to happen is very small.

We will try some feature engineering just to see that it might boost the performance of our model.

Feature Engineering

We have tried few of the new features that we can expect will contribute to the model and will help it to better understand the data and predict the target value.

The new features we tried are X^2, X^3, and ranking the data. Where X is column. SO this feature engennering was done on all the columns in a hope that this will further increase the difference in the features yielding to the target value as 0 and as 1.

After this feature emgeneering the shape of our dataset increased to 800 columns from 200 columns.

Thus though after the modeling we found that these features were not helping our model. Instead they were degrading the model accuracy.

Upsampling Dataset

As we have seen by the bargraph of the target variable that our datset is baised towards 0 i.e. negatively biased target variable. And it is important that we do some of the samoling techiniques in order to increase the count of positive target variable with dummy values.

We have used two upsampling techniques and have found that our model is responding to both of the very well.

The first teachnique is **SMOTE:**

> SMOTE stands for *Synthetic Minority Oversampling Technique* and is an oversampling technique used to increase the samples in a minority class.It generates new samples by looking at the *feature space* of the target and detecting nearest neighbors. Then, it simply selects similar samples and changes a column at a time randomly within the *feature space* of the neighboring samples.
> This technique has done exceptionally well in the Logistic Regression model but was performing poor in the Gaussian Naive Bayes algorithm. It also helped Logistic Regression to acheaive a good ROC score and AUC curve.

The second technique is **Oversampling Minority Class:**

> Oversampling can be defined as adding more copies of the minority class. Oversampling before splitting the data can allow the exact same observations to be present in both the test and train sets. This can allow our model to simply memorize specific data points and cause overfitting and poor generalization to the test data. Hence we first devide the data into X_train and X_test and then aply oversampling in X_train data.
>
> This has increased the quality of the models both Logistic Regression and Gaussian Naive Bayes and so we have selected this oversampling technique inorder to better train ou model.

Training and Modeling:

I have used many sets in order to train the model and get the best trained model.

I have ised 5 models.

> Logistic Regression

Gaussian Naive Bayes

Decession Tree

Random Forest

KNN imputation

Sets here means training all the above mentioned models in different ways inorder to get a better model.Total sets are 8. In all the 8 sets I have a mainted a different procedure., like some has only upsampling and some has only SMOTE

Below is the description of each set:

Model Set 1:- All models in this set are based on normal unteweked dataset

Model	Accuracy	Recall	Precision	AUC
Logistic Regression:	0.914	0.26	0.679	0.624
Gaussian Naive bayes	0.921	0.3661	0.7	0.6744
Decission Tree:	0.837	0.199	0.19	0.553
Random Forest:	0.829	0.175	0.164	0.538
KNN:	0.899	0.002	0.166	0.5

As all of the above models are trained in raw dataset, just to know how are they performing withcout addiiton of any of the tecnique. Here we can see that Logistic and Naive are performing much better than other models. Especially KNN is the worst. So we sill not consider thsi set and move further by appling different tecniques.

Model Set 2:- In this odel we are increasing the minority class by SMOTE

In this we have used SMOTE technique to upsample the dataset minority class and will se how the models wil perform in this upsampling technique.

Model	Accuracy	Recall	Precision	AUC
Logistic Regression:	0.787	0.771	0.288	0.78

Gaussian Naive bayes	0.869	0.083	0.173	0.519
Decission Tree:	0.705	0.31	0.12	0.532
Random Forest:	0.746	0.127	0.263	0.531
KNN:	0.108	0.99	0.099	0.501

We can clearly see that the SMOTE technique has done way better than the raw dataset. That mens upsampling does help in our case. But only with logistic regression and NAive Bayes. While the Tree based models and KNN doesnot show any improvment in the evaluation metrics, rather they have performed porrer than the pervious dataset.

But still after SMOTE we achived AUC score of only 78 and our Recall is also less. Lets try other sampling technique and see if they can achieve better.

Model Set 3:- Upsampling the dataset

Here istead of using SMOTE as an upsampling techniwue we will use simple oversampling method to oversample the minority class.

Model	**Accuracy**	**Recall**	**Precision**	**AUC**
Logistic Regression:	0.782	0.775	0.283	0.779
Gaussian Naive bayes	0.813	0.795	0.322	0.805
Decission Tree:	0.829	0.171	0.162	0.536
Random Forest:	0.865	0.135	0.218	0.54

By this upsampling technique,we can that though there is not much improvement in the other models except Naive Bayes. There is a high improvement in the Naive Bayes. The AUC score has increased to above 80 . That indicates that our model is performing well in upsamplingg technique.

KNN Rejection :-As there was not much improvement in the KNN an was performing very poor. So we are dropping this model from further considerations. We will only apply other techniques to only these above models.

Model Set 4: Dropping few variables from the dataset as those variables contains same value for 0 and 1. Data will be upsampled as the best result was seen in upsampled dataset

In the set we will drop few of the variables from the dataset that we have observed above in the visualization graphs.Those variables were giving almost the same graph for both 0 and 1 target variables. Lets see that droping those variables will make what effect to our model.
We have keept upsampling as that was giving the best results uptill now.

The variables that will be dropped off are 'var_17','var_38','var_46','var_59','var_61','var_65', 'var_73','var_79','var_96','var_100','var_124','var_185'.

Model	Accuracy	Recall	Precision	AUC
Logistic Regression:	0.781	0.775	0.282	0.778
Gaussian Naive bayes	0.812	0.794	0.321	0.804
Decission Tree:	0.831	0.171	0.165	0.537
Random Forest:	0.866	0.132	0.219	0.54

As we have expected, It didnt made any chnages in the models. That means those variables were not assisting the models as they were carring same weight for both of the target variables.

Model Set 5: Dropping few variables from the dataset as those variables conatins same value for 0 and 1. Data will be SMOTE as the best result was seen in upsampled dataset.

Lets try the same data set that we have use in the model set 04 and apply SMOTE.

As Logistics was perfroming well in SMOTE. It might be acase that it will perform much better now.

Model	Accuracy	Recall	Precision	AUC
Logistic Regression:	0.787	0.769	0.287	0.779
Gaussian Naive bayes	0.866	0.1	0.183	0.525
Decission Tree:	0.703	0.319	0.122	0.532

Random Forest:	0.746	0.272	0.138	0.535

As we have expected, It didnt made any chnages in the models. That means those variables were not assisting the models as they were carring same weight for both of the target variables.

But still in the next model set i have kept those just as though they are carring same weight for both of the target set but still there might be a chance that theri weight increses and will effect the model by using further techniquesand also for any magic to happen.

Model Set 6: Featuring Engineering

In this model set we will do some feature engineering in hope that new variables will give model a good understanding of the dataset. As there is very less variation inthe dataset variables for the two target variables so we will square each column to make new column and we will cube each column and make a new column. As the value increases, the difference will also increase and might our model will get enough differnce to get trained well. This might can help lets see.

We will upsample the data as we are getting a good reult with the upsampled data. As the data now becomes huge we will do standard scaling.

Dropped Tree based Algorithms: As tree bsed models were not perferming well continously so we are dropping them. Since the first model set there was no improvement in these models. Further as the number of estimators increses in the Random forest, worst the model performs. Maximum of 3 estimators are performing farely, which is a very less count for such a model.

Model	Accuracy	Recall	Precision	AUC
Logistic Regression:	0.871	0.349	0.696	0.66
Gaussian Naive bayes	0.611	0.904	0.192	0.741

Sadly our both top models performed very bad. So we will drop the idea of Feature engennering.

.

Model Set 7: Feature Engineering 02

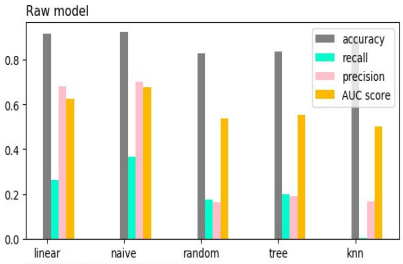
We will use the same dataset in Model set 06, but will increase one extra feature with respect to each variable. We will rank the values of each column inorder that it can support model and AUC Score.

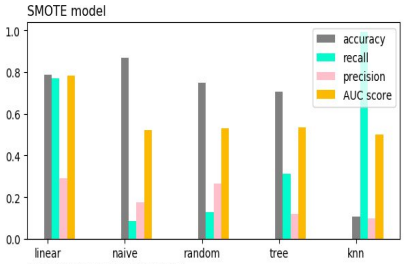
Model	Accuracy	Recall	Precision	AUC
Logistic Regression:	0.871	0.349	0.696	0.66
Gaussian Naive bayes	0.798	0.684	0.286	0.747

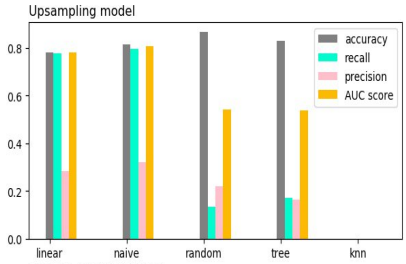
Nope. This model set also dosent performed very well.

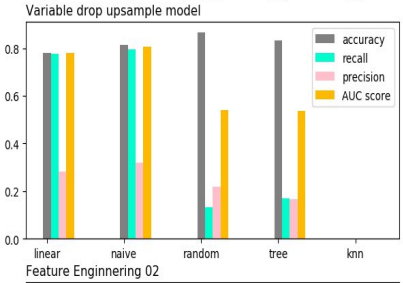
It seems that simple upsampling gives us best fit til now.

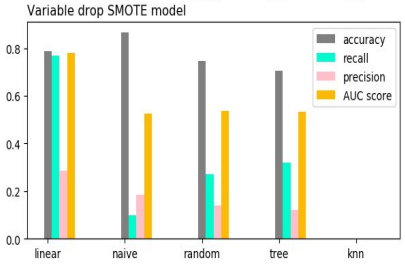
Conclusion and FInal Model

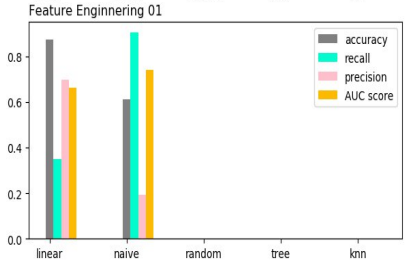


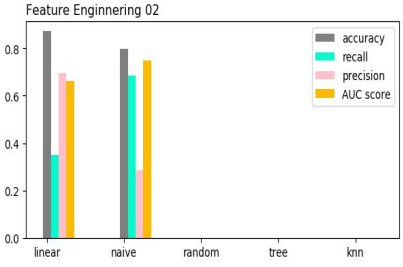












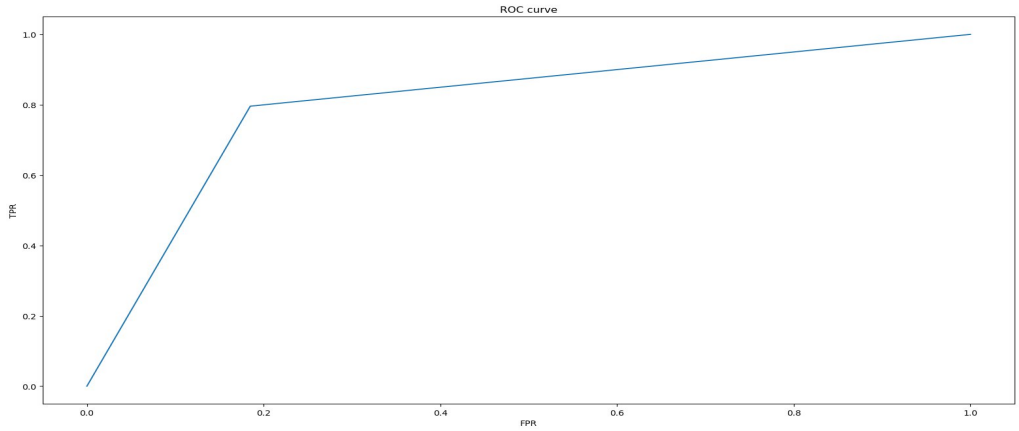
By looking at the various test cases that we have build inorder to get a good fit model, we come to conclusion that simple upsampling Gaussian Naive Bayes is performing well.

Final Model Selected:- Our final model that we have selected is from Model set 03 oversampled Gaussian Naive Bayes. In from all the model test cases that we have generated, it have performed very well.And also It is giving a good AUC score of 88 in probability prediction and a Recall score of 79.5 that is approx to 80 from our split test data. Also the AUC graph is good curve for this model.

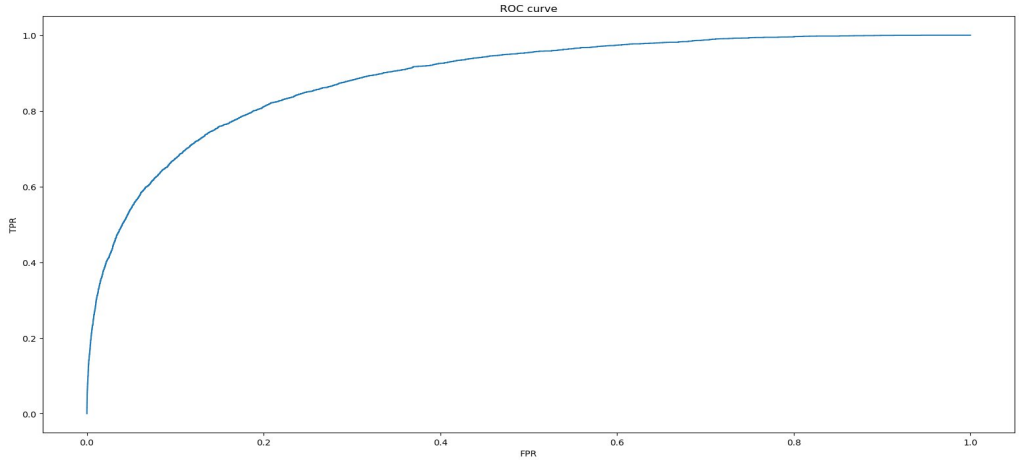
Here I have paid focus on getting good Recall score and AUC score. As in this type of data I think that Recall or sensitivity is must. We can ignore the Precision, as this data is of finding the customers that can buy a specific product. So in this case we can contact few customers who will not buy the product but we cannot loose customers that were actually ready to buy the product but our model predicted that they wount buy so we missed them.

So we cannot take that risk of missing those people that can buy a product. So in this type of data set Recall is much more important than Precision.

Below is the ROC curve for oversampled Gaussian Naive Bayes.



ROC graph of the predicted probabbilites is below.



Rejecting Other Models

1. **Logistic Regression:** Logistic Regression performed approximately same as that of Naive Bayes. But under the condition of oversampling, Naive Bayes gives a good score of AUC and Recall.
2. **Decission Tress:** In all the model test cases that I have taken into account, the tree based models were performing below average. Inspite of so much of diffreent techniques in which other model have shown a good performance, but this model got even worse. So rejecting this model was the only option that I was left with
3. **Random Forest:** The same goes with this tree based model also. This model was performing even worse than Decission Tree. The highest number of estimators under which this model performed averagely was 3-5. If we were to increase the estimators than this number, the model starts to perform even worse. And 3-5 estimators is a very less count for this type of powerfull model So rejected this model
4. **KNN:** From the starting, this alogorithm was having very difficulties in predicting the test class. This ight be because this model works on nearest neighbours and the range of variables by which it differ for the different target class is very less. So this model was having very high difficulties in predicting the test class. It was performing so poor that I have removed this model after second test case only.

Summary

After performing all these test cases we have found that Gaussian Naive Bayes is performing the best in this type of dataset.

Model	Accuracy	Recall	Precision	AUC
Gaussian Naive bayes	0.813	0.795	0.322	0.805

Further the prediction probabilities this model is giving a score of 0.88. All the score are out 1 so this is the best score. Here I have paid focus on getting good Recall score and AUC score. As in this type of data I think that Recall or sensitivity is must. We can ignore the Precision, as this data is of finding the customers that can buy a specific product. So in this case we can contact few customers who will not buy the product but we cannot loose customers that were actually ready to buy the product but our model predicted that they wount buy so we missed them.

So we cannot take that risk of missing those people that can buy a product. So in this type of data set Recall is much more important than Precision.

The I am providing the two final file for submission which contains the prediction of the test dataset. the one file has the prediction probabilities which should be used to calculate the AUC scroe and another submission file contains the binary target values which should be used to cross check the precision and recall.