311 Service Request

Group Members:-

1.Deepak Kadam (F16111035)

2.Arisha Chamadia (F16111049)

3.Swati Patra (F16111065)

Guided By:-

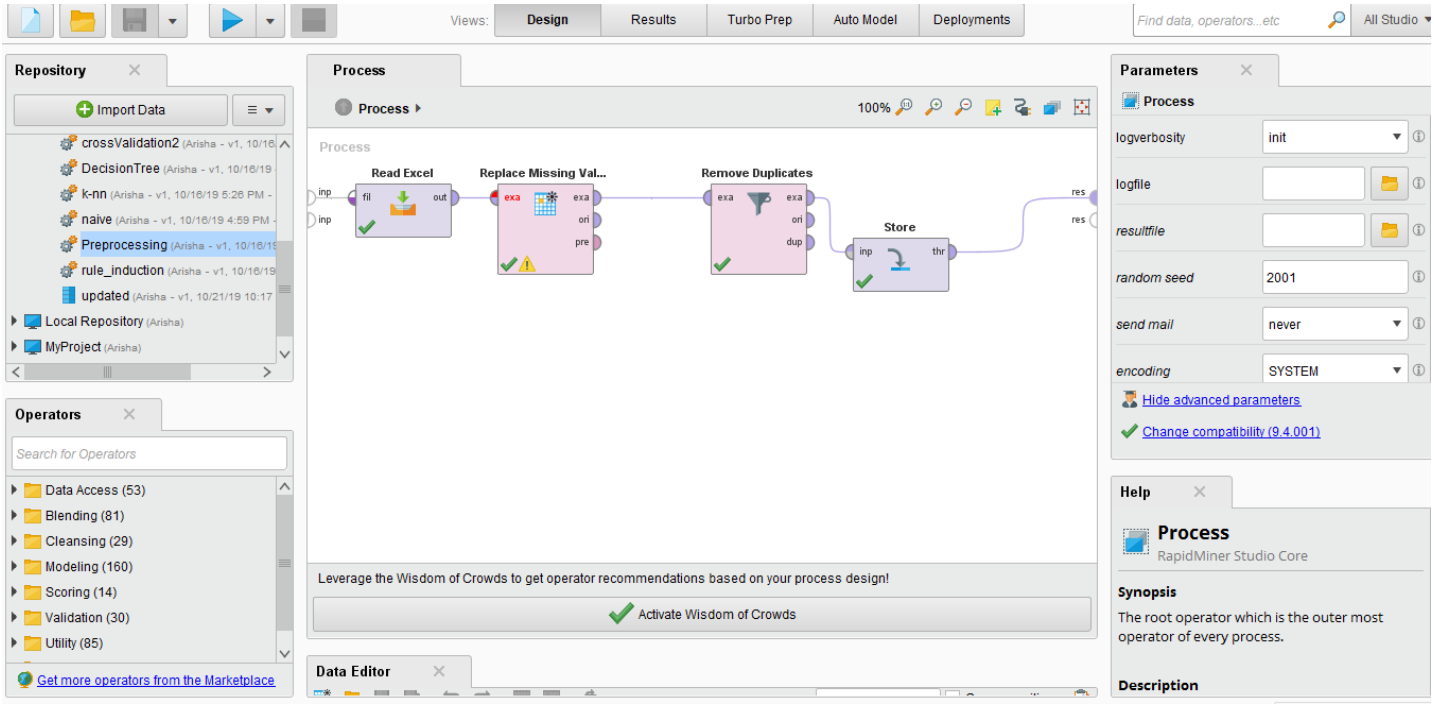
Prof R.A.Khan

Dataset Description: Data on 311 service requests in Pittsburgh. Data is related to every call made on 311. 311 is a non-emergency phone number that people can call in many cities to find information about services, make complaints, or report problems like graffiti or road damage. Even in cities where a different phone number is used, 311 is the generally recognized moniker for non-emergency phone systems.

Attributes	**Type**
REQUEST_ID	Numerical
CREATED_ON	Numerical
REQUEST_TYPE	Categorical
REQUEST_ORIGIN	Categorical
STATUS	Binary
DEPARTMENT	Categorical
NEIGHBORHOOD	Categorical
COUNCIL_DISTRICT	Numerical
WARD	Numerical
PUBLIC_WORKS_DIVISION	Numerical
PLI_DIVISION	Numerical
POLICE_ZONE	Numerical
FIRE_ZONE	Numerical

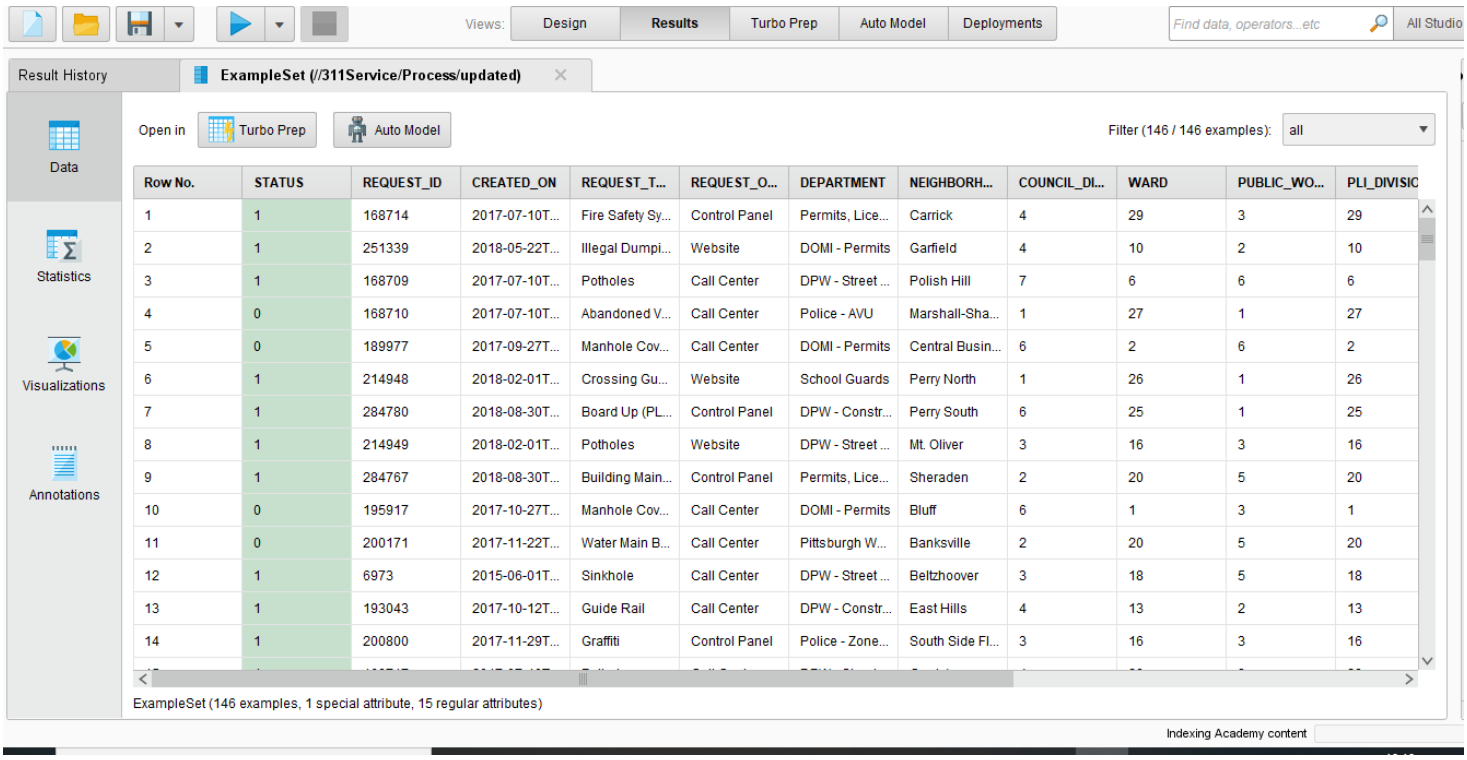
X	Numerical
Y	Numerical
GEO_ACCURACY	Categorical

Import Dataset: Getting the dataset into Rapid Miner for preprocessing.

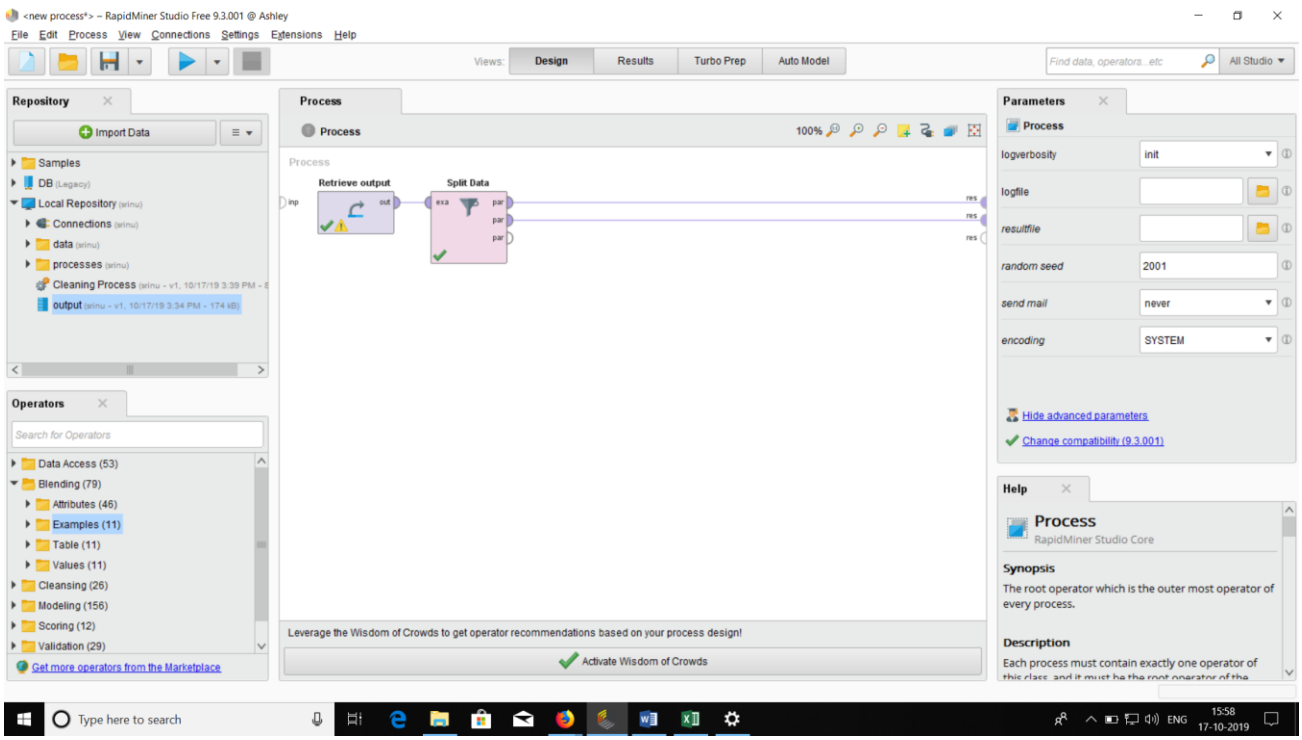


Cleaning Process: The dataset is not perfect i.e it is not is the form we need. The classification algorithms works on the basis on Arithmetic and Statistical rules. Therefore numeric entries are to be fed into the dataset and not other types like binary and categorical. For this we have to convert them into numerical values.

Furthermore, some attributes have missing values so we need to fill these missing values,we need to remove any duplicate entries if present using statistical methodologies which ever promises to yield better results in the future.



Splitting the Result: Now after data pre-processing we need to split the data into train and test set. This data splitting is necessary for training and testing of the model. The train set will be used to train the model and the test set to evaluate the model. the training data set should be relatively larger than the test set as it will define the model performance.



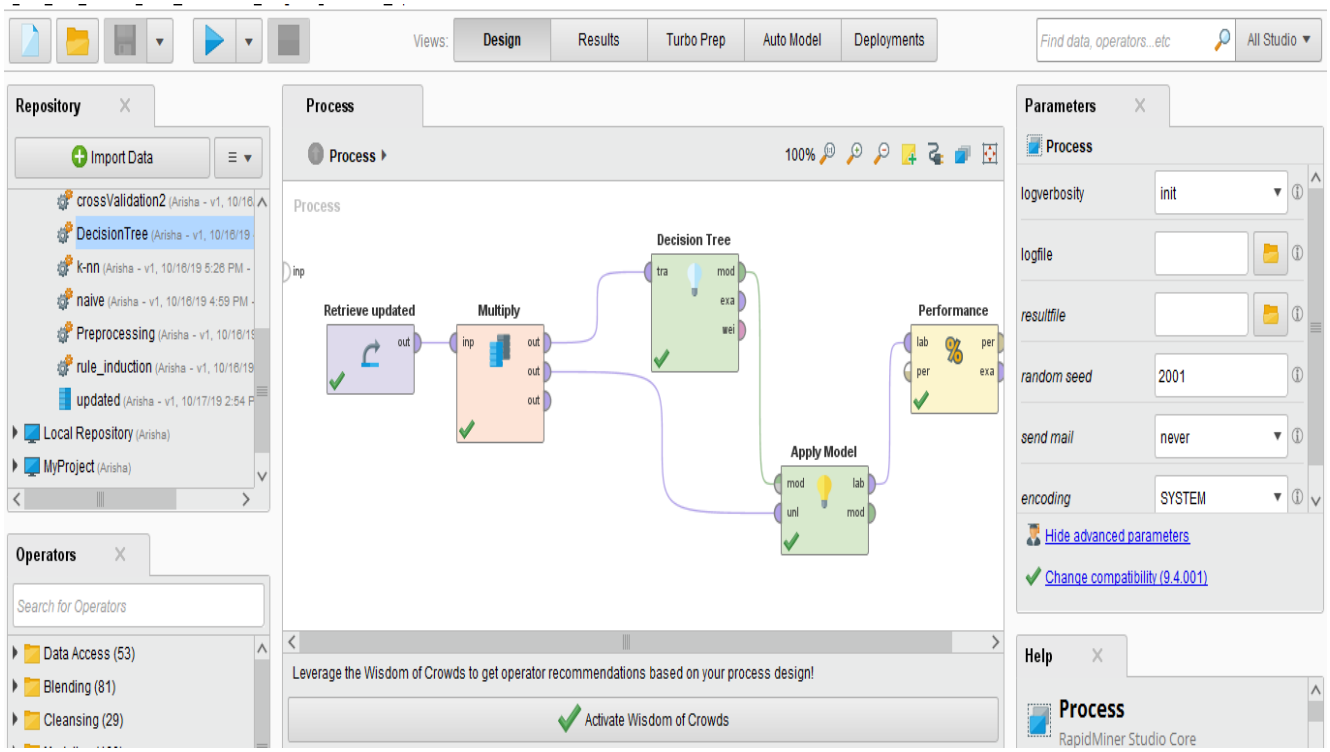
Training the Model: Training of the model here is done by using 3 classification algorithms :

1. Decision Tree
2. K-NN
3. Naïve Bayes

1. Decision Tree:

Decision tree is the most powerful and popular tool for classification and prediction. A Decision Tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

A tree can be learned by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called *recursive partitioning*. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification.



Decision Tree Accuracy/ Confusion Matrix:

eVector (Performance) | ExampleSet (//311Service/Process/updated)

Table View | Plot View

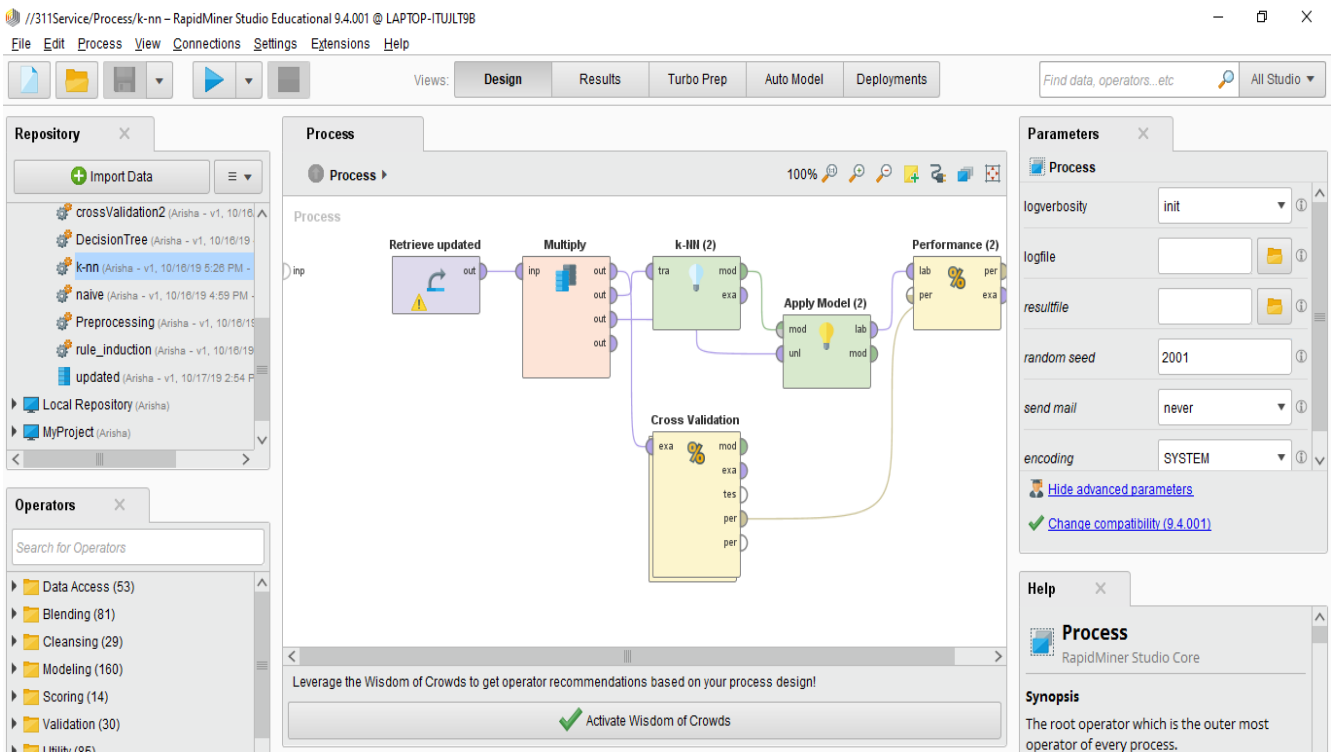
accuracy: 70.55%

	true 1	true 0	class precision
pred. 1	103	43	70.55%
pred. 0	0	0	0.00%
class recall	100.00%	0.00%	

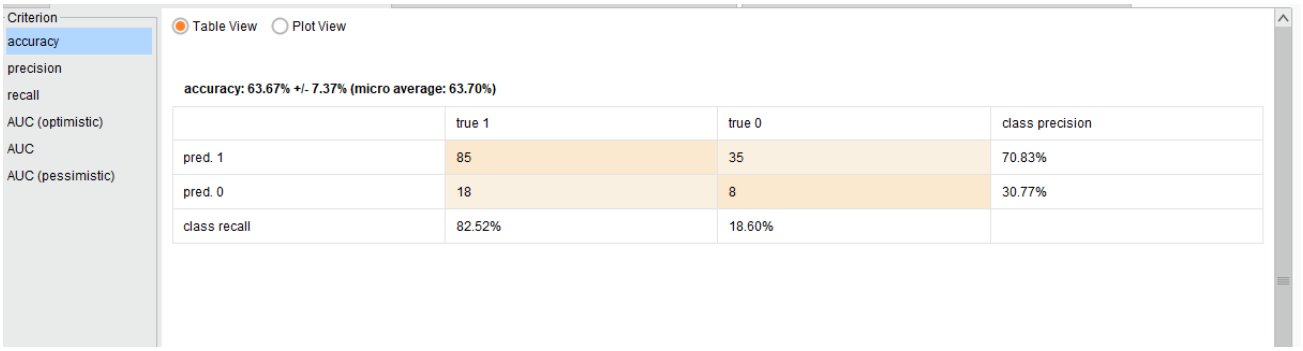
2. K Nearest Neighbour

K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data



KNN Model Accuracy/ Confusion Matrix:



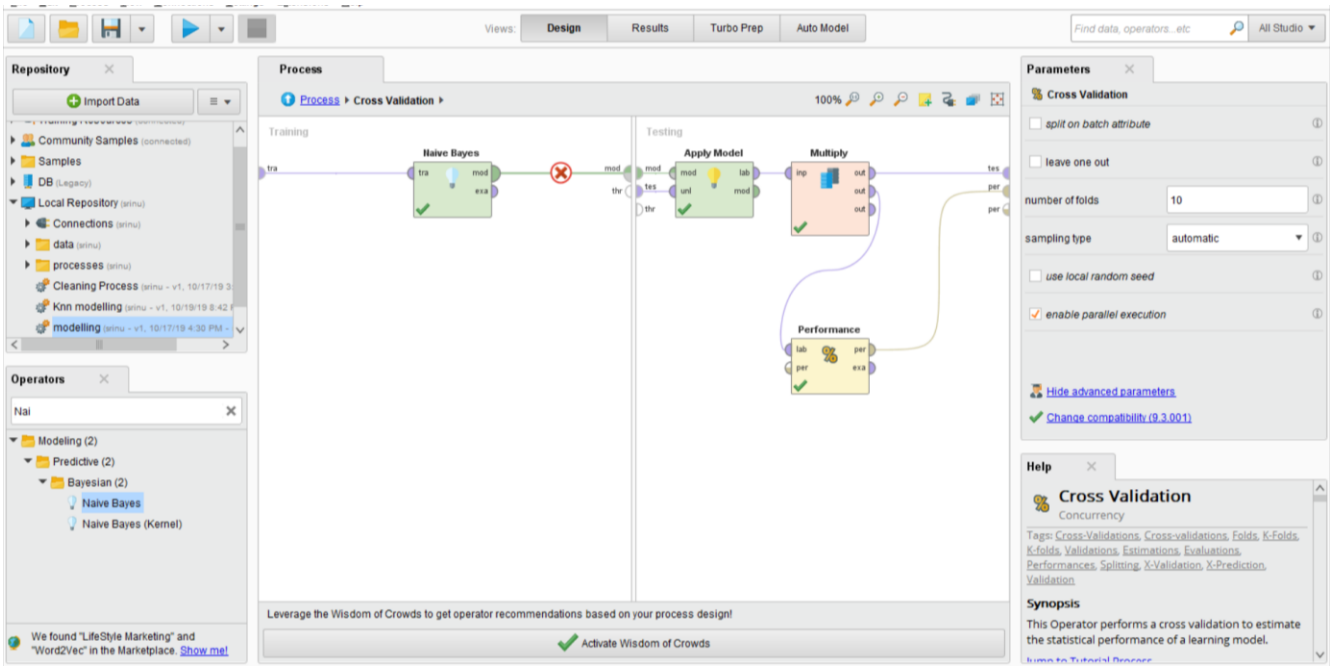
	true 1	true 0	class precision
pred. 1	85	35	70.83%
pred. 0	18	8	30.77%
class recall	82.52%	18.60%	

3. Naïve Bayes:

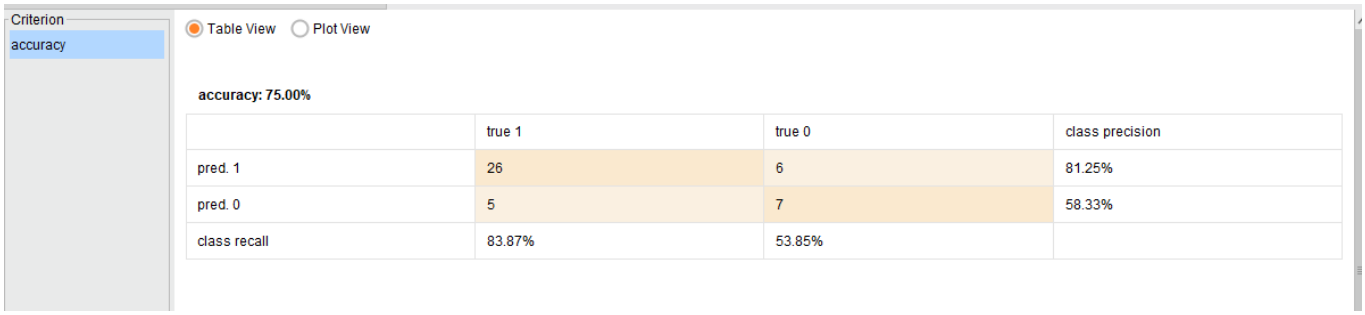
In machine learning, naïve Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong independence assumptions between the features.

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression,[4]:718 which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.



Naivie Bayes Accuracy / Confusion Matrix:



Criterion

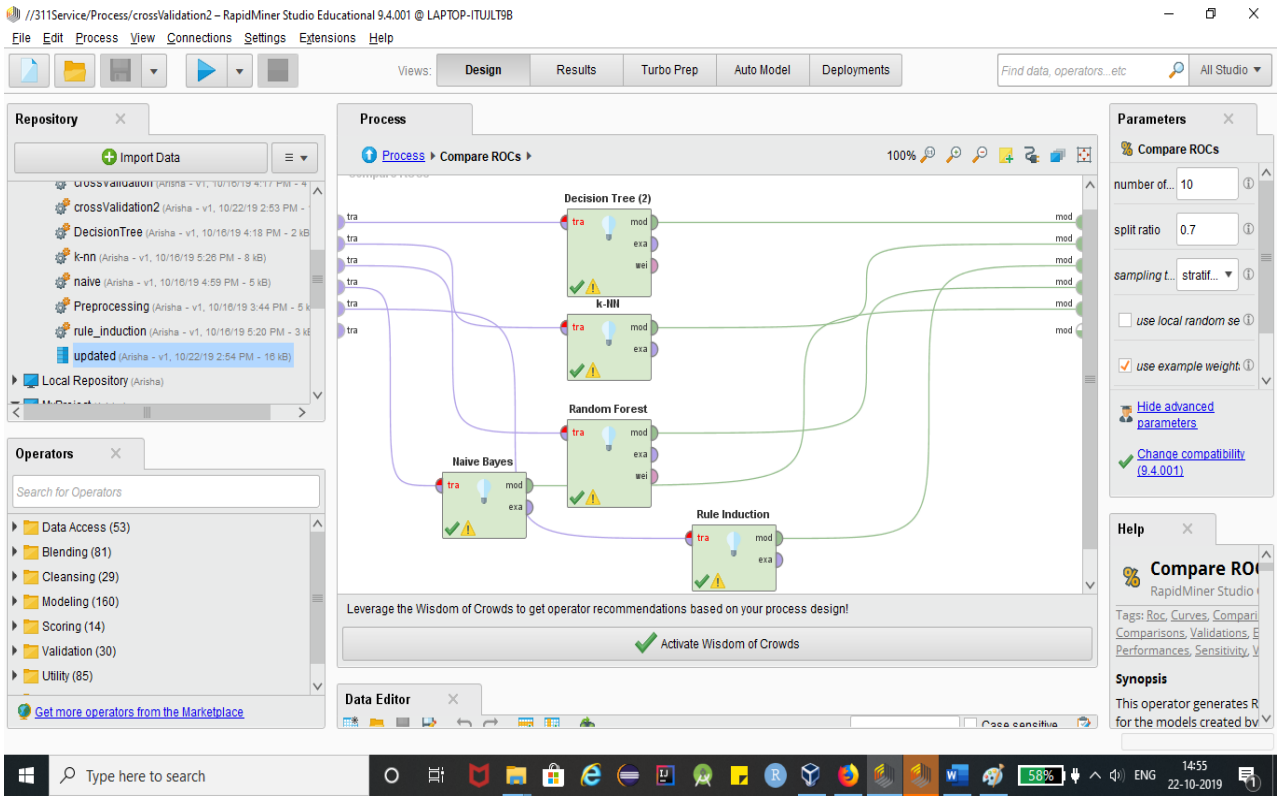
accuracy

Table View Plot View

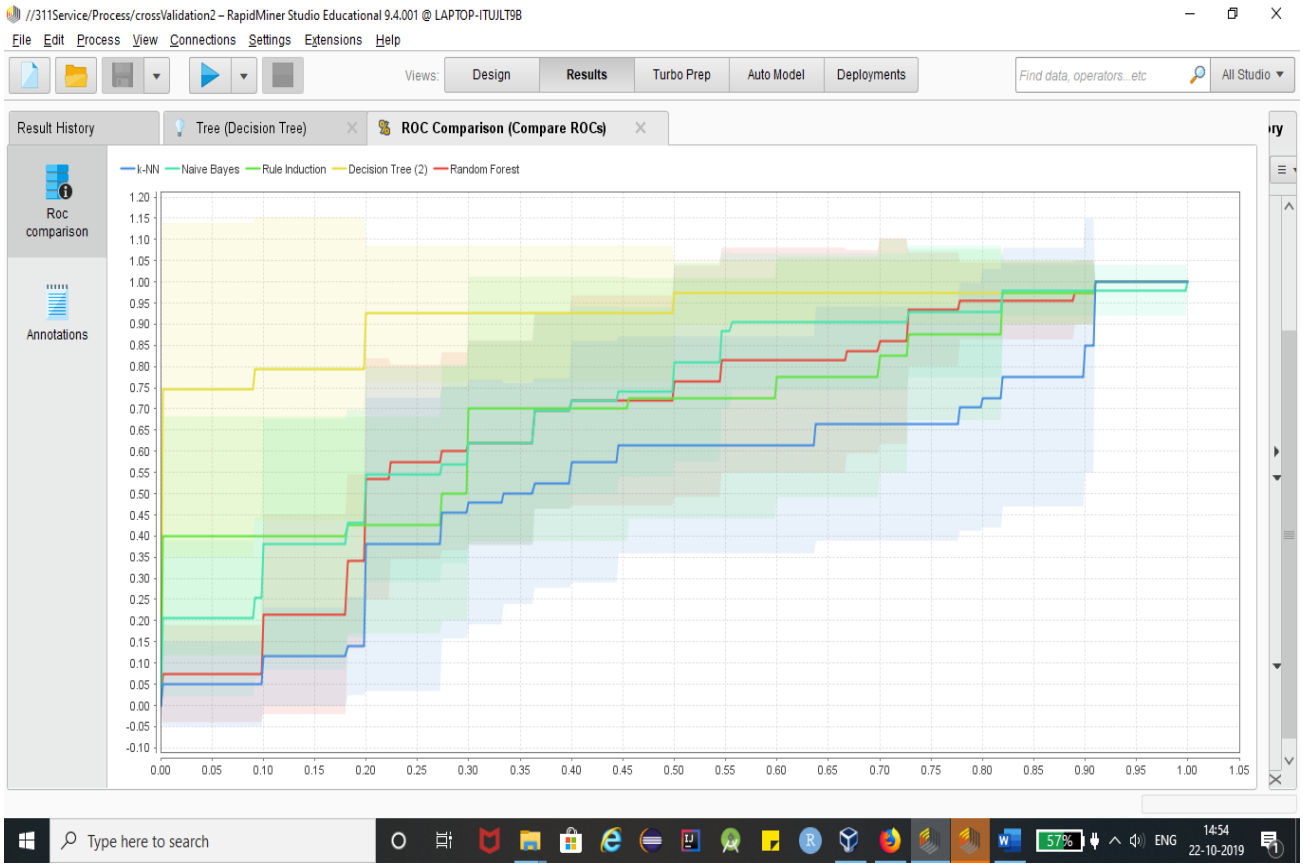
accuracy: 75.00%

	true 1	true 0	class precision
pred. 1	26	6	81.25%
pred. 0	5	7	58.33%
class recall	83.87%	53.85%	

Cross Validation ROC Comparison :



Roc Results:



Conclusion: By the given ROC graph and the confusion matrix, we can conclude that Naïvie Bayes yields better results with the accuracy of 75%.