**Predictions using machine learning**

**Summer Internship**

at C-DAC , pune

(Centre for development of advance computing)

Westend sector III

Aundh, Pune-411007

**PERIOD OF TRAINING**

**(25th MAY – 25th JUNE 2016)**

**Submitted By :**

**ANKITA SINHA**

**ROLL NO:- 13/CS/49**

**B.Tech**

**(Computer Science and Engineering)**

**NIT DURGAPUR**

**ACKNOWLEDGEMENTS**

I would like to take this opportunity to thank the training department of C-DAC, Pune for granting me this golden opportunity to be a part of this esteemed organization as a summer intern . I would like to thank Mr. Aditya kumar Sinha Principal Technical Officer (ACTS and HPCS group) for his constant guidance. I would like to thank Mr Amin Nagiwale and Mrs Shweta(C-DAC ACTS) for their constant assistance provided during the time of this internship.

I would also like to thank Prof. T Dey, Head of the Department of Computer Science and Engineering , and Prof. Mamta Dalui , my project mentor , National Institute of Technology, Durgapur for permitting me to undergo this internship. Last but not the least , I would like to thank my parents for their constant support and guidance.

**ABOUT C-DAC**

Centre for Development of Advanced Computing (C-DAC), Pune occupies a special place in the evolution of the organization as a premier hub for cutting edge R&D . Bestowed with the distinction of being the first C-DAC centre to be established in the country , C-DAC, Pune has been at the forefront of the organizations R&D initiatives and spearheading several national programmes of strategic importance. C-DAC, Pune is credited for the first indigenously developed PARAM supercomputer and establishing the nations credentials as an enabler of advanced technologies. While placing India on the select world map of supercomputing nations, C-DAC, Pune has continued to pioneer the open frame architecture to deliver PARAM Yuva, which was also India's fastest supercomputer and rated 64 among world's top supercomputers.

C-DAC, Pune is also recognized for promoting the concept of multilingual computing in the country to take IT to the grassroots level by defining the standards for the adoption of Indian languages on computers. Since then, the centre has made great strides in this arena through products and technologies that have created a new platform for multilingual users in India as well as abroad.

The expertise garnered through the years of experience has also led C-DAC, Pune to diversify its activities to other domains of advanced R&D namely geoma tics , human-centred design & computing, health informatics, and education & training. This in turn has elevated C-DAC, Pune into the role of a mentor and an incubator of innovation, in the greater national interest.

***INDEX***

**1 Introduction 5**

1.1 Machine learning 5

1.2 Rapid Miner. 6

**2 Classifying check-ins on the basis of timestamp**

2.1 Introduction. 7

2.2 Data set. 8

2.3 Pre-processing of data 9

2.4 Models 11

2.4.1 K-nearest neighbour 11

2.4.2 Clustering (dbscan) 14

2.5 Conclusion 17

2.6 Future 17

**3 Performance measure of different classification algorithms**

**on iris dataset 18**

3.1 Dataset 18

3.2 Pre-processing 18

3.3 Model 19

3.4.1 K-nearest neighbour 20

3.4.2 Naive-Bayes 21

3.4.3 Decision Trees 22

3.4.3 Random Forest 23

3.4 Conclusion 24

**4 Conclusion 24**

**Introduction**

**Machine learning** is a type of artificial intelligence ([AI](http://searchcio.techtarget.com/definition/AI)) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data.

N machine learning the system searches through data to look for patterns. Machine learning uses data to detect patterns in data and adjust program actions accordingly. Machine learning algorithms are often categorized as being supervised or unsupervised. Supervised algorithms can apply what has been learned in the past to new data. Unsupervised algorithms can draw inferences from datasets.

Machine learning algorithms are describes as learning a target function (f) that best maps input variables to an output variable(Y)

Y=f(X)

Machine learning algorithms are divided into

1)

a) **Parametric machine learning algorithms** which make large assumptions about mapping of input variables into output variables and are thus faster to train and require less data but are less powerful.

b)**Non**- **Parametric machine learning algorithms** make few or no assumptions about the target function and in turn require a lot more data , are slower to train and have higher model complexity but result in powerful models.

2)

a) **Supervised machine learning algorithms** - All data is labelled and and algorithms learn to predict output from input data.

b)**Unsupervised machine learning algorithms** -All data is unlabelled and algorithms learn to inherent structure from input data.

The prediction error from any machine learning algorithm can be broken into three parts

1) **Bias** – these are simplifying assumptions to make target function easier to approximate.

2) **Variance**- amount by which estimate of target function changes to diff training data.

(Increasing bias decreases variance this is called trade-off)

3) **Irreducible error-** This error may be caused due to not having sufficient attributes

to sufficiently characterize the best mapping from X to Y.

When model trains data too well. Model learns the details and noise of the data as well to the extent that it negatively affects performance. This is called over fitting of data. Best fit is just when error on data set starts to increase n model has good skill on both test and training data set. To limit over fitting use resampling technique- train & test algo on k diff subsets of training data and build up an estimate on unseen data.

The tool we used for applying machine learning algorithms was **RAPID MINER**. **RapidMiner Studio** is an open source code-free environment for designing advanced analytic processes with machine learning, data mining, text mining, predictive analytics and business analytics. RapidMiner uses a client/server model with the server offered as Software as a service or on cloud infrastructures.

According to Bloor Research, RapidMiner provides 99% of an advanced analytical solution through template-based frameworks that speed delivery and reduce errors by nearly eliminating the need to write code. RapidMiner provides data miming and machine learning procedures including: data loading and transformation (Extract, Transform, Load (ETL)), data preprocessing and visualization, predictive analytics and statistical modelling, evaluation, and deployment. RapidMiner is written in the Java programming language. RapidMiner provides a GUI to design and execute analytical workflows. Those workflows are called “Process” in RapidMiner and they consist of multiple “Operators”. Each operator is performing a single task within the process and the output of each operator forms the input of the next one. Alternatively, the engine can be called from other programs or used as an API. Individual functions can be called from the command line. RapidMiner provides learning schemes, models and algorithms and can be extended using R and python scripts

RapidMiner functionality can be extended with additional plugins which are made available via RapidMiner Marketplace. The RapidMiner Marketplace provides a platform for developers to create data analysis algorithms and publish them to the community.

Rapid miner is used to Streamline Data Prep, Rapidly Build and Train Predictive Models, Validate Model Performance

**CLASSIFYING CHECK-INS FROM SOCIAL SITES ON BASIS OF POPULARITY AND TIME**

**Introduction**

When individuals check-in to different places they upload that information on social networking sites like facebook.This data can be collected and then used to find places frequented by people, the time of the day most places are visited etc.This can be used to boost the tourism of a place and help the tourist by telling them the popular places , the type of crowd there, the time of the day it is most visited etc.

The following paper explores the possibility of boosting the tourism department by providing the tourists with the details of the most visited places for them to know the culture of the place.

We use the tool RapidMiner for cleaning and prediction using machine learning algorithm.

**Dataset**

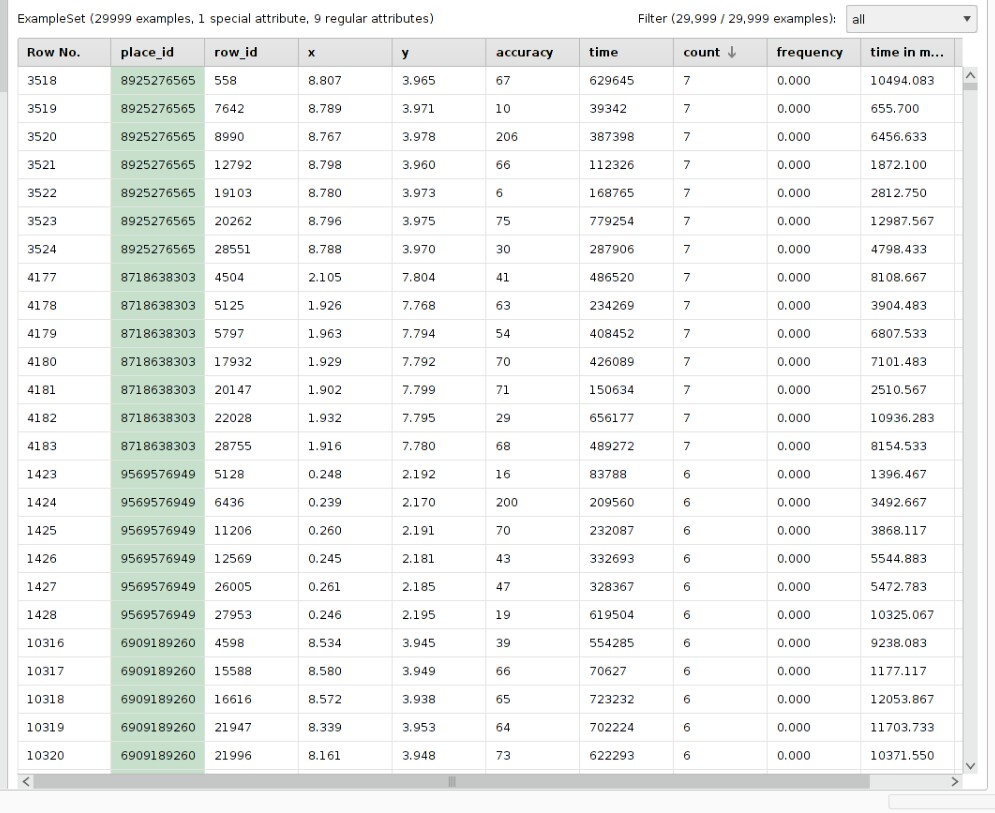
We took the data set consisting of:

more than 30,000 places located in a 10 km by 10 km square.

It has 6 regular rows. All the values are numeric. Some of the place\_id are found to be repeating .

Data set contains data which has been corrupted by inaccurate and noisy values due to local signals from mobiles.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **role** | **type** | **definition** |
| row\_id | regular | int | id of the check-in event |
| x | regular | double | Coordinates of the place\_id |
| y | regular | double | Coordinates of the place\_id |
| time | regular | int | time has been considered in minutes. |
| accuracy | regular | int | tells the accuracy of the predicted place\_id in the data set |
| place\_id | label | double | Gives the id of a place |



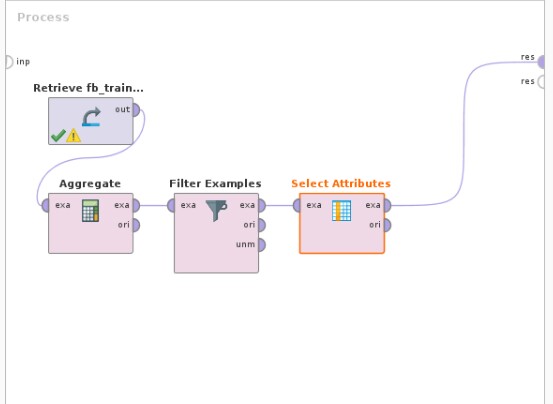
The data set

**Preprocessing the data**

As the data was noisy and inaccurate due to mobile signals we filter the data using the filter examples process which filters the attributes according to the parameters selected and consider only the place\_ids which were more frequent i.e. which occured atleast more than 3 times because those visited less than 3 times were mostly due to noise. We take the average x y and time for these places.The select attributes is used to select the particular attributes on which the label depends and which will be used for prediction.

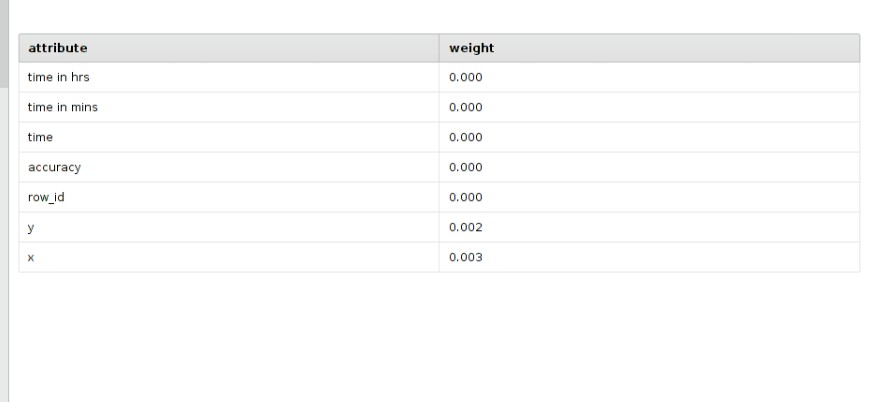
I used aggregate process to count the frequency of place\_ids and filter to filter the place\_ids occurring more than 3 times.

We changed the time stamp to give the time in hours to know at which hour of the day the place was most visited by using generate column and write csv to write it on the data set.



The process for filtering of data.

We then use weight by correlation to find the attributes which affect the label(place\_id) the most . The x y and timestamp attributes affect the place\_id the most.



The weight of differeent attributes.

We conclude that x has a weight of 0.003 and y a weight of 0.002 and only these 2 attributes affect the label.So we use only x and y for making our predictions.

**Models**

1) **Prediction**

The ***k*-Nearest Neighbours algorithm** (or ***k*-NN** for short) is a non-parametric method used for classification

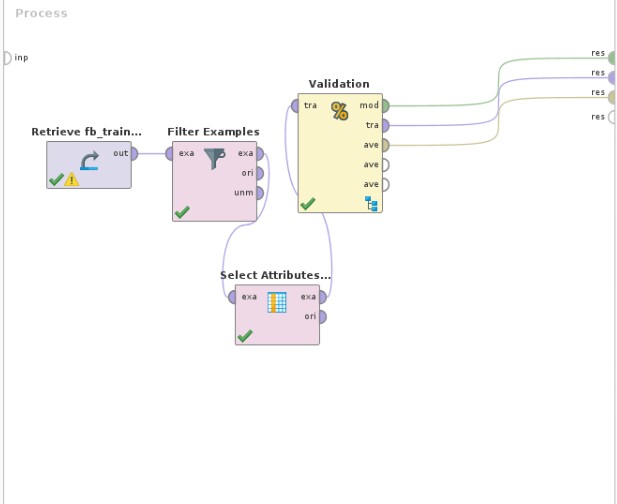
The input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression:

* In *k-NN classification*, the output is a class membership. An object is classified by a majority vote of its neighbours, with the object being assigned to the class
* most common among its *k* nearest neighbours (*k* is a positive [integer](https://en.wikipedia.org/wiki/Integer), typically small). If *k=*1, then the object is simply assigned to the class of that single nearest neighbour.

The neighbours are taken from a set of objects for which the class (for *k*-NN classification) or the object property value (for *k*-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

A shortcoming of the *k*-NN algorithm is that it is sensitive to the local structure of the data

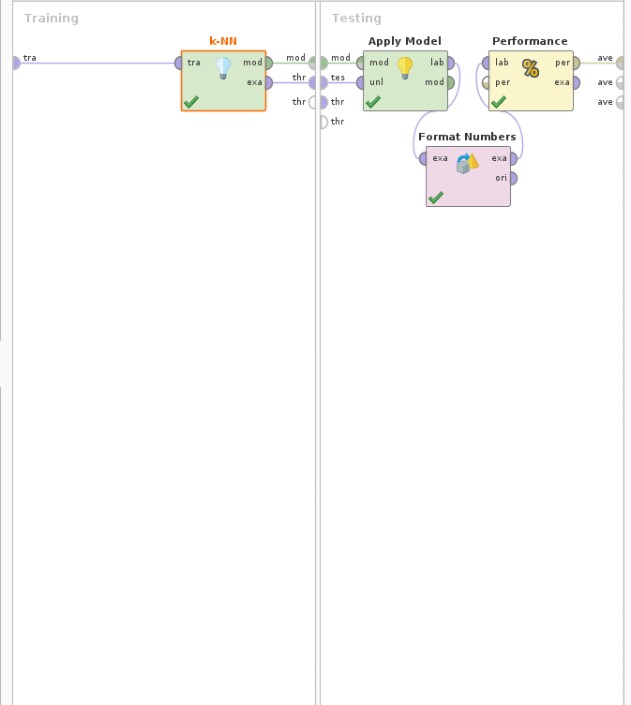
We train **K-Nearest Neighbour algorithm** on the dataset using columns x y as they have the highest weight by correlation to place\_id. To train data first we break the data into a 75% ratio ( using split data operator) and train data using the first 75 % keeping the remaining for testing and validation . We then use the test data to increase the performance of the algorithm.



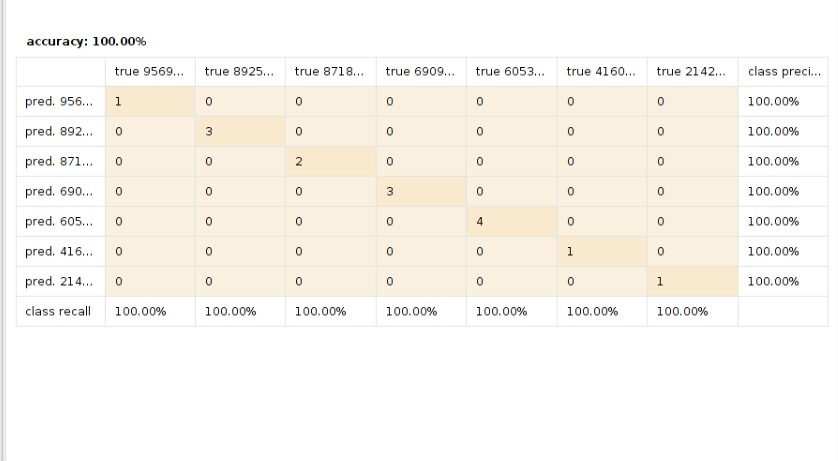
.

The validation process which splits the data into a ratio of 0.75.

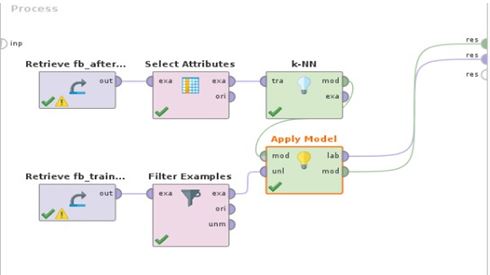
The validation process is used to split the data into 2 parts and keep one part for training the algorithm and the other for testing of the dataset. It finally gives the output as the performance vector of that particular model on the dataset.



The sub-processes of split-validation. The process: performance outputs a confusion matrix telling the accuracy of the model.



The confusion matrix for k-NN. We see from the above image that if we use k=1 we get an accuracy of 100%.

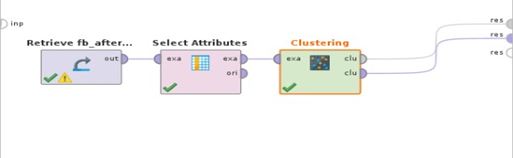


The prediction model for k\_nn.

Apply model is used to apply the trained algo(k-N here ) to apply its learning to the test data set.So finally we use apply model to predict place\_id for the missing place\_ids and for place\_ids that have frequency of less than 3 so as to get the correct place\_id of the place .

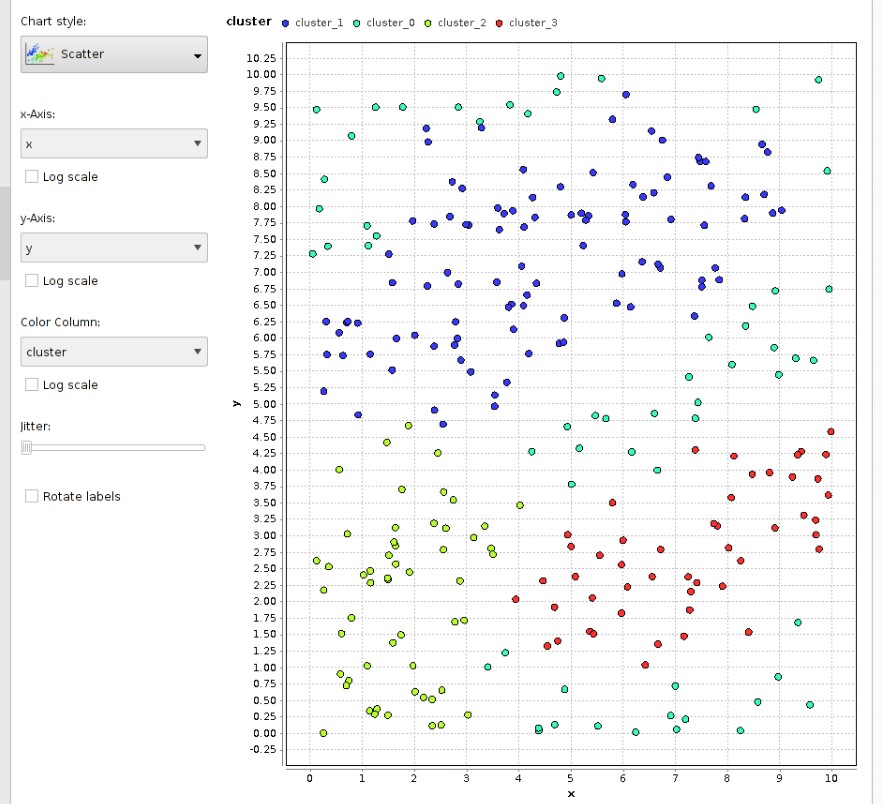
2) **Clustering**

**Density-based spatial clustering of applications with noise** (**DBSCAN**) is a data clustering algorithm. It is a density based algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby query), marking as outliers points that lie alone in low-density regions (whose nearest neighbours are too far away).



Dbscan process

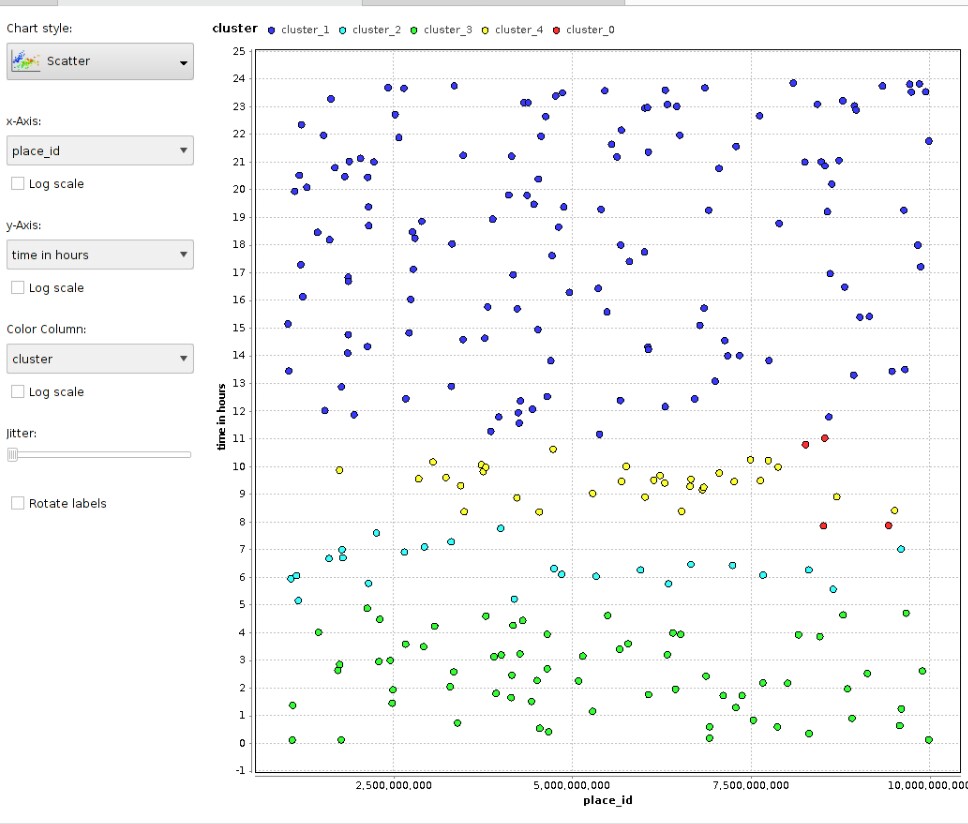
We use **DBSCAN algorithm** to cluster place\_ids based on location(x and y co-orditates ) and also based on timestamp we formatted timestamp to show hour of the day after filtering. DBSCAN cluster automatically clusters the data based on the the density of selected attributes . So it clusters attributes based on timestamp to show the place\_ids visited according to the paticular time of the day.We select the optimim epsilon e=1.0 depends on whether the clusters overlap .



The location wise clustering

We see that cluster 2 is the most densely populated showing that it is the most popular location .

Next to find cluster of most visited localities according to x and y we set x and y as weights and change role of timestamp to id .So the dbscan clusters data based on timestamp.

Time based dbscan.

We see that places in cluster 1 are visited mostly after 12 . Cluster 4 is most densly populated telling that these places are mostly visited from 12 to 3 am.

**Conclusion**

The data set has a number of noisy values so it has to be cleaned first for which we filter the data set by the count of place\_ids and take those who have a count of more than 4 as they may be real place\_ids visited by people. As the data set has limited no. of classes now each having its specific x and y value which forms different specific clusters k-NN can be used for prediction. We have to find the most suitable k by trial and error .The best k is 1 which gives an accuracy of 100% .As performance of k-NN depends mostly on the parameters we should carefully select the parameters on which the label depends on most. So we choose x and y on which the label depends by 0.003 and 0.002 % respectively and use it for learning the dataset. Then using apply model we apply the learning to predict the correct place\_ids for the rest of the places.

Next we use DBSCAN to cluster the data as we need to find the popularity and dbscan automatically divides data into suitable number of clusters according to the density of specified parameters. So first we select x and y attributes and cluster according to that to find the most visited locations. Next we cluster places according to the time after converting it to hours of the day to know at what time the people mostly go to different places.

**Future**

This analysis can be further expanded to create a model to automatically show the popular and most visited places to tourists. If the tourists enter a locality all the places which are frequented by people may it be small unknown dhabas will all be shown to the them. In a new city all the popular localities consisting of maximum number of popular places will be shown. Tourists can know the work culture , party places time related behaviour of the residents and many other things by collecting all the data in one place .This will help them with the most popular hotels, resturants,tourist places, shopping stops etc make their stay at the new place comfortable and making it a beautiful experience.Custering can be used by companies to understand the nature and the habits of its customers to better its products to satisfy the customers need. It can help to more customerize the product to suit the various needs of the customers.

**Performance measure of different classification algorithms on iris dataset**

DATASET

We take the irish data set which has 6 attributes

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **type** | **role** |
| id | nominal | regular |
| label | nominal | label |
| a1 | real | regular |
| a2 | real | regular |
| a3 | real | regular |
| a4 | real | regular |

**Preprocessing the data**

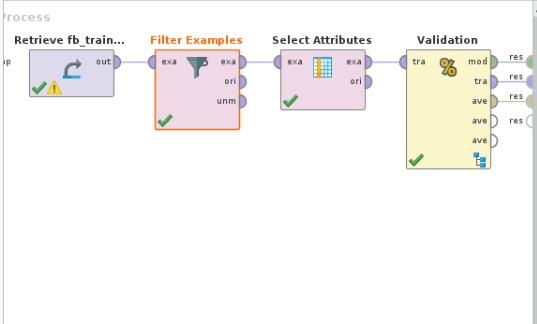
We find the attributes on which the label depends using the process : weight by correlation.



Weight by correlation

We conclude that the label mostly depends on a3 and a4and use these for validation of the label.

**Model**

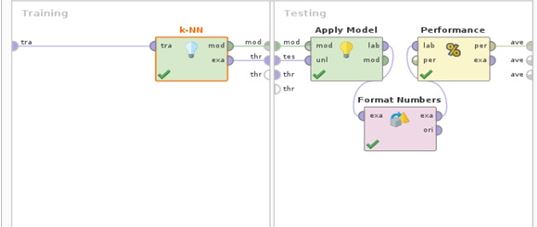


the model for validation .

The filter examples process filters the data and select attributes selects a3 a4 and label for further modelling.

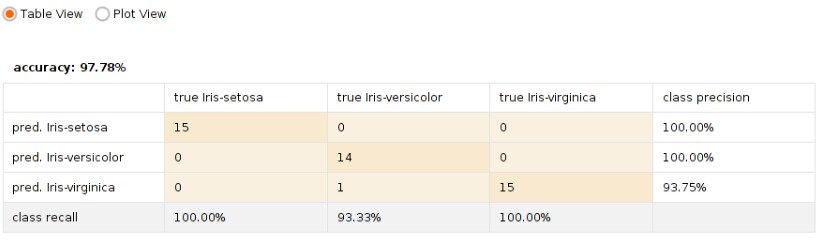
**1) K-nearest neighbour classification**- K-NN makes prediction using the entire data set directly .Predictions are made for a new data point by searching through the entire

Data set for the k most similar instances and summarising the output variables for those instants to determine which of the K instances in the training data set are most similar to a new input a distance measure is used the most common distance measure is Euclidean distance.



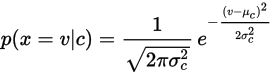
The k-NN sub process of split validation

Format numbers changes the label and the prediction to the nominal data type (discrete data) so that performance(classification) process can work on it.

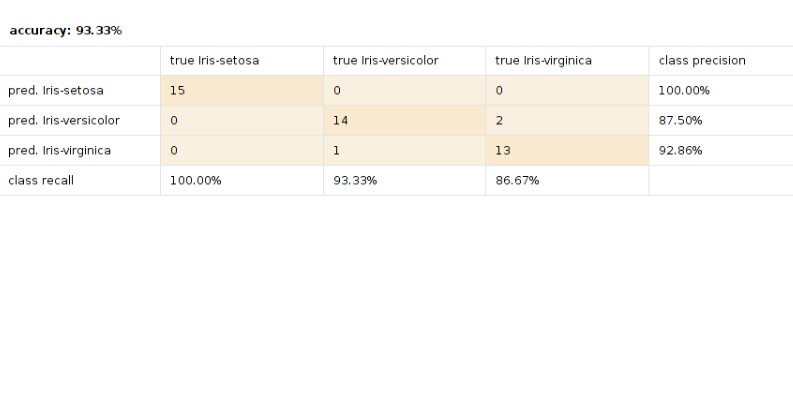


Confusion matrix for k\_nn on iris data set

We conclude that on selection k=1 k-NN gives an accuracy of 97.78% on the iris data-set .

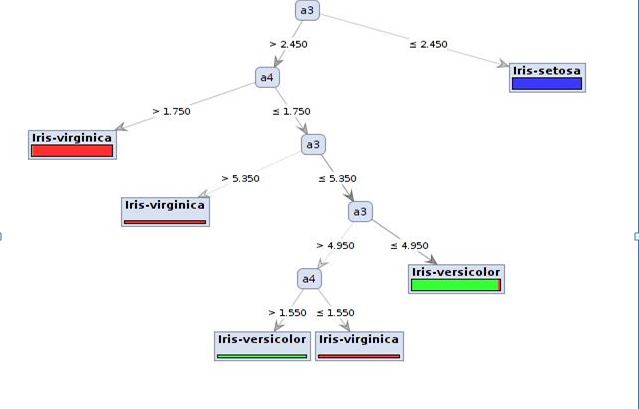
**Naive-Bayes- N**aive Bayes classifier is used for binary and multiclass problems. The representation of nave-Bayes is probability. The Gaussian Naive bayes probability is calculated by 

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of [feature](https://en.wikipedia.org/wiki/Feature_vector) values, where the class labels are drawn from some finite set. It is not a single [algorithm](https://en.wikipedia.org/wiki/Algorithm) for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is [independent](https://en.wikipedia.org/wiki/Independence_(probability_theory)) of the value of any other feature, given the class variable

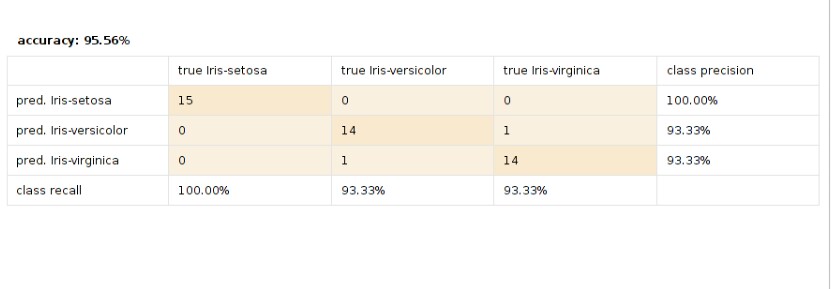
Confusion matrix for naive-bayes on iris data set

The accuracy of naive-bayes on the iris data-set is 93.33% which is less than that of k-NN.

**Decision trees**

Decision trees or classification and regression trees are used for classification of regression predictive modelling problems. It is represented as a binary tree. Predictions are made by traversing the binary tree given a new input data.

The decision tree for iris data set.

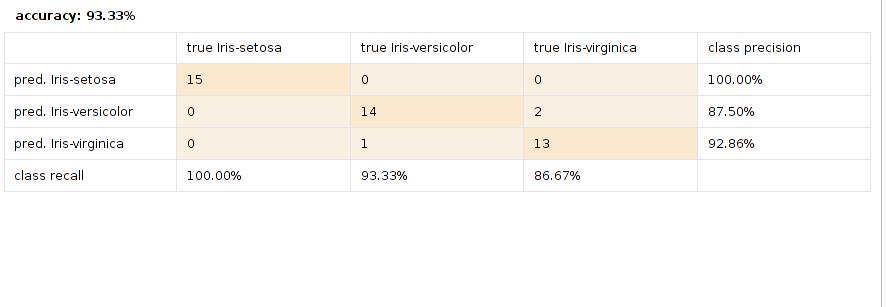
.

Confusion matrix for decision trees on iris dataset

We see that the accuracy is 95.56 % which is better that naive-bayes but not as goos as k-NN .

**Random Forest –**

It is a type of esemble machine learning algorithms. They are type of decision trees but do not use greedy algorithm for pruning. Random forest changes the algorithm in a way that the sub trees are learned so that the resulting predictions from all the subtrees have less correlation. The learning feature is limited to a random sample of features of which to search.

The confusion matrix for random forest taking no . Of subtrees as 5

Random forest is not attribute dependent . So we need not select attributes by weights. It will itself do it. It gives the same result in any combination of parameters.

**Conclusion**

In Iris data set the attributes a3 and a4 affect the label the most. On selecting these attributes and applying various classification algorithms we get different accuracy for each of the model telling us about their performances using the split validation process which splits the data set into a ratio of 0.75 and uses one half as the training data and the other half as the validation data to test the learning model. The performance (classification) process gives the result as the confusion matrix which tells the percent of accuracy for each class . On applying various classification models and testing we find that the k-nearest neighbour algorithm works best on this data set gives an accuracy of 97.78%. But as it is attribute dependent we have to carefully select the attributes on which the label depends so that the algo performs well. Decision trees are the second best with an accuracy percentage of 95.56%. Naive -Bayes and random trees both give an accuracy of 93.33% .