Classifying Churn

Panga Tharun Kumar

25th March, 2019

CONTENTS

GitHub Link for this Repo : --

https://github.com/Tharun-999/CHURN-PREDICTION/blob/master/Loan.ipynb

1 . Introduction

1.1 Problem statement :-

In this problem the main problem statement was the customers usage. company have given data to find the chruns moving or not from the given problem. We have different columns with different data and from the classification of the data it will help company which customers are going to move from company or not. This is done bythe using of machine learning classification problem.

1.2 About Data :-

Data was consist of 3333 instances or rows, 21 types of Attributes. 21st column was class column it consists customers will move or not.

Columns of the test data

columns(1 - 13)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	state	account length	area code	phone number	international plan	voice mail plan	number vmail messages	total day minutes	total day calls	total day charge	total eve minutes	total eve calls	total eve charge
2	HI	101	510	354-8815	no	no	0	70.9	123	12.05	211.9	73	18.01
3	MT	137	510	381-7211	no	no	0	223.6	86	38.01	244.8	139	20.81
4	OH	103	408	411-9481	no	yes	29	294.7	95	50.1	237.3	105	20.17
5	NM	99	415	418-9100	no	no	0	216.8	123	36.86	126.4	88	10.74
6	SC	108	415	413-3643	no	no	0	197.4	78	33.56	124	101	10.54
7	IA	117	415	375-6180	no	no	0	226.5	85	38.51	141.6	68	12.04
8	ND	63	415	348-8073	no	yes	32	218.9	124	37.21	214.3	125	18.22
9	LA	94	408	359-9881	no	no	0	157.5	97	26.78	224.5	112	19.08
10	MO	138	510	353-6954	no	no	0	89.1	117	15.15	126.8	46	10.78
11	TX	128	415	403-4933	no	yes	43	177.8	100	30.23	147.3	89	12.52
12	AR	113	510	360-3811	no	yes	39	209.8	77	35.67	164.1	90	13.95
13	TX	140	415	353-1755	no	no	0	93.2	109	15.84	197.6	116	16.8
14	ME	102	415	372-8233	no	no	0	228.1	86	38.78	156	97	13.26
15	ND	108	415	371-5951	no	no	0	112.6	86	19.14	114.9	101	9.77
16	DE	60	408	381-5937	no	no	0	207.3	77	35.24	207.9	105	17.67
17	MN	96	408	357-1784	no	no	0	208.1	93	35.38	189.2	107	16.08
18	KS	178	415	350-8209	no	yes	22	112.8	66	19.18	232.6	100	19.77
19	MN	75	415	400-5627	no	no	0	225.3	124	38.3	228	81	19.38
20	NC	106	415	365-2473	no	yes	25	169.4	105	28.8	240.5	108	20.44
21	HI	158	510	357-3134	no	no	0	193.3	121	32.86	208.1	97	17.69
22	NV	111	415	396-8198	no	yes	35	161.2	142	27.4	159.1	104	13.52
23	CO	102	510	382-1445	no	no	0	95.6	88	16.25	167.6	106	14.25
24	TN	92	510	391-3827	no	yes	25	79.8	99	13.57	313.6	120	26.66
25	DE	42	415	365-4330	no	yes	31	170.8	101	29.04	233.4	104	19.84
26	OH	69	415	328-6124	no	no	0	229.2	111	38.96	165.3	104	14.05
27	OR	117	415	328-1642	no	yes	38	259.3	94	44.08	245.6	71	20.88
28	NE	76	415	419-9753	no	yes	41	212.6	110	36.14	172.7	97	14.68
29	ID	72	415	413-5754	yes	no	0	101	110	17.17	240	70	20.4
30	WY	115	415	373-8390	yes	yes	6	140.1	125	23.82	157.9	100	13.42

Last column consists of classes

columns(13 - 21)

	P	Q	R	S	T	U
1	total night charge	total intl minutes	total intl calls	total intl charge	number customer service calls	Churn
2	10.62	10.6	3	2.86	3	False.
3	4.24	9.5	7	2.57	0	False.
4	13.51	13.7	6	3.7	1	False.
5	9.93	15.7	2	4.24	1	False.
6	9.2	7.7	4	2.08	2	False.
7	10.04	6.9	5	1.86	1	False.
8	11.71	12.9	3	3.48	1	False.
9	13.99	11.1	6	3	0	False.
10	8.57	9.9	4	2.67	2	False.
11	8.74	11.9	1	3.21	0	False.
12	7.19	9	4	2.43	1	False.
13	9.89	10.5	2	2.84	1	False.
14	10.26	10.6	9	2.86	1	False.
15	8	7.2	6	1.94	3	False.
16	4.87	12.9	5	3.48	1	False.
17	12.58	7.4	2	2	1	False.
18	8.77	14.3	3	3.86	1	False.
19	11.44	11.7	3	3.16	1	False.
20	7.17	13.9	5	3.75	4	False.
21	10.26	7.1	9	1.92	1	False.
22	7.56	14.7	5	3.97	1	False.
23	7.98	9.8	2	2.65	3	False.
24	6.1	9.3	8	2.51	2	False.
25	7.84	11	3	2.97	2	False.
26	10.58	5.2	5	1.4	1	False.
27	12.12	9.2	1	2.48	3	False.
28	8.38	10.1	5	2.73	0	False.
29	15.01	11.1	6	3	0	False.
30	11.22	10.1	3	2.73	1	False.

columns of the train data

columns (1- 13)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	state	account length	area code	phone number	international plan	voice mail plan	number vmail messages	total day minutes	total day calls	total day charge	total eve minutes	total eve calls	total eve charge
2	KS	128	415	382-4657	no	yes	25	265.1	110	45.07	197.4	99	16.78
3	OH	107	415	371-7191	no	yes	26	161.6	123	27.47	195.5	103	16.62
4	NJ	137	415	358-1921	no	no	0	243.4	114	41.38	121.2	110	10.3
5	OH	84	408	375-9999	yes	no	0	299.4	71	50.9	61.9	88	5.26
6	OK	75	415	330-6626	yes	no	0	166.7	113	28.34	148.3	122	12.61
7	AL	118	510	391-8027	yes	no	0	223.4	98	37.98	220.6	101	18.75
8	MA	121	510	355-9993	no	yes	24	218.2	88	37.09	348.5	108	29.62
9	MO	147	415	329-9001	yes	no	0	157	79	26.69	103.1	94	8.76
10	LA	117	408	335-4719	no	no	0	184.5	97	31.37	351.6	80	29.89
11	WV	141	415	330-8173	yes	yes	37	258.6	84	43.96	222	111	18.87
12	IN	65	415	329-6603	no	no	0	129.1	137	21.95	228.5	83	19.42
13	RI	74	415	344-9403	no	no	0	187.7	127	31.91	163.4	148	13.89
14	IA	168	408	363-1107	no	no	0	128.8	96	21.9	104.9	71	8.92
15	MT	95	510	394-8006	no	no	0	156.6	88	26.62	247.6	75	21.05
16	IA	62	415	366-9238	no	no	0	120.7	70	20.52	307.2	76	26.11
17	NY	161	415	351-7269	no	no	0	332.9	67	56.59	317.8	97	27.01
18	ID	85	408	350-8884	no	yes	27	196.4	139	33.39	280.9	90	23.88
19	VT	93	510	386-2923	no	no	0	190.7	114	32.42	218.2	111	18.55
20	VA	76	510	356-2992	no	yes	33	189.7	66	32.25	212.8	65	18.09
21	TX	73	415	373-2782	no	no	0	224.4	90	38.15	159.5	88	13.56
22	FL	147	415	396-5800	no	no	0	155.1	117	26.37	239.7	93	20.37
23	CO	77	408	393-7984	no	no	0	62.4	89	10.61	169.9	121	14.44
24	AZ	130	415	358-1958	no	no	0	183	112	31.11	72.9	99	6.2
25	SC	111	415	350-2565	no	no	0	110.4	103	18.77	137.3	102	11.67
26	VA	132	510	343-4696	no	no	0	81.1	86	13.79	245.2	72	20.84
27	NE	174	415	331-3698	no	no	0	124.3	76	21.13	277.1	112	23.55
28	WY	57	408	357-3817	no	yes	39	213	115	36.21	191.1	112	16.24
29	MT	54	408	418-6412	no	no	0	134.3	73	22.83	155.5	100	13.22
30	MO	20	415	353-2630	no	no	0	190	109	32.3	258.2	84	21.95

Last column consists of the classes

columns(13 – 21)

	M	N	O	P	Q	R	S	T	U
1	total eve charge	total night minutes	total night calls	total night charge	total intl minutes	total intl calls	total intl charge	number customer service calls	Churn
2	16.78	244.7	91	11.01	10	3	2.7	1	False.
3	16.62	254.4	103	11.45	13.7	3	3.7	1	False.
4	10.3	162.6	104	7.32	12.2	5	3.29	0	False.
5	5.26	196.9	89	8.86	6.6	7	1.78	2	False.
6	12.61	186.9	121	8.41	10.1	3	2.73	3	False.
7	18.75	203.9	118	9.18	6.3	6	1.7	0	False.
8	29.62	212.6	118	9.57	7.5	7	2.03	3	False.
9	8.76	211.8	96	9.53	7.1	6	1.92	0	False.
10	29.89	215.8	90	9.71	8.7	4	2.35	1	False.
11	18.87	326.4	97	14.69	11.2	5	3.02	0	False.
12	19.42	208.8	111	9.4	12.7	6	3.43	4	True.
13	13.89	196	94	8.82	9.1	5	2.46	0	False.
14	8.92	141.1	128	6.35	11.2	2	3.02	1	False.
15	21.05	192.3	115	8.65	12.3	5	3.32	3	False.
16	26.11	203	99	9.14	13.1	6	3.54	4	False.
17	27.01	160.6	128	7.23	5.4	9	1.46	4	True.
18	23.88	89.3	75	4.02	13.8	4	3.73	1	False.
19	18.55	129.6	121	5.83	8.1	3	2.19	3	False.
20	18.09	165.7	108	7.46	10	5	2.7	1	False.
21	13.56	192.8	74	8.68	13	2	3.51	1	False.
22	20.37	208.8	133	9.4	10.6	4	2.86	0	False.
23	14.44	209.6	64	9.43	5.7	6	1.54	5	True.
24	6.2	181.8	78	8.18	9.5	19	2.57	0	False.
25	11.67	189.6	105	8.53	7.7	6	2.08	2	False.
26	20.84	237	115	10.67	10.3	2	2.78	0	False.
27	23.55	250.7	115	11.28	15.5	5	4.19	3	False.
28	16.24	182.7	115	8.22	9.5	3	2.57	0	False.
29	13.22	102.1	68	4.59	14.7	4	3.97	3	False.
30	21.95	181.5	102	8.17	6.3	6	1.7	0	False.

Datatypes of the columns in training data

```
In [68]: data_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 21 columns):
state                             3333 non-null object
account length                    3333 non-null int64
area code                         3333 non-null int64
phone number                      3333 non-null object
international plan                3333 non-null object
voice mail plan                   3333 non-null object
number vmail messages             3333 non-null int64
total day minutes                 3333 non-null float64
total day calls                   3333 non-null int64
total day charge                  3333 non-null float64
total eve minutes                 3333 non-null float64
total eve calls                   3333 non-null int64
total eve charge                  3333 non-null float64
total night minutes               3333 non-null float64
total night calls                 3333 non-null int64
total night charge                3333 non-null float64
total intl minutes                3333 non-null float64
total intl calls                  3333 non-null int64
total intl charge                 3333 non-null float64
number customer service calls     3333 non-null int64
Churn                             3333 non-null object
dtypes: float64(8), int64(8), object(5)
memory usage: 546.9+ KB
```

Datatypes for the test data

```
In [72]: data_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1667 entries, 0 to 1666
Data columns (total 16 columns):
account length                   1667 non-null int64
international plan               1667 non-null object
voice mail plan                  1667 non-null object
total day calls                  1667 non-null int64
total day charge                 1667 non-null float64
total eve minutes                1667 non-null float64
total eve calls                  1667 non-null int64
total eve charge                 1667 non-null float64
total night minutes              1667 non-null float64
total night calls                1667 non-null int64
total night charge               1667 non-null float64
total intl minutes               1667 non-null float64
total intl calls                 1667 non-null int64
total intl charge                1667 non-null float64
number customer service calls    1667 non-null int64
Churn                            1667 non-null object
dtypes: float64(7), int64(6), object(3)
memory usage: 208.5+ KB
```

Info function in python will give information of all the data like how many **instances, datetypes**,it contains **null values** or not, **memory usage** of the data

2. Methodology :-

2.1 Data preprocessing

2.1.1 Outlinear analysis:

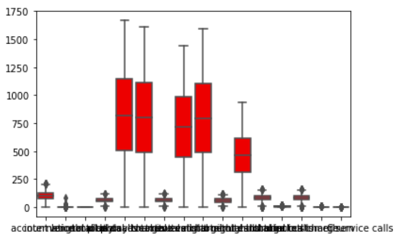
In statistics, an **outlier is** an observation point that is distant from other observations. An **outlier** may be due to variability in the measurement or it may indicate experimental error the latter are sometimes excluded from the data set. An **outlier** can cause serious problems in **statistical analyses**.*O*utlinear analysis will help to find the which is out side of the box. It help us to which points are away from the region of the data. This can find with help of the Box plot in python it in the library of the seaborn.

Train Data

Outlinear Analysis

```
In [29]: import seaborn as sns

In [30]: sns.boxplot(data = data_train, color = 'r')
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0b9af990b8>
```

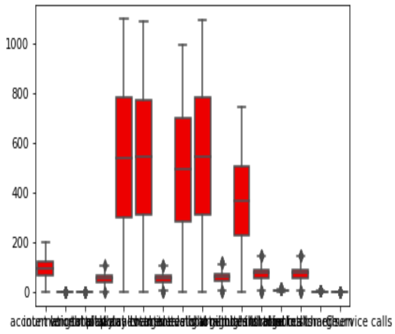


This analysis for the train data

In train data presence of out linear was not high. So we have no need to drop any values.

Test Data

```
In [32]: sns.boxplot(data = data_test, color = 'r')
Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0b974c4358>
```



This boxplot is for test data

In this test data outlinears are not very high so there is no need of dropping or any analysis of the data

2.1.2 Feature Analysis :-

In feature selection we are going to divide the data into the **Features, Labels** . With some size of the data

```
In [15]: X_train = data_train.iloc[:,:-1].values

In [16]: y_train = data_train.iloc[:, -1:].values

In [17]: X_train.shape
Out[17]: (3333, 15)
```

X_train says about training features of the data and it has taken all the rows and all the columns **except** last column

y_train says about training data labels of the data and it has taken all the rows and only with **last** column

```
In [27]: X_test = data_test.iloc[:, :-1].values
         y_test = data_test.iloc[:, -1].values

In [28]: X_test.shape, y_test.shape

Out[28]: ((1667, 15), (1667,))
```

X_test says about training features of the data and it has taken all the rows and all the columns **except** last column

y_test says about training data labels of the data and it has taken all the rows and only with **last** column

Features are used as the charecters of the data and labels are the drug. In the terms of the Medicine.This are most important to the data which column is the label or not to find the class.

2.1.3 Scaling the data :

scaling of the data was the most important in data analysis because in will have large digit numbers and outlinears to bring all of them to one place we use the technique is called scaling. They have two techniques in scaling there are **StandarScaling** and **Normalization** . Both of this will bring any value in to between (0 – 1) it helps to the algorithms to reduce the complexity and increase to compile fast like KNN, SVM etc

$$z = \frac{x_i - \mu}{\sigma}$$

StandardScaler. In standard scaler in data that will decrease by both of the **mean** and **Standard deviation** in the each and every value in the data

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Normalization is used to normalize the data with the **min** and **max** values in the data it will bound in between the (0-1).it is more used for the **Outlinear** data to bring into the **same range** of all data

validation of the data

```
In [36]: # Scaling of the data

In [37]: from sklearn.preprocessing import StandardScaler
         sc = StandardScaler()
         X_train = sc.fit_transform(X_train)
         X_test = sc.transform(X_test)

/home/sai/anaconda3/lib/python3.6/site-packages/sklearn/utils/validation.py:595: DataConversionWarning: Data with input dtype int64 was converted to float64 by StandardScaler.
  warnings.warn(msg, DataConversionWarning)
/home/sai/anaconda3/lib/python3.6/site-packages/sklearn/utils/validation.py:595: DataConversionWarning: Data with input dtype int64 was converted to float64 by StandardScaler.
  warnings.warn(msg, DataConversionWarning)
/home/sai/anaconda3/lib/python3.6/site-packages/sklearn/utils/validation.py:595: DataConversionWarning: Data with input dtype int64 was converted to float64 by StandardScaler.
  warnings.warn(msg, DataConversionWarning)
```

Standard Scaler of the is imported from the preprocessing class in sklearn library

we will fit the data into the formula and then we will transpose it from bigger digits to range of (0-1). we do scaling for only the features of the data because Standard Scaler will only accepts the 2D and more dimensions of the data. But y label was the 1D dimension data.

3 Modeling: -

3.1.1 Model Selection :

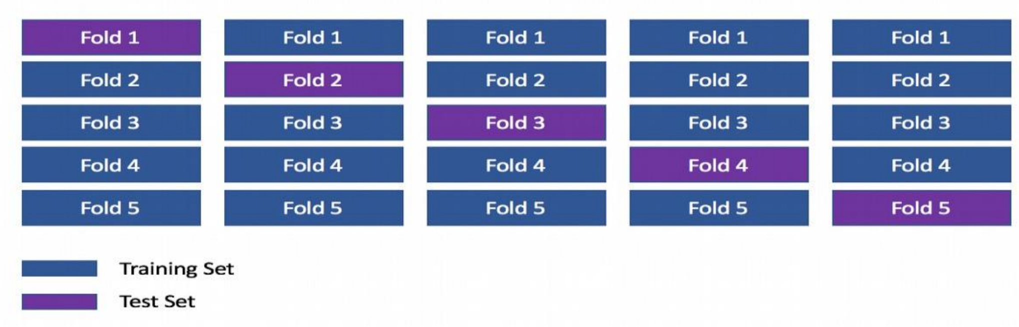
Model selection is class which is used to train the data, test the data, split the data. It has the most popular function **K Folds Cross Score** and **train_test_split** data

Cross Validation K folds

```
In [40]: from sklearn.model_selection import cross_val_score
         cross_val_score(Dtc,X_train, y_train ,cv = 40, n_jobs=-1).mean()

Out[40]: 0.9373011795079478
```

Here cross validation score used to train the data.in baches with k samples and the mean of the data is equal to the Accuracy of the data. We can get more by using this technique.



Train_test_split is another famous function to train and test data it takes features and labels in to the data. And Test or train size of the data to test or train the data

TrainTestSplit

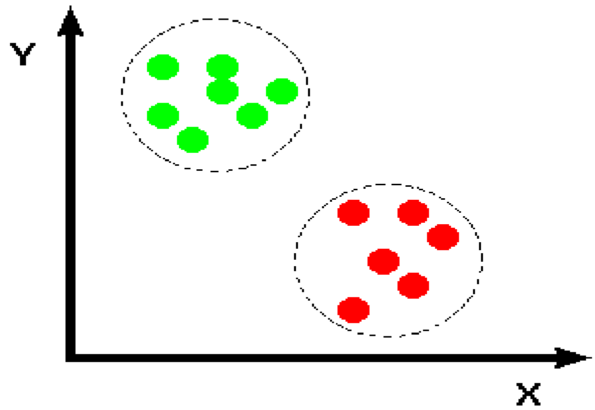
```
In [10]: #TRAINING NAD TESTING THE DATA
         from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
```

This also a famous function in sklearn library to split data into the tarin and test.

3.1.2 Classification:

classification is one of the model in supervised learning. It mainly used for the classification problems

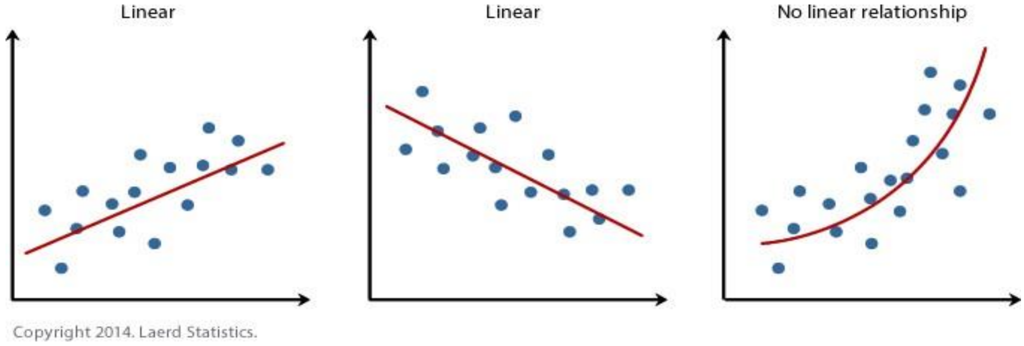
- Classification problems means it will have the classes like the Male, female , boy, girl, 0, 1, true, false etc..
- Classification algorithms are the SVC, DecisionTreeClassifier, RandomForestClassifier etc..
- In classification there will have metrices mainly. Means the confusion matrix, Accuracy score, f1 score etc...
- Classification is used to classifiy the data into the different groups and helps to the algorithms which one belongs to which group



3.1.3 Regression :

Regression is one of the important model in supervised learning .It i mainly used for the Regression problems.

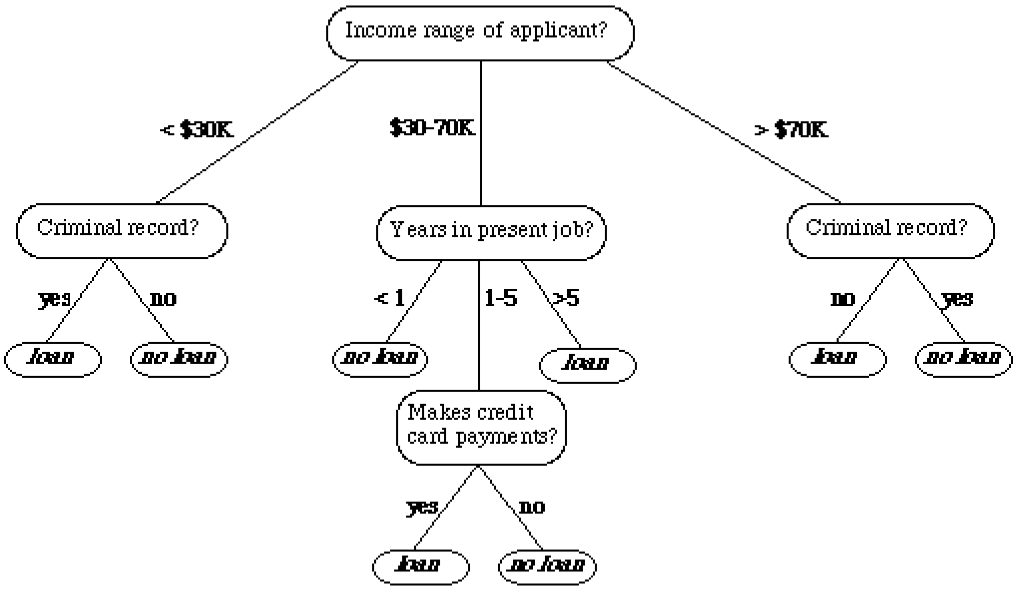
- Regression is mainy used for the prediction of the data like waether prediction , stockmarket, House rate prediction etc...
- Regression algorithms are the SVR, DecisionTreeRegressor, LinearRegression etc....
- In regression there will have the metrics class with functions like mean_square_error, log_mean_square_error, etc...
- This problems mainly used for the prediction of the data



Mainly this problem was going to discuss about the Classification model because at the last column label is the discrete data, so here DecisionTreeClaasifier is using to find the Accuracy or to classifiy the data

3.1.4 Decision Tree :

Decision tree is one of the most popular machine learning algorithms used all along, This story I wanna talk about it so let's get started!!! Decision trees are used for both classification and regression problems, this story we talk about classifications



Decision tree will work like the decision making it help us to find which label is for the give feature

1. Decision tress often mimic the human level thinking so its so simple to understand the data and make some good interpretations.

2. Decision trees actually make you see the logic for the data to interpret(not like black box algorithms like SVM,NN,etc..)

For example : if we are classifying *bank loan* application for a customer, the decision tree may look like this

Decision tree are very powerful there are used for the Bagging and Boosting algorithms they work like Human brains. They are good decision makers.

Information gain **:-**

$$\text{Info(D)}= - \sum_{i=1}^{m} pi \log_2 pi$$

$$\text{Info}_\text{A}\text{(D)}=\sum_{j=1}^{V} \frac{|Dj|}{|D|} \quad \text{X Info(D}_\text{j}\text{)}$$

$$\text{Gain(A)=Info(D)- Info}_\text{A}\text{(D)}$$

GiniIndex :-

$$\text{Gini}_\text{A}\text{(D)}=\frac{|D1|}{|D|}\text{Gini(D}_1\text{)}+ \frac{|D2|}{|D|}\text{Gini(D}_2\text{)}$$

$$\Delta Gini(A) = Gini(D) - Gini_A(D).$$

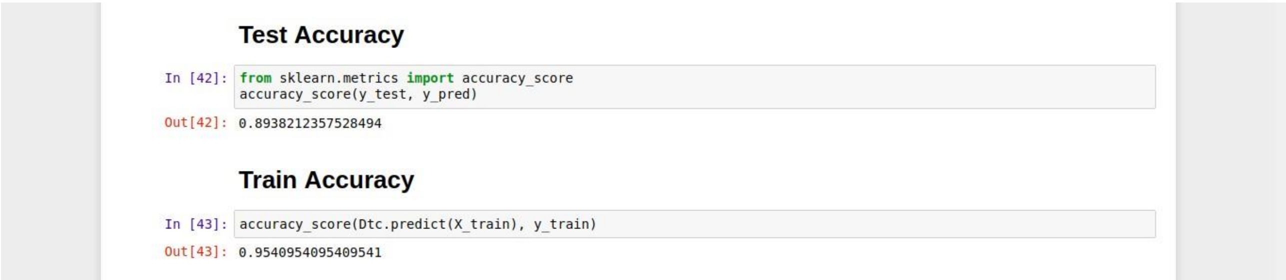
3.CONCLUSION

3.1 Metrics :

The most important thing data science when we are working with the data is Accuracy Score, Confusion Matrix this functions are in the scikit learn library in Metrics class

3.1.1 Accuracy Score :

Accuracy score was the most important in algorithm it say about the how the algorithm was trined with given data. It says about how much learned from the tested data



This figure will says about the how much data was trained and tested by the algorithm.The accuracy score on Test data is 89.38 and on trained data 95.48. We can find the algorithm was overfitted or not by seeing the accuracy scores.

3.1.2 Confusion Matrix :-

A **confusion matrix** is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The **confusion matrix**itself is relatively simple to understand, but the related terminology can be **confusing**.

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

This says the comparision between the true predicted values and the flase predicted values

ConfusionMatrix

```
In [41]: from sklearn.metrics import confusion_matrix
         confusion_matrix(y_test, y_pred)

Out[41]: array([[1399,   44],
                [ 133,   91]])
```

3.1.3 Missclassification values :-

Missclassification says about the how many Actual values are Not equal to the Predicted values

missclassification

```
In [40]: print("missclassification:-", (y_pred != y_test).sum())

         missclassification:- 177
```

Missclassification in this data was the 177 .It will say about the and its accuracy

4. Extra in the python code for the data :-

I tried to increase the accuracy in the data with the help of the values are passing in to the lopp of data

After creating iterations

In [44]:
```python
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
Accu, Miss = [], []
ith,TAccu = [], []
Err, TErr = [], []
n = int(input("enter the number of the Best max_Depth:->>"))
for i in range(1, n): #ITERATIONS OF THE DATA
    ith.append(i)
    classifier = DecisionTreeClassifier(max_depth = i)
    classifier.fit(X_train,y_train) #FITTING THE DATA
    y_pred = classifier.predict(X_test) #PREDICTING THE DATA
    Miss.append((y_test != y_pred).sum()) #FINDING MISSCLASSIIFCATION OF THE DATA
    Accu.append(accuracy_score(y_test,y_pred)) #APPENDING THE ACCURACY SCORE OF THE DATA
    TAccu.append(accuracy_score(y_train, classifier.predict(X_train)))#Training accuracy of the data
    error = 1 - (accuracy_score(y_test,y_pred)) #Erroe for the testing accuracy
    Err.append(error) #Error
    terror = 1 - accuracy_score(y_train, classifier.predict(X_train)) #ERROR for training accuracy
    TErr.append(terror) #Training Error
```

```
enter the number of the Best max_Depth:->>20
```

```
===========================Testing ACCURACIES===================================================
All accuracies of the data:- [0.865626874625075, 0.8620275944811038, 0.8812237552489502, 0.891421715656868 7, 0.899
8200359928015, 0.8938212357528494, 0.8926214757048591, 0.8944211157768446, 0.8938212357528494, 0.8944211157768446,
0.8920215956808638, 0.8878224355128974, 0.886622675464907, 0.8824235152969406, 0.877624475104979, 0.87222555488902
22, 0.8764247150569886, 0.8722255548890222, 0.8692261547690462]
==========================MissClassification====================================================
All Missclassifion of the data:- [224, 230, 198, 181, 167, 177, 179, 176, 177, 176, 180, 187, 189, 196, 204, 213,
206, 213, 218]
==========================Ith-Iteration=========================================================
ith values :- [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
==========================Training Accuracy=====================================================
Training Accuracy of the data:- [0.8679867986798679, 0.8811881188118812, 0.9060906090609061, 0.9231923192319232,
0.9411941194119412, 0.954995499549955, 0.9600960096009601, 0.9636963696369637, 0.9675967596759676, 0.9705970597059
705, 0.9750975097509751, 0.9786978697869787, 0.9843984398439845, 0.9891989198919892, 0.993999399939994, 0.99729972
99729973, 0.9987998799879988, 0.9996999699969997, 1.0]
==========================Testing Error=========================================================
Training Error of the data:- [0.13437312537492496, 0.13797240551889622, 0.11877624475104975, 0.10857828434313133,
0.10017996400719853, 0.10617876424715056, 0.10737852429514094, 0.10557888422315542, 0.10617876424715056, 0.1055788
8422315542, 0.10797840431913619, 0.11217756448710259, 0.11337732453509297, 0.11757648470305937, 0.1223755248950210
2, 0.1277744451109778, 0.1235752849430114, 0.1277744451109778, 0.1307738452309538]
==========================Training Error========================================================
Training Accuracy of the data:- [0.13201320132013206, 0.1188118811881188, 0.0939093909390939, 0.07680768076807676,
0.0588058805880588, 0.045004500450045004, 0.03990399039903991, 0.0363036303630363, 0.03240324032403241, 0.02940294
0294029456, 0.02490249024902491, 0.021302130213021298, 0.015601560156015548, 0.010801080108010841, 0.0060006000600
06023, 0.0027002700270026825, 0.0012001200120012046, 0.0003000300030002734, 0.0]
```

This will give all the information about the missclassification, error, Accuracy

In [47]:
```python
print("===========================Testing ACCURACIES===================================================")
print("All accuracies of the data:-", max(Accu))
print("==========================MissClassification====================================================")
print("All Missclassifion of the data:-", min(Miss))
print("==========================Ith-Iteration=========================================================")
print("ith values :-", ith)
print("==========================Training Accuracy=====================================================")
print("Training Accuracy of the data:-", max(TAccu))
print("==========================Testing Error=========================================================")
print("Training Error of the data:-", min(Err))
print("==========================Training Error========================================================")
print("Training Accuracy of the data:-", min(TErr))
```

```
===========================Testing ACCURACIES===================================================
All accuracies of the data:- 0.8998200359928015
==========================MissClassification====================================================
All Missclassifion of the data:- 167
==========================Ith-Iteration=========================================================
ith values :- [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
==========================Training Accuracy=====================================================
Training Accuracy of the data:- 1.0
==========================Testing Error=========================================================
Training Error of the data:- 0.10017996400719853
==========================Training Error========================================================
Training Accuracy of the data:- 0.0
```

It Gives All the best values from the data

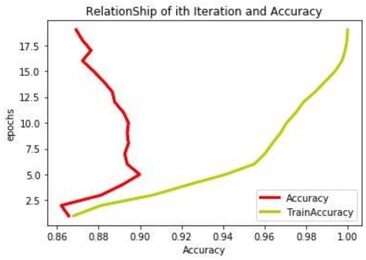
5. *PLOTTING OF THE DATA*

5.1 Accuracies Plotting :-

Plotted with the help of matplotlib in Sklearn Library

Plotting Accuracies

In [49]:
```
plt.plot(Accu, ith, c = 'r', lw = 3,label='Accuracy' )
plt.plot(TAccu, ith, c = 'y',lw = 3, label = "TrainAccuracy")
plt.title("RelationShip of ith Iteration and Accuracy")
plt.xlabel("Accuracy")
plt.ylabel("epochs")
plt.legend()
plt.show()
```

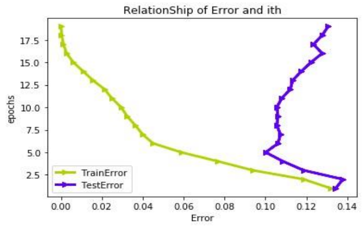


Accuracy plotting The diffrence between the train and the test Accuracy on the data in 20 epochs

5.2 Plotting Error :-

plotting Error

In [50]:
```
plt.plot(TErr, ith, c = 'y',lw = 3, label = "TrainError", marker = ">")
plt.plot(Err, ith, c = 'b',lw = 3, label = "TestError", marker = ">")
plt.title("RelationShip of Error and ith")
plt.xlabel("Error")
plt.ylabel("epochs")
plt.legend()
plt.show()
```



The difference between the train and test in 20 epochs of the data

5.3 Plotting for MissClassification :-

Missclassification says about the data missclassified wile testing with actual values

5.4 Plotiing for HeatMaps :-

Heat maps are mostly drawn with the help of the Seaborn which called as the advance matplotlib library. It helps us to find the correlation between two variables