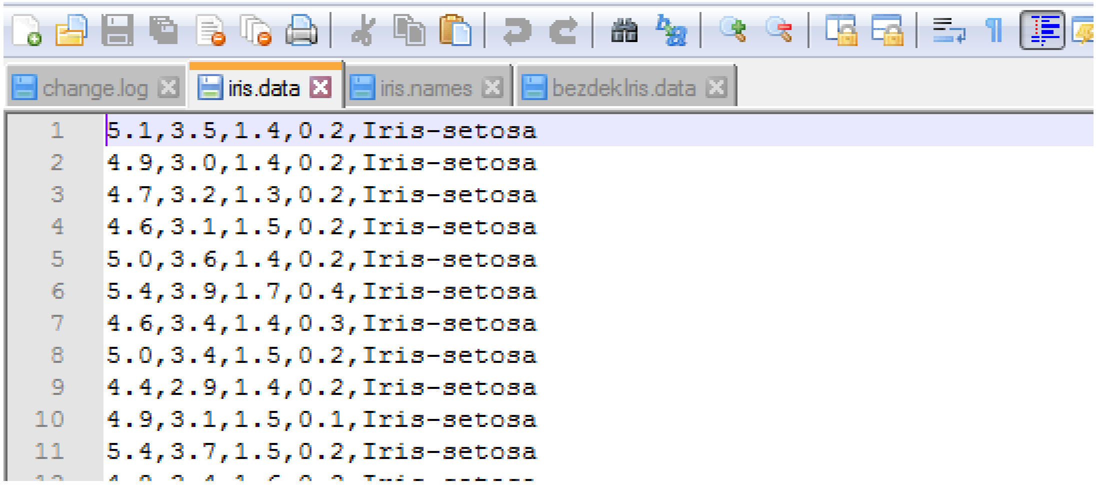
Assignment 1.1

Consider a benchmark toy dataset like Iris1,2. Download the dataset and inspect it manually using a text editor (like Notepad++). Next, inspect Iris using some tool-kits (like R-Studio, Weka GUI, KNIME, Python/Numpy/Pandas/Sk-learn). In context of Iris dataset, provide examples, screen-shots and explain the following in 1-2 short sentences:

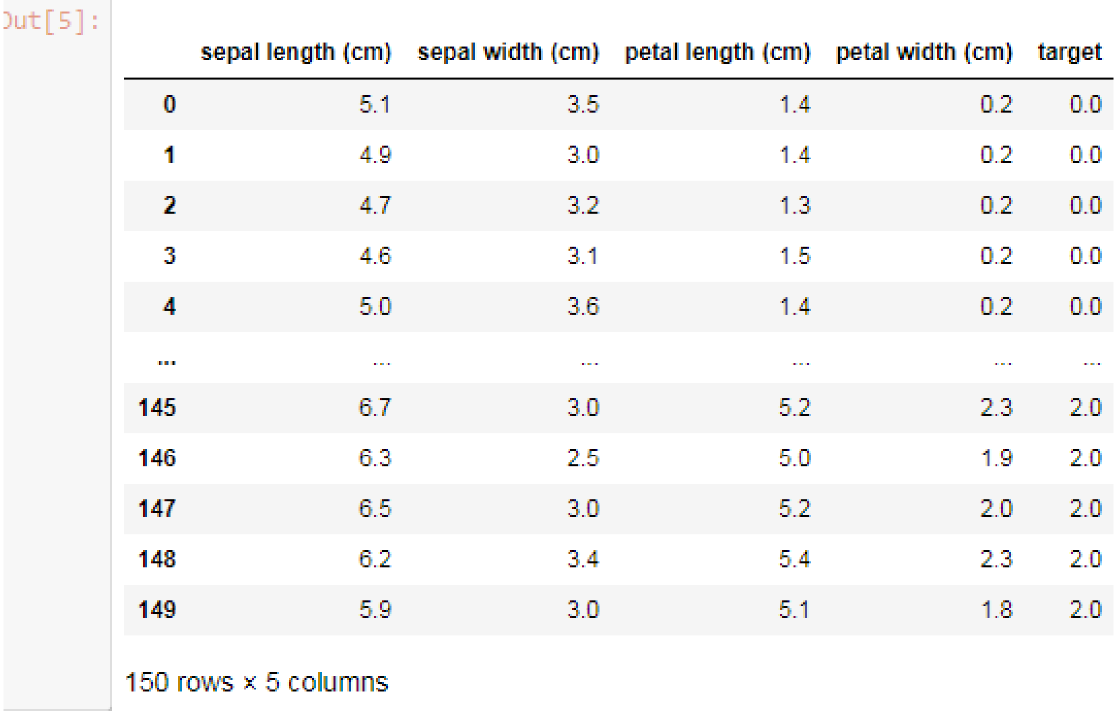
a) How does the raw data look like?

First 10 rows looks like below in notepad++



```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
5.4,3.7,1.5,0.2,Iris-setosa
```

Using jyputer notebooks looks like below after importing from sklearn



	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0.0
1	4.9	3.0	1.4	0.2	0.0
2	4.7	3.2	1.3	0.2	0.0
3	4.6	3.1	1.5	0.2	0.0
4	5.0	3.6	1.4	0.2	0.0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2.0
146	6.3	2.5	5.0	1.9	2.0
147	6.5	3.0	5.2	2.0	2.0
148	6.2	3.4	5.4	2.3	2.0
149	5.9	3.0	5.1	1.8	2.0

150 rows × 5 columns

b) What are Instances, Records, Observations?

All means the same thing, instance is set of items over which concept is defined. Here are 150 in total , 50 in each class. So 1st row shown in notepad++ screenshot above is 1 instance.

b) What are Attributes, Features, Feature Vectors?

Again all means the same thing below are the features taken from iris.names

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
   -- Iris Setosa
   -- Iris Versicolour
   -- Iris Virginica

d) What are Categories, State-of-Nature, Labels, Class-labels, Class, Target, Target-Variables?

Here are 3 classes,
5. class:
   -- Iris Setosa
   -- Iris Versicolour
   -- Iris Virginica

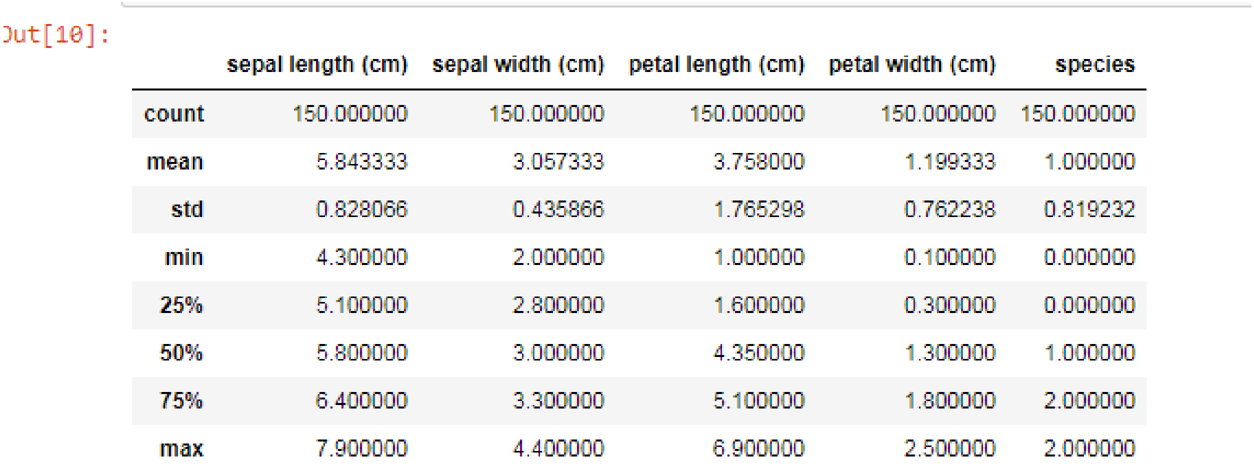
e) What are Explanatory Variables Vs. Response Variables, Dependent Vs. Independent variables?

Dependent/Response variables are the outcome/target variables which values is dependent of the other attributes/features, here speciies. And Independent/explanatory is which doesn't depend on any other feature.

Here all other but class variable is independent.

f) What is meant by distribution of a feature? (like Sepal length as an example)

It tells what kind of data is there in each feature from data types to ranges to min max values etc like below. And if data is normally distributed, if there is erroneous data etc.



Out[10]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333	1.000000
std	0.828066	0.435866	1.765298	0.762238	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

g) What are common methods to visualize more than 3 dimensions? Try PCA on IRIS data, what do you observe?

t-SNE Distributed Stochastic Neighbor Embedding

Diffusion Maps

KernelIPCA

PCA

Heatmaps

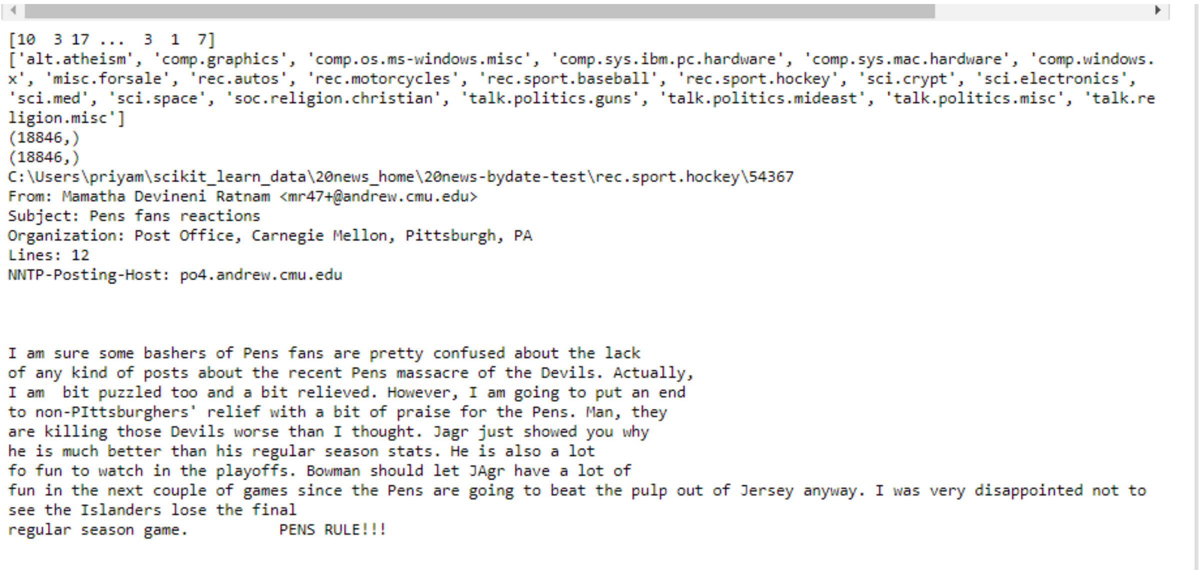
PCA on IRIS-

PCA on IRIS data shows that only 3 features and PCA1 and PCA2 are enough to shows the distribution despite having 4 features.

Assignment 1.2

Let us try to understand vectorization, visualize feature and class distributions. Try to load 20 Newsgroups[3,4] - feel free to use ML tool-kits like R-Console, Weka GUI, Python Sklearn etc. with APIs/methods to load and visualize data. Provide screenshots. Explain the following in 1-2 sentences:

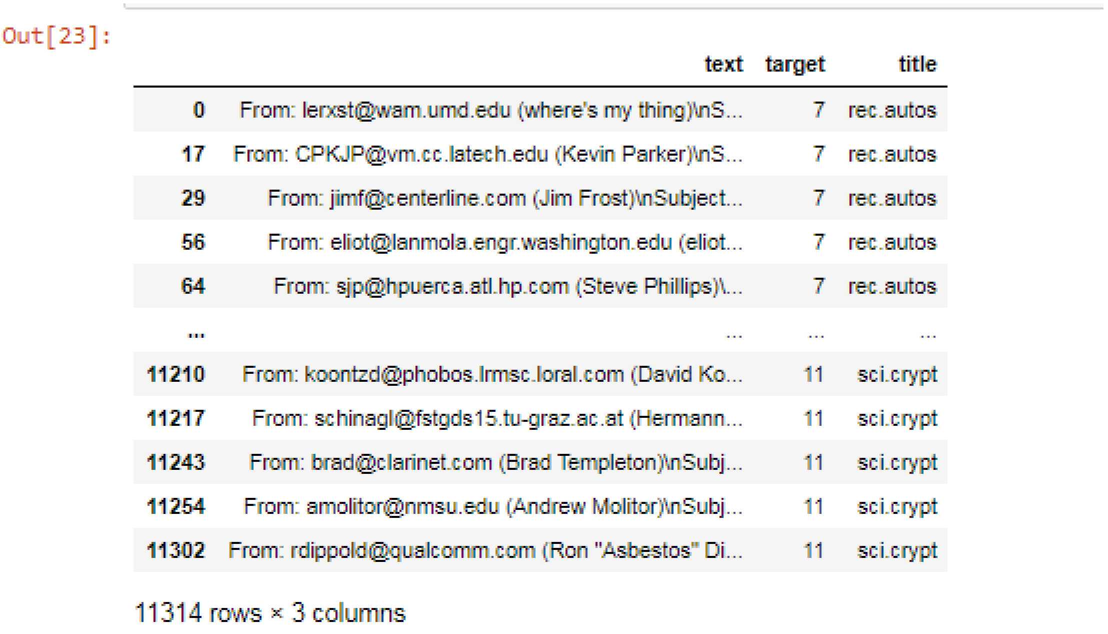
a) How does the raw data look like? Load using any toolkit and view the features and labels.



```
[10  3 17 ...  3  1  7]
['alt.atheism', 'comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'comp.windows.
x', 'misc.forsale', 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey', 'sci.crypt', 'sci.electronics',
'sci.med', 'sci.space', 'soc.religion.christian', 'talk.politics.guns', 'talk.politics.mideast', 'talk.politics.misc', 'talk.re
ligion.misc']
(18846,)
(18846,)
C:\Users\priyam\scikit_learn_data\20news_home\20news-bydate-test\rec.sport.hockey\54367
From: Mamatha Devineni Ratnam <mr47+@andrew.cmu.edu>
Subject: Pens fans reactions
Organization: Post Office, Carnegie Mellon, Pittsburgh, PA
Lines: 12
NNTP-Posting-Host: po4.andrew.cmu.edu



I am sure some bashers of Pens fans are pretty confused about the lack
of any kind of posts about the recent Pens massacre of the Devils. Actually,
I am  bit puzzled too and a bit relieved. However, I am going to put an end
to non-PIttsburghers' relief with a bit of praise for the Pens. Man, they
are killing those Devils worse than I thought. Jagr just showed you why
he is much better than his regular season stats. He is also a lot
fo fun to watch in the playoffs. Bowman should let JAgr have a lot of
fun in the next couple of games since the Pens are going to beat the pulp out of Jersey anyway. I was very disappointed not to
see the Islanders lose the final
regular season game.          PENS RULE!!!
```

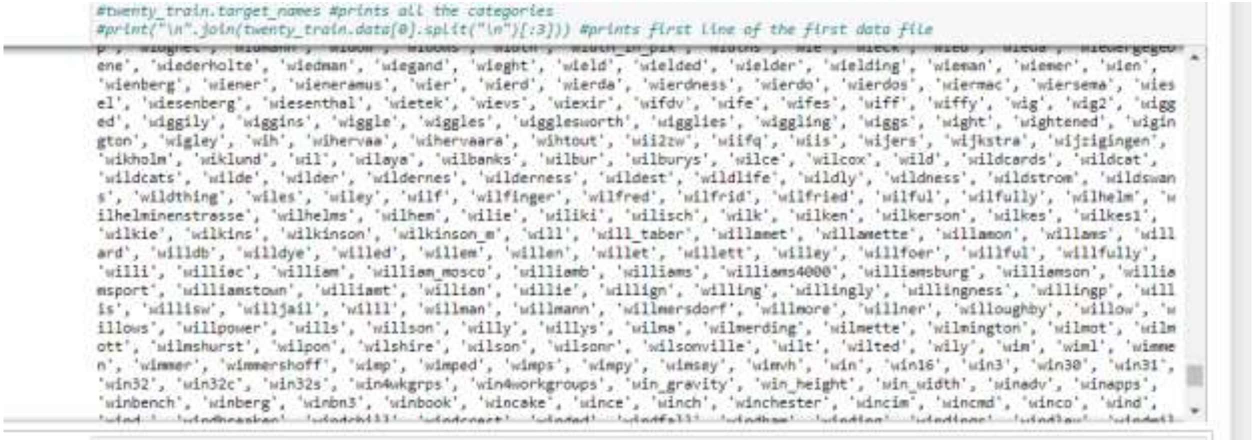


Out[23]:

	text	target	title
0	From: lerxst@wam.umd.edu (where's my thing)\nS...	7	rec.autos
17	From: CPKJP@vm.cc.latech.edu (Kevin Parker)\nS...	7	rec.autos
29	From: jimf@centerline.com (Jim Frost)\nSubject...	7	rec.autos
56	From: eliot@lanmola.engr.washington.edu (eliot...	7	rec.autos
64	From: sjp@hpuerca.atl.hp.com (Steve Phillips)\...	7	rec.autos
...	...	...	...
11210	From: koontzd@phobos.lrmsc.loral.com (David Ko...	11	sci.crypt
11217	From: schinagl@fstgds15.tu-graz.ac.at (Hermann...	11	sci.crypt
11243	From: brad@clarinet.com (Brad Templeton)\nSubj...	11	sci.crypt
11254	From: amolitor@nmsu.edu (Andrew Molitor)\nSubj...	11	sci.crypt
11302	From: rdippold@qualcomm.com (Ron "Asbestos" Di...	11	sci.crypt

11314 rows × 3 columns

Features without vectorization-



```
#twenty_train.target_names #prints all the categories
#print("\n".join(twenty_train.data[0].split("\n")[:3])) #prints first line of the first data file
```

```
ene', 'wiederholte', 'wiedman', 'wiegand', 'wieght', 'wield', 'wielded', 'wielder', 'wielding', 'wieman', 'wiemer', 'wien',
'wienberg', 'wiener', 'wieneramus', 'wier', 'wierd', 'wierda', 'wierdness', 'wierdo', 'wierdos', 'wiermac', 'wiersema', 'wies
el', 'wiesenberg', 'wiesenthal', 'wietek', 'wievs', 'wiexir', 'wifdv', 'wife', 'wifes', 'wiff', 'wiffy', 'wig', 'wig2', 'wigg
ed', 'wiggily', 'wiggins', 'wiggle', 'wiggles', 'wigglesworth', 'wigglies', 'wiggling', 'wiggs', 'wight', 'wightened', 'wigin
gton', 'wigley', 'wih', 'wihervaa', 'wihervaara', 'wihtout', 'wii2zw', 'wiifq', 'wiis', 'wijers', 'wijkstra', 'wijzigingen',
'wikholm', 'wiklund', 'wil', 'wilaya', 'wilbanks', 'wilbur', 'wilburys', 'wilce', 'wilcox', 'wild', 'wildcards', 'wildcat',
'wildcats', 'wilde', 'wilder', 'wildernes', 'wilderness', 'wildest', 'wildlife', 'wildly', 'wildness', 'wildstrom', 'wildswan
s', 'wildthing', 'wiles', 'wiley', 'wilf', 'wilfinger', 'wilfred', 'wilfrid', 'wilfried', 'wilful', 'wilfully', 'wilhelm', 'w
ilhelminenstrasse', 'wilhelms', 'wilhem', 'wilie', 'wiliki', 'wilisch', 'wilk', 'wilken', 'wilkerson', 'wilkes', 'wilkes1',
'wilkie', 'wilkins', 'wilkinson', 'wilkinson_m', 'will', 'will_taber', 'willamet', 'willamette', 'willamon', 'willams', 'will
ard', 'willdb', 'willdye', 'willed', 'willem', 'willen', 'willet', 'willett', 'willey', 'willfoer', 'willful', 'willfully',
'willi', 'williac', 'william', 'william_mosco', 'williamb', 'williams', 'williams4000', 'williamsburg', 'williamson', 'willia
msport', 'williamstown', 'williamt', 'willian', 'willie', 'willign', 'willing', 'willingly', 'willingness', 'willingp', 'will
is', 'willisw', 'willjail', 'willl', 'willman', 'willmann', 'willmersdorf', 'willmore', 'willner', 'willoughby', 'willow', 'w
illows', 'willpower', 'wills', 'willson', 'willy', 'willys', 'wilma', 'wilmerding', 'wilmette', 'wilmington', 'wilmot', 'wilm
ott', 'wilmshurst', 'wilpon', 'wilshire', 'wilson', 'wilsonr', 'wilsonville', 'wilt', 'wilted', 'wily', 'wim', 'wiml', 'wimme
n', 'wimmer', 'wimmershoff', 'wimp', 'wimped', 'wimps', 'wimpy', 'wimsey', 'wimvh', 'win', 'win16', 'win3', 'win30', 'win31',
'win32', 'win32c', 'win32s', 'win4wkgrps', 'win4workgroups', 'win_gravity', 'win_height', 'win_width', 'winadv', 'winapps',
'winbench', 'winberg', 'winbn3', 'winbook', 'wincake', 'wince', 'winch', 'winchester', 'wincim', 'wincmd', 'winco', 'wind',
```

b) How do we convert the 20 Newsgroups raw text to numeric data in rows and columns? Discuss the potential options.

Process is called vectorization. Pandas can be used to but its better to turn each documents to feature vectors. Options available are-
1.Bag of words
2.TFIDF
3.Word2Vec

b) Visualize the popularly known tag-cloud on features or classes.

pc hardware
med politics
hardware sci
crypt sci
politics sci
graphics
windows sci
sci crypt
motorcycles
motorcycles sci
crypt politics
atheism sci
forsale politics
sci space
space sci
politics mideast
hardware windows
autos sci
windows sport
ibm pc
sport hockey
baseball sci
hardware religion
forsale
electronics sci
sci med
hardware politics
mideast sci
christian politics
mac hardware
atheism
med sci
religion sci
religion christian
windows politics
autos
autos politics
windows windows
guns sci
christian sci
windows religion
hardware sport
electronics politics
graphics sci
sport baseball
politics guns
sci electronics
hockey sci
motorcycles sport

d) Explore and look for simple text features like n-grams, or and rare words by IDF values.

For bag of words, n –gram can use n numbers of words as a single token.

IDF values basically is calculated by log(number of documents/number of documents in which word appear) which gives the rare words which are of most importance.

Assignment 1.3

Consider the 20 Newsgroups dataset (you may try these steps[5] or anything similar), explain the following components of a data science pipe-line in context of a classification task (in 1-2 sentences):

a) Explain the goal of a text classification task in 1-2 simple sentences.

Its used to classify a new document to correct category it belongs to.

b) What is meant by preprocessing in this context? Provide examples, discuss potential benefits.

Pre processing includes checking for null values, missing values, filling null values with mean values or standard values etc. it converts text into better format /numeric so that algorithms can perform better.

c) Provide an example of a machine learning model for this task.

There can by many model applied but most used is Naïve bayes, SVM also gave good accuracy.

d) What is a model? How do you represent a model?

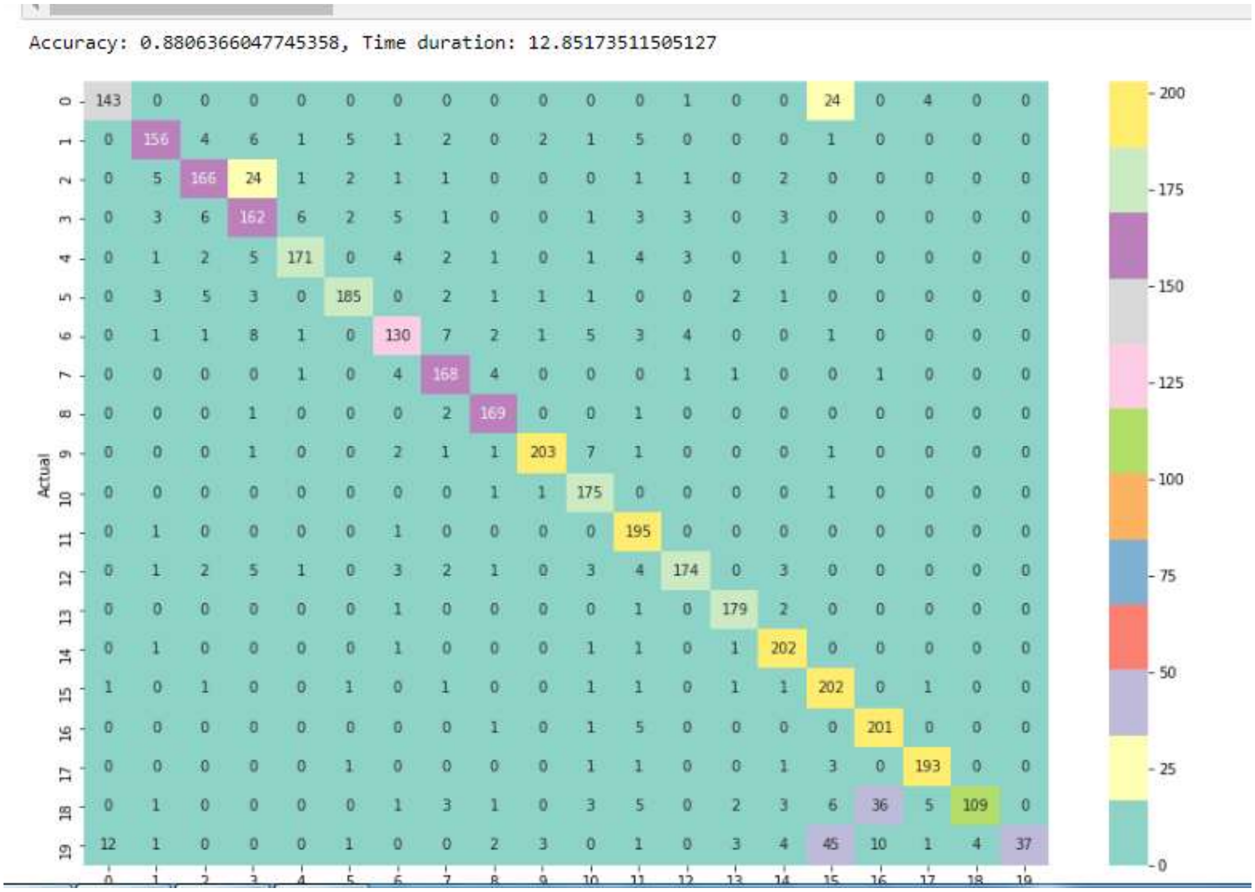
Machine learning model is a complete process in which a learning algorithms can be used to learn from the data and predicting the output on unseen data.

e) What learning method was used by your model? Discuss.

Learning methods depends on the kind of data being learn and output expected, there can be regression, classification, Baye’s learning methods coming under supervised /unsupervised/self-learning.

f) Revisit the goal in the first step, how do we objectively measure if we were able to achieve the goal? (Try to visualize)

We can use confusion matrix, compare accuracy, precision, recall, F1 scores etc of various models and choose which gives the best accuracy.



```
Accuracy: 0.8806366047745358, Time duration: 12.85173511505127
```

Actual	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	143	0	0	0	0	0	0	0	0	0	0	0	1	0	0	24	0	4	0	0
1	0	156	4	6	1	5	1	2	0	2	1	5	0	0	0	1	0	0	0	0
2	0	5	166	24	1	2	1	1	0	0	0	1	1	0	2	0	0	0	0	0
3	0	3	6	162	6	2	5	1	0	0	1	3	3	0	3	0	0	0	0	0
4	0	1	2	5	171	0	4	2	1	0	1	4	3	0	1	0	0	0	0	0
5	0	3	5	3	0	185	0	2	1	1	1	0	0	2	1	0	0	0	0	0
6	0	1	1	8	1	0	130	7	2	1	5	3	4	0	0	1	0	0	0	0
7	0	0	0	0	1	0	4	168	4	0	0	0	1	1	0	0	1	0	0	0
8	0	0	0	1	0	0	0	2	169	0	0	1	0	0	0	0	0	0	0	0
9	0	0	0	1	0	0	2	1	1	203	7	1	0	0	0	1	0	0	0	0
10	0	0	0	0	0	0	0	0	1	1	175	0	0	0	0	1	0	0	0	0
11	0	1	0	0	0	0	1	0	0	0	0	195	0	0	0	0	0	0	0	0
12	0	1	2	5	1	0	3	2	1	0	3	4	174	0	3	0	0	0	0	0
13	0	0	0	0	0	0	1	0	0	0	0	1	0	179	2	0	0	0	0	0
14	0	1	0	0	0	0	1	0	0	0	1	1	0	1	202	0	0	0	0	0
15	1	0	1	0	0	1	0	1	0	0	1	1	0	1	1	202	0	1	0	0
16	0	0	0	0	0	0	0	0	1	0	1	5	0	0	0	0	201	0	0	0
17	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	3	0	193	0	0
18	0	1	0	0	0	0	1	3	1	0	3	5	0	2	3	6	36	5	109	0
19	12	1	0	0	0	1	0	0	2	3	0	1	0	3	4	45	10	1	4	37