**PLEASE NOTE THAT IF YOUR CODE IS IDENTICAL TO SOMEONE ELSE’S CODE – YOU WILL BE REPORTED TO THE ETHICS COMMITTEE AND AUTOMATICALLY FAIL THE EXAM! THIS IS NOT A GROUP PROJECT!**

* 1. Download the dataset (images and metadata) from the following location: <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DBW86T>

Import into your Python/R notebook and create a data-frame by converting image data into pixels and merging with the metadata (ground truth) file. Note the total size of the data is around 2.6GB.

**(WRITE YOUR ANSWER HERE)**

Data processing is Kaggle notebook online and the file will be submitted together with the file.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A picture containing table

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* 1. Assess the quality of data and explain any potential problems with this dataset in the context of clustering.

**(WRITE YOUR ANSWER HERE)**

There is a potential problem on clustering with this dataset (imbalance dataset), since clustering is measure by distance between samples within a certain maximum. In this dataset, 'nv' is a large and unequally scatter class when applying k-means, other not-so-prominent clusters may not be found. In other word, clustering is quite difficult on this dataset.

* 1. Perform exploratory analysis of the dataset and determine if there are any differences between classes

**(WRITE YOUR ANSWER HERE)**

Graphical user interface, text, application, email

Description automatically generated

Chart, bar chart

Description automatically generated

Shape

Description automatically generated with medium confidence

Chart

Description automatically generated with medium confidence Chart, bar chart

Description automatically generated

A picture containing text

Description automatically generated

Chart, bar chart

Description automatically generated

A picture containing diagram

Description automatically generated

Chart, histogram

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

**Part 2 – Complete during the exam time**

* 1. Run K means on the dataset including graphically representing the clusters. What is the appropriate number of clusters to use?

**(WRITE YOUR ANSWER HERE)**

* Running elbow method showing the number of optimal K is around 6 or 7
* Silhouette method showing the number of optimal K is 3

Since we know there are seven labels in the dataset, therefore we use K = 7 and we have the graph below.

|  |  |
| --- | --- |
| Chart, line chart  Description automatically generated |  |
|  | Chart, surface chart  Description automatically generated |

* 1. Assess the quality of your model using appropriate metrics. Please explain your rationale for employing specific metrics you used.

**(WRITE YOUR ANSWER HERE)**

K-means clustering with 7 clusters of sizes 1874, 1061, 1278, 1848, 1728, 1298, 928 and the sum of square by cluster is 50.9%, which means 50.9 % of the data point can be explained by the cluster itself.

Text, letter

Description automatically generated

* 1. Considering that you were given a dataset containing the ground truth, what would be the reason to use unsupervised machine learning rather than supervised.

**(WRITE YOUR ANSWER HERE)**

Ground trust may group the observations with bias as the number of Ks are predefined with human knowledge. The idea of clustering is to calculate to data centroid and grouping the observations into nearest centroids and maximum distance among other clusters. In other words, supervised machine learning may not be able to discovery new cluster.

* 1. Did you get a good accuracy in your model? Which methods can be used to assess the accuracy, and which one did you use, please provide rationale? Create a confusion matrix and assess whether the observations were correctly classified. Comment on what do you think “good” enough result would be?

**(WRITE YOUR ANSWER HERE)**

It is not a good model as the accuracy of the classification version the actual label is low with 9.84% (985/10,015).

Table

Description automatically generated

Confusion matrix created below:

* Image that classified correctly with the label are highlighted in yellow.
* Image no highlighted are clustered in correctly.

Diagram, schematic

Description automatically generated

Furthermore, when taking deeper look into the cluster, each pixel is hard to clearly define belongs to which clusters.

Diagram, box and whisker chart

Description automatically generated

* 1. What are some known limitations of K-means clustering? How can you improve the accuracy of the model? Provide evidence to support your answer.

**(WRITE YOUR ANSWER BELOW THIS LINE)**

Refers to the analysis sections of the data, we realise the number of observations is highly unbalanced (based on the dx grouping).

Chart

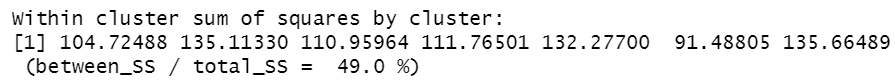
Description automatically generated with medium confidence

To improve the model, we can create a balanced dataset with the following steps:

1. Split the dataset by dx value
2. Define a sampling size, let’s say 100 observations in all data subsets. (Closest value for bcc)
3. Reduce the size of nv, mel, bkl and bcc by randomly remove observations.
4. Increate the size of akiec, vasc and df by manipulate random sample.
5. Then concatenate all seven dataset and perform analysis with 3,500 samples.

Now we create a data subset from the ground\_truth file by keeping 100 observations randomly per label group and rerun the K\_means model again.





The model is still not a good model, even the accuracy has slightly increase by 0.01. 9.85% (69/700).

Calendar

Description automatically generated

* 1. What are some other clustering techniques that you might try and why do you think they may work better? Provide evidence to support your answer.

**(WRITE YOUR ANSWER BELOW THIS LINE)**

K-mean is not ideal to work on nonglobular clusters and clusters of multiple sizes, Compute divisive hierarchical clustering may help of this model. Using Hierarchical clustering on the subset data we have the divise coefficient below which indicates the amount of clustering structure found in the dataset

Text

Description automatically generated with medium confidence

* 1. What are sampling techniques that you might try to improve miss-classification classes? Provide evidence to support your answer.

To improve the miss-classification classes, one of the sampling techniques we can use is cluster centroids. This method uses the clusters of majority class with K-mean algorithms and treat other cluster centroids as the new majority samples.

**THANK YOU!**

**APPENDIX 1 - Python Code**

Python Code:

# %% [code] {"execution":{"iopub.status.busy":"2022-08-13T00:09:49.878838Z","iopub.execute\_input":"2022-08-13T00:09:49.879291Z","iopub.status.idle":"2022-08-13T00:09:50.540549Z","shell.execute\_reply.started":"2022-08-13T00:09:49.879256Z","shell.execute\_reply":"2022-08-13T00:09:50.539124Z"},"jupyter":{"outputs\_hidden":false}}

import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

from PIL import Image

from glob import glob

# %% [markdown]

# # Loading the data

# %% [code] {"execution":{"iopub.status.busy":"2022-08-13T00:09:50.542857Z","iopub.execute\_input":"2022-08-13T00:09:50.543249Z","iopub.status.idle":"2022-08-13T00:09:50.609677Z","shell.execute\_reply.started":"2022-08-13T00:09:50.543215Z","shell.execute\_reply":"2022-08-13T00:09:50.608214Z"},"jupyter":{"outputs\_hidden":false}}

ham=pd.read\_csv(os.path.join('/kaggle/input/skin-cancer-mnist-ham10000','HAM10000\_metadata.csv'))

ham.info()

# %% [code] {"execution":{"iopub.status.busy":"2022-08-13T00:09:50.611684Z","iopub.execute\_input":"2022-08-13T00:09:50.612535Z","iopub.status.idle":"2022-08-13T00:09:52.012877Z","shell.execute\_reply.started":"2022-08-13T00:09:50.612482Z","shell.execute\_reply":"2022-08-13T00:09:52.011464Z"},"jupyter":{"outputs\_hidden":false}}

#Get folder path

base\_skin\_dir = os.path.join('..', 'input')

# loop into folder and loop all the images of the folder

imageid\_path\_dict = {os.path.splitext(os.path.basename(x))[0]: x

for x in glob(os.path.join(base\_skin\_dir,"skin-cancer-mnist-ham10000/", '\*', '\*.jpg'))}

# %% [code] {"execution":{"iopub.status.busy":"2022-08-13T00:09:52.016781Z","iopub.execute\_input":"2022-08-13T00:09:52.017161Z","iopub.status.idle":"2022-08-13T00:14:16.975228Z","shell.execute\_reply.started":"2022-08-13T00:09:52.017127Z","shell.execute\_reply":"2022-08-13T00:14:16.974030Z"},"jupyter":{"outputs\_hidden":false}}

# Adding cell\_type and image\_path columns

ham['path'] = ham['image\_id'].map(imageid\_path\_dict.get)

# Adding image pixels

# Reshape image with 7 x 7

ham['image\_pixel'] = ham['path'].map(lambda x: np.asarray(Image.open(x).resize((7,7))).flatten())

# %% [code] {"execution":{"iopub.status.busy":"2022-08-13T00:14:16.978086Z","iopub.execute\_input":"2022-08-13T00:14:16.978419Z","iopub.status.idle":"2022-08-13T00:14:17.020688Z","shell.execute\_reply.started":"2022-08-13T00:14:16.978386Z","shell.execute\_reply":"2022-08-13T00:14:17.019320Z"},"jupyter":{"outputs\_hidden":false}}

#Check on ham dataframe

ham

# %% [code] {"execution":{"iopub.status.busy":"2022-08-13T00:14:17.022542Z","iopub.execute\_input":"2022-08-13T00:14:17.023113Z","iopub.status.idle":"2022-08-13T00:14:17.846313Z","shell.execute\_reply.started":"2022-08-13T00:14:17.023063Z","shell.execute\_reply":"2022-08-13T00:14:17.844979Z"},"jupyter":{"outputs\_hidden":false}}

# Create image dataframe to store all the pixel columns

# Extract the value from image\_pixe' then split the values into columns

arr = np.array(ham['image\_pixel'].values.tolist())

df = pd.DataFrame(columns=['c'+str(i) for i in range(147)]) # range is 7 x 7 x 3

df[df.columns] = arr

df

# %% [code] {"execution":{"iopub.status.busy":"2022-08-13T00:14:17.848050Z","iopub.execute\_input":"2022-08-13T00:14:17.848539Z","iopub.status.idle":"2022-08-13T00:14:17.860178Z","shell.execute\_reply.started":"2022-08-13T00:14:17.848491Z","shell.execute\_reply":"2022-08-13T00:14:17.859027Z"},"jupyter":{"outputs\_hidden":false}}

# Concat two table as ground\_truth

ground\_truth = pd.concat([ham, df], axis=1)

# %% [code] {"execution":{"iopub.status.busy":"2022-08-13T00:14:17.861352Z","iopub.execute\_input":"2022-08-13T00:14:17.861843Z","iopub.status.idle":"2022-08-13T00:14:17.873029Z","shell.execute\_reply.started":"2022-08-13T00:14:17.861789Z","shell.execute\_reply":"2022-08-13T00:14:17.871465Z"},"jupyter":{"outputs\_hidden":false}}

# Create label mapping on dx column

label\_mapping = {

'nv':0,

'mel':1,

'bkl':2,

'bcc':3,

'akiec':4,

'vasc':5,

'df':6

}

# Create label column based on dx column

ground\_truth['label'] = ground\_truth['dx'].map(label\_mapping)

# %% [code] {"execution":{"iopub.status.busy":"2022-08-13T00:14:17.874819Z","iopub.execute\_input":"2022-08-13T00:14:17.875405Z","iopub.status.idle":"2022-08-13T00:14:17.912201Z","shell.execute\_reply.started":"2022-08-13T00:14:17.875354Z","shell.execute\_reply":"2022-08-13T00:14:17.910805Z"},"jupyter":{"outputs\_hidden":false}}

ground\_truth.head()

# %% [markdown]

# # Descriptive Analysis

#

# There are seven features in the dataset.

# \* Lession ID - Unique ID of Lession

# \* Image ID - Unique ID of the image

# \* DX - DX

# \* DX Type - DX Type

# \* Age - Age of the Patient

# \* Sex - Gender of the Patient

# \* Localization - location of the dermatological images

# %% [code] {"execution":{"iopub.status.busy":"2022-08-13T00:14:17.915786Z","iopub.execute\_input":"2022-08-13T00:14:17.916126Z","iopub.status.idle":"2022-08-13T00:14:17.939976Z","shell.execute\_reply.started":"2022-08-13T00:14:17.916094Z","shell.execute\_reply":"2022-08-13T00:14:17.938612Z"},"jupyter":{"outputs\_hidden":false}}

#Pre-screen the first few observations of the dataset

ham.head()

# %% [markdown]

# ## Perform Null checking on the dataset

#

# Perform NA and Null checking

# %% [code] {"execution":{"iopub.status.busy":"2022-08-13T00:14:17.941638Z","iopub.execute\_input":"2022-08-13T00:14:17.941981Z","iopub.status.idle":"2022-08-13T00:14:17.957489Z","shell.execute\_reply.started":"2022-08-13T00:14:17.941931Z","shell.execute\_reply":"2022-08-13T00:14:17.956174Z"},"jupyter":{"outputs\_hidden":false}}

#NA Check

ham.isna().isna().sum()

# %% [markdown]

# There is 53 records missing in age feature which around 0.6% of the total observations, since age is the only missing value while other features do not have missing.

#

# Therefore we can use mean value to replace the missing value.

# %% [code] {"execution":{"iopub.status.busy":"2022-08-13T00:14:17.959930Z","iopub.execute\_input":"2022-08-13T00:14:17.960933Z","iopub.status.idle":"2022-08-13T00:14:19.663610Z","shell.execute\_reply.started":"2022-08-13T00:14:17.960882Z","shell.execute\_reply":"2022-08-13T00:14:19.662435Z"},"jupyter":{"outputs\_hidden":false}}

## Null Checking

plt.figure(figsize=(6,6))

sns.heatmap(ham.isna().transpose(),

cmap="YlGnBu",

cbar\_kws={'label': 'Missing Data'})

plt.savefig("Missing data check on HAM10000.png", dpi=100)

# Handling missing values - replaced by mean

ham['age'].fillna(value=int(ham['age'].mean()), inplace=True)

# %% [markdown]

# ## Export the dataset

# %% [code] {"execution":{"iopub.status.busy":"2022-08-13T00:14:19.665065Z","iopub.execute\_input":"2022-08-13T00:14:19.665391Z","iopub.status.idle":"2022-08-13T00:14:23.781840Z","shell.execute\_reply.started":"2022-08-13T00:14:19.665358Z","shell.execute\_reply":"2022-08-13T00:14:23.780384Z"},"jupyter":{"outputs\_hidden":false}}

#Export the file into CSV

ground\_truth.to\_csv(r'GroupTruth.csv')

# %% [markdown]

# ## Summary of the dataset

#

# Using the describe function, we have table below. The first column meaning as below:

# \* count - The number of not-empty values.

# \* mean - The average (mean) value.

# \* std - The standard deviation.

# \* min - the minimum value.

# \* 25% - The 25% percentile\*.

# \* 50% - The 50% percentile\*.

# \* 75% - The 75% percentile\*.

# \* max - the maximum value.

#

# Since Age is numerical variable, therefore describe function can provide standard deviation, percentile, min and max information.

# %% [code] {"execution":{"iopub.status.busy":"2022-08-13T00:16:28.390015Z","iopub.execute\_input":"2022-08-13T00:16:28.390455Z","iopub.status.idle":"2022-08-13T00:18:17.832391Z","shell.execute\_reply.started":"2022-08-13T00:16:28.390412Z","shell.execute\_reply":"2022-08-13T00:18:17.830682Z"},"jupyter":{"outputs\_hidden":false}}

#Describe the dataset with all columns

ground\_truth.describe(include='all')

# %% [markdown]

# ## Distribustion of Age

#

# Reading the distribution of age group is show normally distributed graph, which saying majority of the observations are from the age 40 - 55.

# Other indicators can be found:

# \* An upward trend after age 20.

# \* A downward trend after age 55.

# %% [code] {"jupyter":{"outputs\_hidden":false},"execution":{"iopub.status.busy":"2022-08-13T00:35:44.885013Z","iopub.execute\_input":"2022-08-13T00:35:44.885502Z","iopub.status.idle":"2022-08-13T00:35:45.222369Z","shell.execute\_reply.started":"2022-08-13T00:35:44.885464Z","shell.execute\_reply":"2022-08-13T00:35:45.221061Z"}}

#Then plot the graph for visualisation

plt.figure(figsize=(18,6))

plot1 = sns.countplot(data = ground\_truth,x='age',palette="crest")

plot1.set\_xlabel('Age')

plot1.set\_ylabel('Count')

plt.title('Age Distribution');

# %% [markdown]

# # Skin Lesion Class

#

# There are seven classification on cancerous types in the dataset, they are:

# \* Actinic Keratosis (AKIEC)

# \* Basal Cell Carcinoma (BCC)

# \* Benign Keratosis (BKL)

# \* Dermatofibroma (DF)

# \* Melanoma (MEL)

# \* Melanomic Neves (NV)

# \* Vascular Lesion (VASC)

#

# When we try to explore the combination of classification with dx\_type or patient's gender, we can see the dataset is imbalance (the majorities of contributions are from nv class). For examples, dx\_type is follow up, only has nv class observations and no others classes.

# %% [code] {"jupyter":{"outputs\_hidden":false},"execution":{"iopub.status.busy":"2022-08-13T00:36:36.243942Z","iopub.execute\_input":"2022-08-13T00:36:36.245042Z","iopub.status.idle":"2022-08-13T00:36:36.254914Z","shell.execute\_reply.started":"2022-08-13T00:36:36.244946Z","shell.execute\_reply":"2022-08-13T00:36:36.253250Z"}}

print(ground\_truth.dx.value\_counts())

# %% [code] {"jupyter":{"outputs\_hidden":false},"execution":{"iopub.status.busy":"2022-08-13T00:36:10.907004Z","iopub.execute\_input":"2022-08-13T00:36:10.907407Z","iopub.status.idle":"2022-08-13T00:36:11.350643Z","shell.execute\_reply.started":"2022-08-13T00:36:10.907372Z","shell.execute\_reply":"2022-08-13T00:36:11.348832Z"}}

# We count the data by output class DX

sns.catplot(x="dx",

kind="count",

data=ground\_truth)

# %% [markdown]

# ### Image collection channel

# In this section, we exploring the data collection type to identified the source of the image is from which channel.

# As per graph below, we can see the majority of image collection is from histo. Image collection from other channels are not that much.

# %% [code] {"jupyter":{"outputs\_hidden":false},"execution":{"iopub.status.busy":"2022-08-13T00:38:08.464950Z","iopub.execute\_input":"2022-08-13T00:38:08.465436Z","iopub.status.idle":"2022-08-13T00:38:09.353124Z","shell.execute\_reply.started":"2022-08-13T00:38:08.465401Z","shell.execute\_reply":"2022-08-13T00:38:09.351468Z"}}

# We count the data by output class DX and DX\_TYPE

sns.catplot(x="dx",

col="dx\_type",

kind="count",

data=ham)

# %% [markdown]

# ### Gender and Cancer

#

# Distribution of gender is quite nice, the proportion of distribution across male and female are not much difference.

# %% [code] {"jupyter":{"outputs\_hidden":false},"execution":{"iopub.status.busy":"2022-08-13T00:38:59.801426Z","iopub.execute\_input":"2022-08-13T00:38:59.801943Z","iopub.status.idle":"2022-08-13T00:39:00.574522Z","shell.execute\_reply.started":"2022-08-13T00:38:59.801906Z","shell.execute\_reply":"2022-08-13T00:39:00.573021Z"}}

# We count the data by output class DX and Gender

sns.catplot(x="dx",

col="sex",

kind="count",

data=ground\_truth)

# %% [code] {"jupyter":{"outputs\_hidden":false},"execution":{"iopub.status.busy":"2022-08-13T00:39:19.384259Z","iopub.execute\_input":"2022-08-13T00:39:19.385557Z","iopub.status.idle":"2022-08-13T00:39:20.258343Z","shell.execute\_reply.started":"2022-08-13T00:39:19.385494Z","shell.execute\_reply":"2022-08-13T00:39:20.256760Z"}}

#Age distribution on difference dx group

ground\_truth.age.hist(by=ground\_truth.dx)

# %% [code] {"jupyter":{"outputs\_hidden":false},"execution":{"iopub.status.busy":"2022-08-13T00:39:21.703139Z","iopub.execute\_input":"2022-08-13T00:39:21.704264Z","iopub.status.idle":"2022-08-13T00:39:22.988231Z","shell.execute\_reply.started":"2022-08-13T00:39:21.704213Z","shell.execute\_reply":"2022-08-13T00:39:22.987128Z"}}

# We count the data by output class DX and Age

sns.catplot(x="dx",

hue="age",

kind="count",

data=ham)

# %% [code] {"jupyter":{"outputs\_hidden":false},"execution":{"iopub.status.busy":"2022-08-13T00:40:06.329679Z","iopub.execute\_input":"2022-08-13T00:40:06.330123Z","iopub.status.idle":"2022-08-13T00:40:07.307243Z","shell.execute\_reply.started":"2022-08-13T00:40:06.330087Z","shell.execute\_reply":"2022-08-13T00:40:07.306200Z"}}

# We count the data by output class DX and Age

sns.catplot(x="dx",

hue="localization",

kind="count",

data=ham)

# %% [markdown]

# # Protential Problem when clustering

# There is a potential problems on clustering with this dataset, since clustering is measure by distance between samples within a certain maximum.

# In this dataset, 'nv' is a large and unequally scatteer class when applying k-means, other not-so-prominent clusters may be not be found. In other word, clustering is quite difficult on this dataset.

**APPENDIX 2 - R Code**

---

title: "DANA4840\_Final\_Exam"

author: "Billy Choy"

date: '2022-08-13'

output:

pdf\_document: default

html\_document: default

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

```

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the \*\*Knit\*\* button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}

gt = read.csv("GroundTruth.csv")

```

## Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}

# Remove unuse column

gt$X <- NULL

gt$lesion\_id <- NULL

gt$sex <- NULL

gt$image\_id <- NULL

gt$path <- NULL

gt$dx\_type <- NULL

gt$localization <- NULL

gt$image\_pixel <- NULL

#Label replace dx column

gt$dx <- NULL

gt$age <-as.numeric(gt$age)

gt$label <-as.factor(gt$label)

```

# Normalisation on pixel

For column with c1 to c146 are pixel value in RGB, therefore we have to divide them by 256 bits for normalization.

```{r}

gt[2:148] = gt[2:148]/255

```

# Normalisation on age

Next we normalise the age column.

```{r}

gt[1] = scale(gt[1], center = TRUE)

```

Let's have a quick check on the data

```{r}

View(gt)

```

```{r}

colSums(is.na(gt))

```

# Kmeans

```{r}

# Within cluster sum of squares by cluster:

# [1] 1856.834 1454.450 2293.645 1743.224 1634.729 1876.404 1487.317

# (between\_SS / total\_SS = 50.9 %)

library(philentropy)

library(stats)

library(factoextra)

KM <- distance(gt[,c(2:148)], method= "dice")

KM <- kmeans(gt[,c(2:148)], centers = 7)

KM

```

# Performance

```{r}

dfCombine <- cbind(gt, Cluster = KM$cluster-1)

#Table Cluster and Symboling

table(dfCombine$Cluster == gt$label)

table(dfCombine$Cluster, gt$label)

```

```{r}

#Visualization in Clustering

library(factoextra)

fviz\_cluster(KM, data = gt[,c(2:148)],

# palette = c("#2E9FDF", "#00AFBB", "#E7B800", "#9B5DE5"),

geom = "point",

ellipse.type = "convex",

ggtheme = theme\_bw()

)

```

```{r}

par(mfrow=c(2,2))

for (i in 2:148) {

boxplot(gt[,i]~KM$cluster, xlab="Cluster",

ylab=names(gt)[i], varwidth=T)

}

```

#Elbow Graph on optimal k

Looking at the elbom graph seems it make more sense to run a clustering with k=3. We try to cluster the universities in high, medium and low.

```{r}

library(factoextra)

fviz\_nbclust(gt[,c(2:148)], kmeans, method = "wss",k.max=8)

```

```{r}

#Silhouette method

fviz\_nbclust(gt[,c(2:148)], FUN = kmeans, method = "silhouette")

```

# Subset data

```{r}

library(tidyverse)

df1 = gt %>% filter (label == '0')

df2 = gt %>% filter (label == '1')

df3 = gt %>% filter (label == '2')

df4 = gt %>% filter (label == '3')

df5 = gt %>% filter (label == '4')

df6 = gt %>% filter (label == '5')

df7 = gt %>% filter (label == '6')

set.seed(451)

# Extract 5 random rows without replacement

df1\_100 <- df1 %>% sample\_n(100, replace = FALSE)

df2\_100 <- df2 %>% sample\_n(100, replace = FALSE)

df3\_100 <- df3 %>% sample\_n(100, replace = FALSE)

df4\_100 <- df4 %>% sample\_n(100, replace = FALSE)

df5\_100 <- df5 %>% sample\_n(100, replace = FALSE)

df6\_100 <- df6 %>% sample\_n(100, replace = FALSE)

df7\_100 <- df7 %>% sample\_n(100, replace = FALSE)

#Combine gt\_subset\_100

gt\_subset\_100 <- rbind(df1\_100, df2\_100, df3\_100, df4\_100, df5\_100, df6\_100, df7\_100)

#Kmean on gt\_subset\_100

KM\_100 <- distance(gt\_subset\_100[,c(2:148)], method= "dice")

KM\_100 <- kmeans(gt\_subset\_100[,c(2:148)], centers = 7)

KM\_100

```

# Performance

```{r}

dfCombine2 <- cbind(gt\_subset\_100, Cluster = KM\_100$cluster-1)

#Table Cluster and Symboling

table(dfCombine2$Cluster == gt\_subset\_100$label)

table(dfCombine2$Cluster, gt\_subset\_100$label)

par(mfrow=c(2,2))

for (i in 2:148) {

boxplot(gt\_subset\_100[,i]~KM\_100$cluster, xlab="Cluster",

ylab=names(gt\_subset\_100)[i], varwidth=T)

}

```

# For agglomerative clustering technique, identify the best approach for the data ("average", "single", "complete", "ward")

```{r}

library(factoextra)

library(cluster)

library(purrr)

d = dist(gt\_subset\_100, method = "euclidean")

# methods to assess

m <- c( "average", "single", "complete", "ward")

names(m) <- c( "average", "single", "complete", "ward")

# function to compute coefficient

ac <- function(x) {

agnes(gt\_subset\_100, method = x)$ac

}

map\_dbl(m, ac)

```

Run agglomerative clustering technique with "ward"

```{r}

#Hierarchical clustering using Ward linkage

hc1 = hclust(d, method = "ward.D2")

hc1

#cutting the tree

plot(hc1, cex = 0.6)

rect.hclust(hc1, k = 7, border = 2:5)

##based on experience, slicing horizontally to produce unique clusters either by specifying a similarity or the number of clusters desired

democut<-cutree(hc1,k=7)

democut

```

```{r}

# compute divisive hierarchical clustering

hc4 <- diana(gt\_subset\_100[,c(2:148)])

# Divise coefficient is the amount of clustering structure found

hc4$dc

## [1] 0.8806264

```

```{r}

# compute divisive hierarchical clustering

hc7 <- diana(gt[,c(2:148)])

# Divise coefficient is the amount of clustering structure found

hc7$dc

## [1] 0.8806264

```

THANKS FOR YOUR PARTICIPATION! THE END