DANA 4840
Spring 2021
Final Project Report

Breast Cancer Proteomes:
Dividing Breast Cancer Patients into Separate Sub-classes

Group 4
David Shih 100334127
Zhixuan Zhang (Eric) 100338057
Chun Ching Look (Cyrus) 100347726
Simranjit Singh (Simran) 100348495
Kailash Sukumaran 100350193

Introduction

Breast Cancer accounts for a quarter of all cancers in females (Baskin 2010). The prevalence stems from the epithelial cells of the mammary ductal systems that produce a secretion called nipple aspirate fluid or NAF (Brunoro 2019). NAF is a protein rich breast proximal fluid related to the tumor microenvironment in cancer patients (Brunoro 2019). Thus NAF is a valuable biological sample to study secreted proteins without contamination by other interstitial fluids or cells. Although proteomic studies of human body fluids and tissues are challenging by nature due to the variability among each individual, proteomic studies for breast cancer have strong statistical power because the breast is a pair organ. Thus we can use the contralateral non-diseased breast from the same individual as the control in unilateral breast cancer studies (Brunoro 2019).

Lots of awareness has been made about breast cancer that led to the success of early screening programs and new therapeutic strategies but due to metastatic relapses lives are still lost (Johannson 2019). Breast cancers can be sensitive to the hormones that the body produces as it has receptors that can form a relationship with the hormones. The key hormone receptors that play a role are estrogen receptors (ER), and progesterone receptors (PR). These hormones may increase the severity of breast cancer (Mayo Clinic 2020). Another important indicator of how aggressive breast cancer can be is the Human Epidermal Growth Factor Receptor 2 (HER2) gene. The HER2 gene when present in breast cancer cells produces an excess amount of HER2 protein which can increase the growth rate of breast cancer (Mayo Clinic 2020). Using a combination of these markers allows doctors to categorize breast cancers and provide the best treatment.

Research Question

Using the genetic makeup of breast cancer as well as whether hormone receptors are present, we want to use classification algorithms to classify patients into four different breast cancer categories and use clustering algorithms to see if we can group breast cancer patients based on breast cancer categories treatment recommendations by doctors.

Dataset Description

The first dataset contains published iTRAQ proteome profiling of 77 breast cancer samples generated by the Clinical Proteomic Tumor Analysis Consortium (NCI/NIH). It contains expression values of samples for approximately 12,000 proteins, with missing values present when a given protein could not be quantified. The second dataset includes patient demographic information as well as also containing the column ‘PAM50 mRNA’ which labels our patients into the four different breast cancer categories we will be using for our report. The four labels are Luminal A, Luminal B, HER2 Positive, and Basal-Like. Each of the aforementioned categories have their own recommended treatment as prescribed by doctors such as chemotherapy, hormone therapy, HER2 targeted therapy etc. The last dataset contains a list of 50 genes used for PAM50 (biological test). They are used to identify the intrinsic subtypes of breast cancer as these 50 genes are the biological makeup of the underlying breast cancer.

Variable Description

(1) 77_cancer_proteomes_CPTAC_itraq.csv; this file contains: RefSeq_accession_number, which is RefSeq protein ID (each protein has a unique ID in a RefSeq database); gene_symbol which indicate a symbol unique to each gene because every protein is encoded by some gene; gene name, which is a full name of that gene. The rest of the columns, with the exception of the last three, represents protein data of cancer patients. The last three columns are from healthy individuals. The positive values of the expression data denote a simulation action in the person and the negative values of the expression data denote an inhibition action.

(2) clinical_data_breast_cancer.csv; included is clinical data on two male and 103 female subjects. This dataset has 30 features which break down as follows: the first column, "Complete TCGA ID", is used to match the sample IDs in the main cancer proteomes file. All other columns have self explanatory names, containing data about the cancer classification of a given sample using different methods. The main features referenced in this dataset are Age at Initial Pathologic Diagnosis, ER Status, PR Status, HER2 Final Status, Tumor, Node, Metastasis, AJCC Stage, Converted Stage, Survival Data Form, OS event, and PAM50 mRNA.

(3) PAM50_proteins.csv; this is a list of genes and proteins used by the PAM50 classification system. The column RefSeqProteinID contains the protein IDs that can be matched with the IDs in the main protein expression data set (77_cancer_proteomes_CPTAC_itraq.csv).

Describe the features of clean dataset

Our main dataset, 77_cancer_proteomes_CPTAC_itraq.csv, contained 12553 rows and 86 variables. Each row represented a unique ID representing a protein in the RefSeq database. The important features of the columns were protein data of cancer patients (i.e. patient is the variable on the column, protein data on the rows). Missing values in this dataset was only found in the protein data of cancer patients. This problem was tackled by replacing the missing values with the medium. No other missing values were found in this dataset and the following datasets.

Our secondary dataset, clinical_data_breast_cancer.csv, contained patient information such as age, estrogen receptor (ER) status, progesterone receptor (PR) status, human epidermal growth factor receptor 2 (HER2) status, and tumor stage just to name a few of the important features.

In our secondary dataset, the rows contained patient IDs while the column contained their information. This was different from our first dataset. In order for us to join the two datasets together, we transposed the first dataset so that the patient IDs were on the rows instead of columns, and the protein data became the column variables. The two datasets were then joined based on

patient IDs. The end result is a dataset consisting of 80 rows (patient IDs) and 11542 columns (demographic info + protein data).

Our last dataset, PAM50_proteins.csv, is a list of genes and proteins used by the PAM50 classification system. This system is now known as the Prosigna Breast Cancer Prognostic Gene Signature Assay and is a genomic test that analyzes the activity of certain genes in early-stage, hormone-receptor-positive breast cancer.

At this point we realized we may have to condense our dataset further as we ran into a lot of computational issues when trying to run dimensional reduction algorithms and clustering algorithms in the dataset with 80 rows and 11542 columns. We made a decision to tackle this problem by filtering out our 11542 columns of protein data with the matching proteins found in the PAM50 dataset. Genes used for PAM50 (biological test) are used to identify the intrinsic subtypes of breast cancer (LA, LB, HER2 and Basal). It is likely to be relevant to be telling relevant information about breast cancer in many different treatment settings as it measures the underlying gene makeup of breast cancer. We also wanted to focus on breast cancer patients who are female only so we removed 2 rows of male patients. The end result is a scaled data frame consisting of 78 rows representing 78 female patients and 39 columns representing the protein data found in both PAM50 dataset and the 77_cancer_proteomes dataset.

Methods

Building different logistic models

For validation purposes, we built 3 different logistic models for 3 variables: ER, PR, and HER2. Before we ran those models, we applied PCA techniques on the scaled data frame. We then selected the first 9 components which all have standard deviation larger than 1. Those components were combined into a new data frame. Next, we added corresponding response variables (ER, PR, and HER2) for each model. The variables have binary values (1 and 0) only. After that, we split and partition the new data frame into training set and test set. The training set has 90% of data while the test set has 10% of data. Using the training set, we built different logistic models. Once the models were ready, we applied the training set and test set to our models and made predictions. We also measured the accuracy of the training set and test set.

SVM

For predicting breast cancer categories Luminal A, Luminal B, HER2 positive, Basal-like, we are using SVM (support vector Machines). It is a non-linear classifier and is robust to outliers as well.

Our main goal is to build the most generalized model which should perform well on test data as well, we tuned hyperparameters efficiently to avoid overfitting or underfitting problems.

K-means Clustering

K-means algorithm starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids.

As our main goal is to cluster the four breast cancer categories mentioned above (Luminal A, Luminal B, etc) in order to recommend treatment. We applied K-means clustering on the scaled data frame. We used 3 methods, including elbow method, silhouette method, NbClust() method, to find the optimal K.

K-medoids Clustering

K-medoids algorithm (also known as Partitioning Around Medoids (PAM)) selects the most centrally located object in a cluster, instead of the mean value of the object in a cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids.

As previously mentioned our main goal is to cluster the four breast cancer categories and recommend treatment, we applied K-medoids clustering on the scaled data frame. We used 3 methods, including elbow method, silhouette method, NbClust() method, to find the optimal K.

Hierarchical Clustering

Hierarchical clustering is an algorithm used to group similar objects into groups called clusters. These clusters are distinct from one another and the observations within each cluster contain various similarities.

We will be using 4 clustering methods (single linkage, average linkage, complete linkage, and ward linkage) to see if we can group the four breast cancer categories.

Results

Building Different Logistic Models

ER (Estrogen Receptor)

Logistic regression model is trained using first 9 principle components with ER as a response variable. In order to ensure the accuracy of the model, the model is trained and tested repeatedly for 5 times with random samples each time. The average test and train accuracy rate is around 94% which means our model is well generalized for unseen data.

	Predicted	
Actual	ER Negative	ER Positive
ER Negative	3	2
ER Positive	1	9

Train set overall accuracy for 5 iterations :

0.9193548 0.9354839 0.9193548 0.9193548 1.0000000

Test set overall accuracy for 5 iterations:

1.0000 1.0000 0.9375 1.0000 0.8000

PR (Progesterone Receptor)

Logistic regression model is trained using first 9 principle components with PR as a response variable. In order to ensure the accuracy of the model, the model is trained and tested repeatedly 5 times with random samples as well. The average test and train accuracy is around 77% and 87% respectively.

	Predicted	
Actual	PR Negative	PR Positive
PR Negative	3	0
PR Positive	1	4

Train set overall accuracy for 5 iterations:

0.9154930 0.8714286 0.8857143 0.8857143 0.8714286

Test set overall accuracy for 5 iterations:

0.7142857 1.0000000 0.7500000 0.7500000 0.8750000

HER2

Logistic regression model is trained using first 9 principle components with HER2 as a response variable. In order to ensure the accuracy of the model, the model is trained and tested repeatedly 5 times with random samples as well.

	Predicted	
Actual	HER2 Negative	HER2 Positive
HER2 Negative	12	0
HER2 Positive	0	3

Train set overall accuracy for 5 iterations:

```
0.9193548 0.8870968 0.8870968 0.8870968 0.8888889
```

Test set overall accuracy for 5 iterations:

```
0.8750 1.0000 1.0000 0.9375 1.0000
```

SVM

If we train a Linear SVM model with all the predictors, we have a problem of overfitting. We can clearly see the below confusion matrices where train set accuracy is 100% while test accuracy is only 56 %

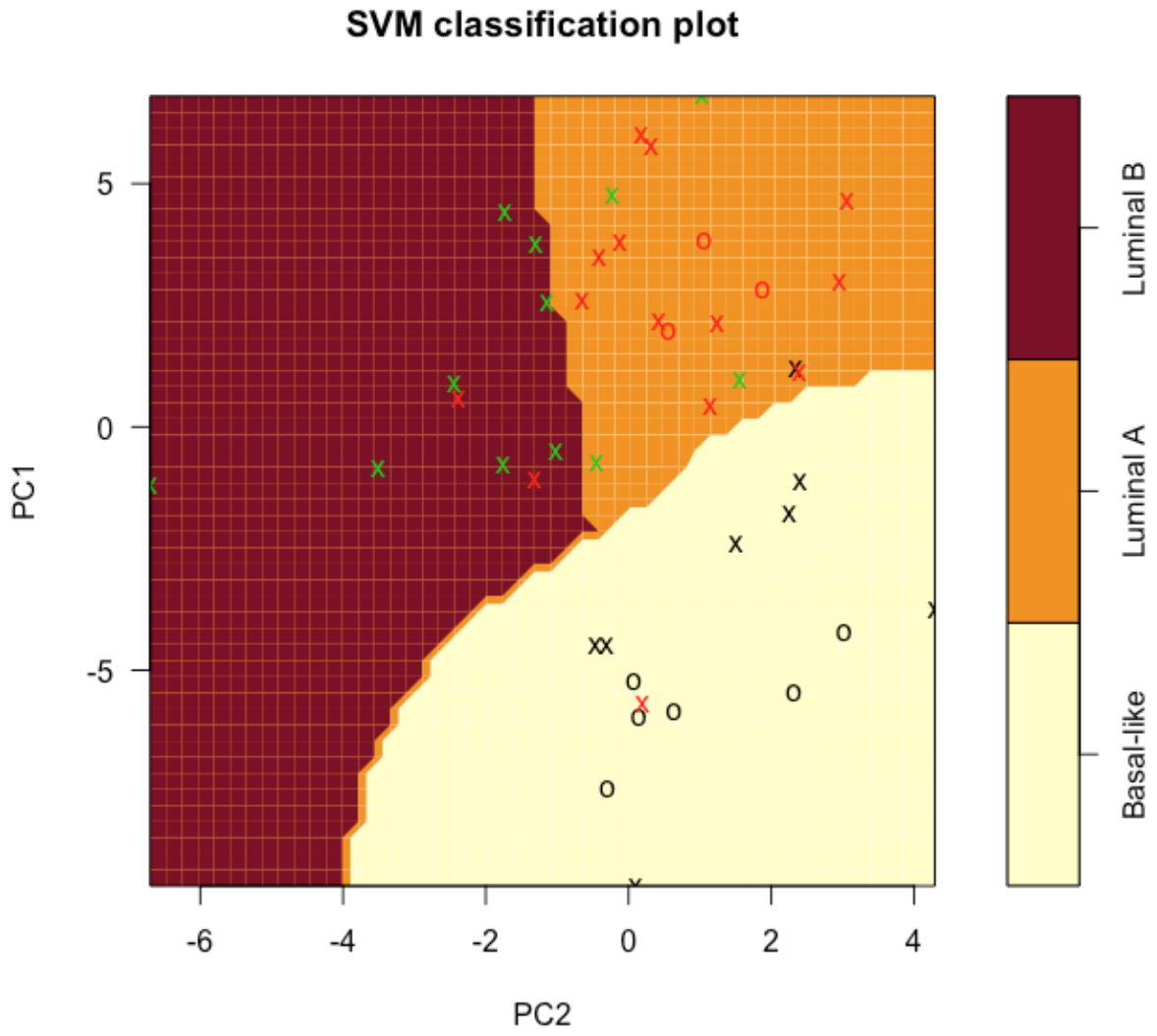
Confusion matrix for train set

	Actual			
Predicted	Basal-like	HER2-enriched	Luminal A	Luminal B
Basal-like	16	0	0	0
HER2-enriched	0	9	0	0
Luminal A	0	0	20	0
Luminal B	0	0	0	17

Confusion matrix for test set

	Actual			
Predicted	Basal-like	HER2-enriched	Luminal A	Luminal B
Basal-like	3	1	0	0
HER2-enriched	0	1	0	2
Luminal A	0	0	2	1
Luminal B	0	2	1	3

In order to avoid the problem of overfitting, we used principal component analysis. Moreover, there are not enough data points which belong to the HER2-enriched category. We decided to remove the rows which belong to HER2. Additionally, we can see the plot below i.e. if we consider only the first two principal components and use SVM, we can get good accuracy for the both test and train set.



Nonlinear SVM with radial based kernel is used for predicting breast cancer categories because it performed better as compared to other Linear Classifiers like logistic regression and linear SVM.

Using SVM for predicting breast cancer categories, the accuracy of the test set is around 90 %.

Train set Confusion Matrix

Predicted	Actual Basal-like	Luminal A	Luminal B
Basal-like	13	1	0
Luminal A	1	14	4
Luminal B	0	2	8

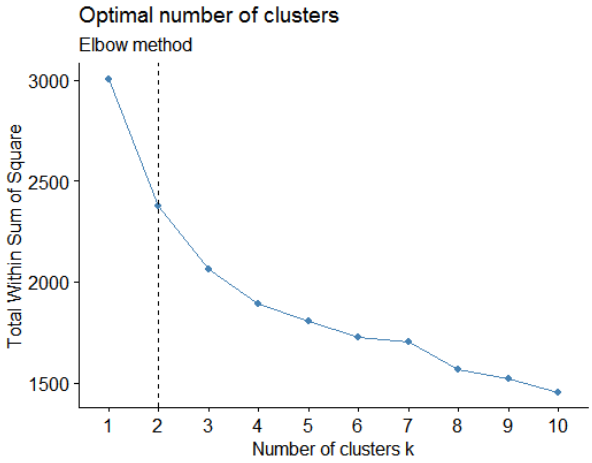
Train set Confusion Matrix

	Actual		
Predicted	Basal-like	Luminal A	Luminal B
Basal-like	5	0	0
Luminal A	0	6	1
Luminal B	0	0	10

K-means Clustering

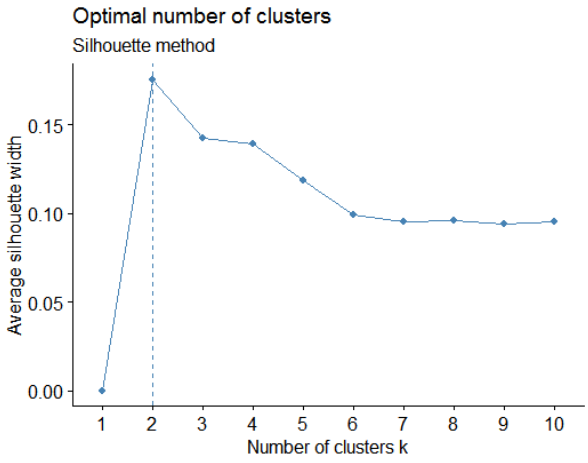
Elbow method

In cluster analysis, the elbow method is used in determining the number of clusters in a data set. The method consists of plotting the explained variation as a function of the number of clusters, and picking the elbow of the curve as the number of clusters to use. From the plot, the joint is at x=2 and hence K=2.



Silhouette method

Silhouette refers to a method of interpretation and validation of consistency within clusters of data. The technique provides a succinct graphical representation of how well each object has been classified. From the plot, the largest average silhouette width is at x=2 and hence K=2.



NbClust() method

From the conclusion, it is suggested that K=3.

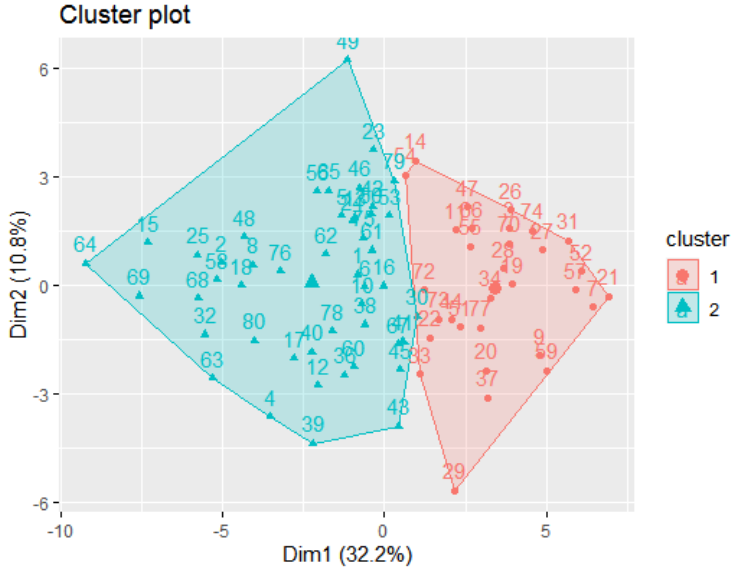
```
Among all indices:
5 proposed 2 as the best number of clusters
15 proposed 3 as the best number of clusters
3 proposed 4 as the best number of clusters

                 ***** Conclusion *****

According to the majority rule, the best number of clusters is  3
```

Quality of K-means Clustering

The ground truth suggests that we should adopt K=2. From the plot, it shows that clusters are well separated when K=2. Still, when calculating the percentage of the TSS "explained", it is only 0.2087298. As it is not close to 1, we can conclude the quality of K-means clustering is not good.




```
> model <- kmeans(treatement[-length(treatement)], centers = 2, iter.max = 500)
> model$betweenss/model$totss
[1] 0.2087298
```

ER

Clusters	Negative	Positive
1	24	19
2	2	33

PR

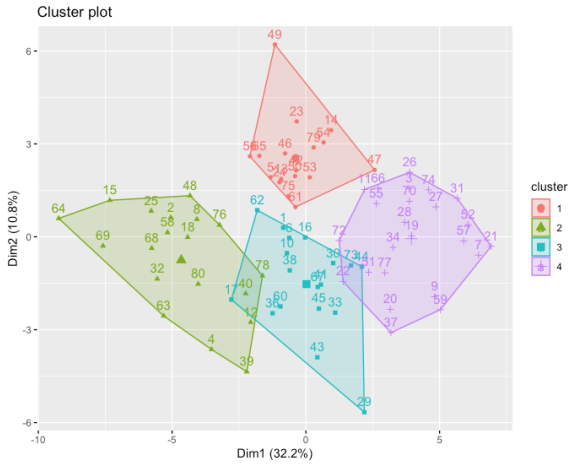
Clusters	Negative	Positive
1	28	15
2	8	27

HER2

Clusters	Negative	Positive
1	29	14
2	30	5

We can clearly see from the above tables that two clusters are not able to categorize breast cancer patients based on their hormone status.

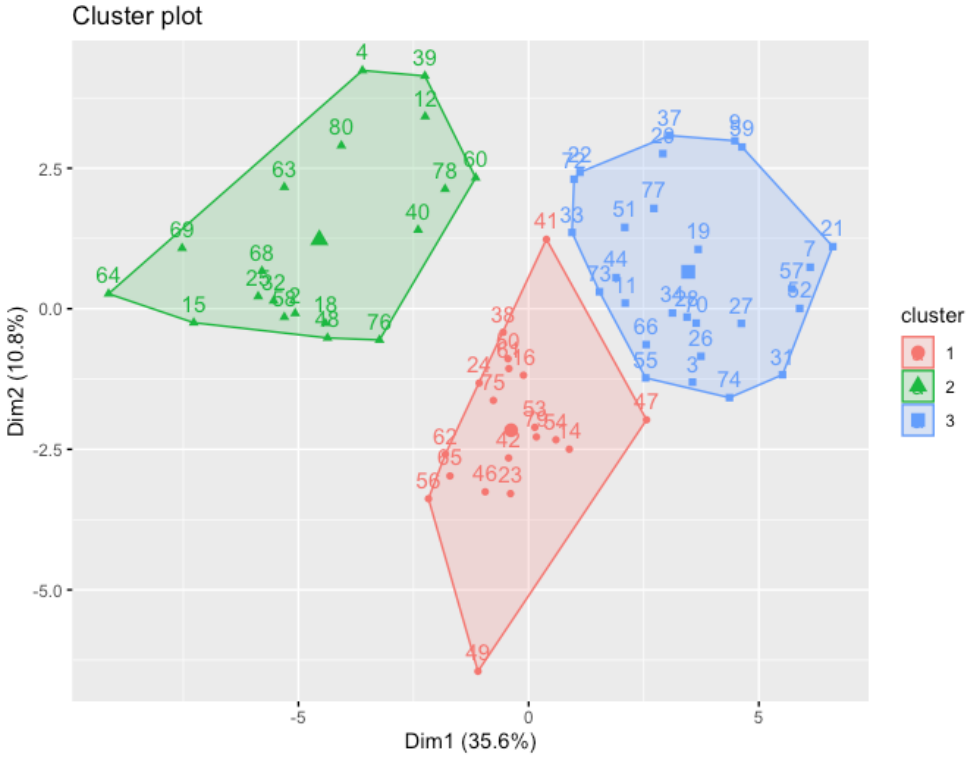
For k = 4



	Basal-like	HER2-enriched	Luminal A	Luminal B
1	0	2	2	13
2	17	1	1	0
3	1	10	2	5
4	1	0	18	5

Cluster 1 - Most of the patients in cluster 1 belong to the category of Luminal B.
Cluster 2 - Most of the patients in this cluster belong to the category basal-like.
Cluster 3 - This cluster comprises patients from HER2.
Cluster 4 - This cluster comprises patients belonging to Luminal A

Without HER2 For K = 3



	Basal-like	Luminal A	Luminal B
1	0	3	16
2	18	1	0
3	1	19	7

Cluster 1 - Most of the patients in the cluster 1 belong to the category of Luminal B. These patients may benefit from a combination of chemotherapy and hormone therapy.

Cluster 2 - Most of the patients in this cluster belong to the category basal-like. Basal-like breast cancers are likely to benefit from chemotherapy.

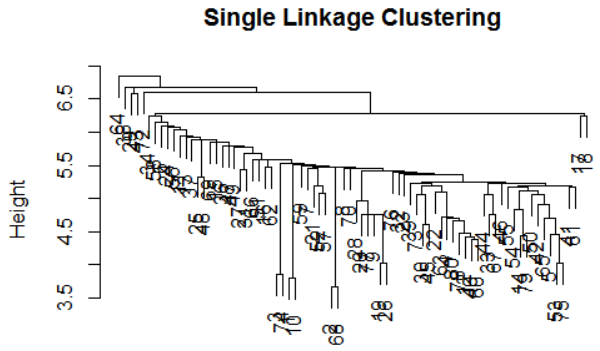
Cluster 3 - Most of the patients in cluster 3 belong to the category of Luminal A. Luminal A breast cancers are more likely to benefit from hormone therapy and may benefit from chemotherapy.

K-medoids Clustering

Since the results are similar to that of K-means clustering, we exclude the outputs here.

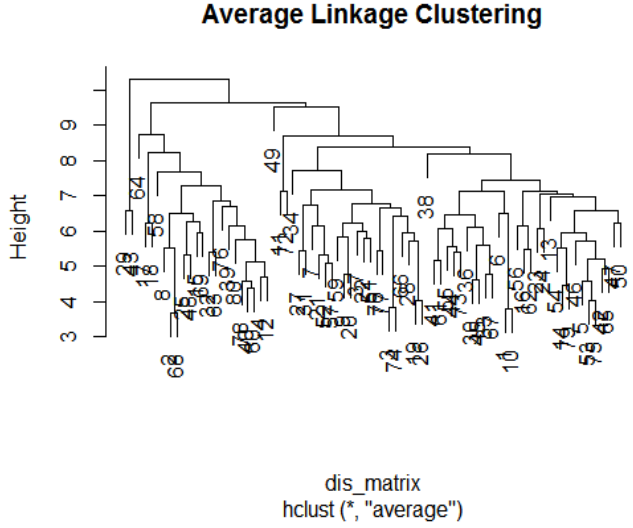
Hierarchical Clustering

Single Linkage Clustering: Since the chaining effect appears, we would reject this approach.



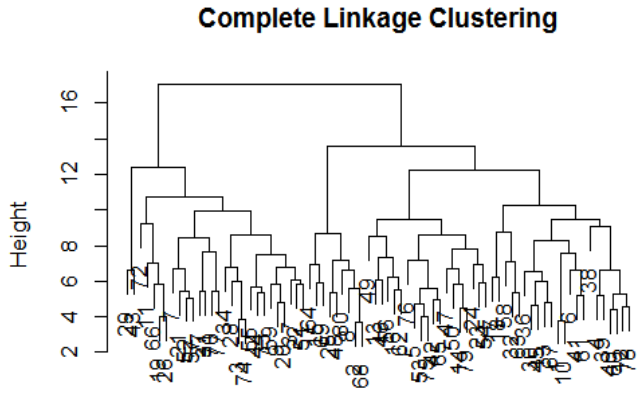
Clusters	Basal-like	HER2-enriched	Luminal A	Luminal B
1	18	11	23	22
2	0	2	0	0
3	0	0	0	1
4	1	0	0	0

Average Linkage Clustering: Since the chaining effect appears, we would reject this approach.



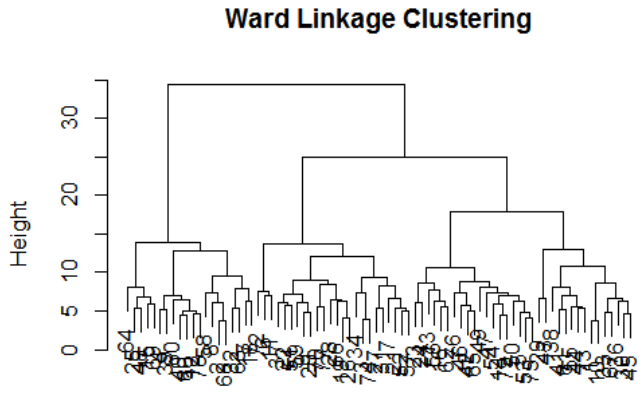
Clusters	Basal-like	HER2-enriched	Luminal A	Luminal B
1	1	9	22	22
2	18	2	1	0
3	0	2	0	0
4	0	0	0	1

Complete Linkage Clustering: There are 4 clusters but clusters are not well separated.



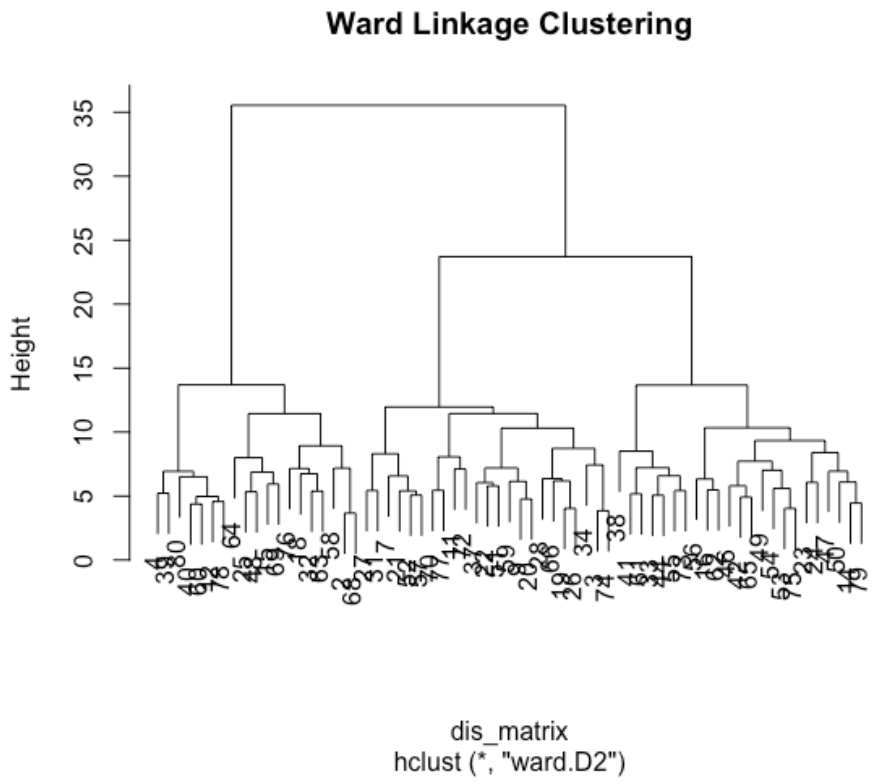
Clusters	Basal-like	HER2-enriched	Luminal A	Luminal B
1	10	10	4	17
2	8	1	0	0
3	1	0	19	6
4	0	2	0	0

Ward Linkage Clustering: There are 4 clusters and clusters are well separated.



Clusters	Basal-like	HER2-enriched	Luminal A	Luminal B
1	0	8	2	5
2	17	2	1	0
3	1	1	18	4
4	1	2	2	14

Ward Linkage Clustering (Without HER2)



Clusters	Basal-like	Luminal A	Luminal B
1	18	1	0
2	1	18	4
3	0	4	19

We can see from the above table that ward linkage without the data points belong to HER2, able to form the clear cluster.
Cluster 1 consisted of patients with the cancer of type basal-like
Cluster 2 consisted of patients with the cancer of type Luminal A
Cluster 3 consisted of patients with the cancer of type Luminal B

Ward Linkage Hierarchical clustering performed better as compared to K-mean clustering in categorizing the breast cancer patients after removing the rows belonging to HER2.

Discussion

After processing the data, we apply supervised as well as unsupervised machine learning techniques to acquire insights from the protein data. The predictors in this case will be the

proteomes we collected after merging with the PAM50 dataset. According to our findings, the proteome's information can predict the hormone status (ER, PR, HER2) which are introduced in clinical_data_breast_cancer data of breast cancer patients using logistic regression. After getting relatively good accuracy predicting ER, PR and HER2 hormones characteristics of the patients, the group looked deeper into the categories of breast cancer. Medical researchers are increasingly using genetic information about breast cancer cells to categorize breast cancers (Mayo Clinic 2020). The breast cancer groups includes:

	Group 1: Luminal A (1 of PR or ER is positive)	Group 2: Luminal B (1 of PR or ER is positive)	Group 3: HER2 Positive	Group 4: Basal Like
ER Status	Positive or Negative	Positive or Negative	Negative	Negative
PR Status	Positive or Negative	Positive or Negative	Negative	Negative
HER2 Status	Negative	Positive or Negative	Positive	Negative
KI 67 Protein levels	Low	High	/	/
Treatment	Likely to benefit from Hormone therapy. May benefit from chemotherapy	Likely to benefit from Chemotherapy. May benefit from Hormone therapy + HER2 treatment	Likely to benefit from HER2 treatment & Chemotherapy	Likely to benefit from Chemotherapy

With the information above, our group labels the patients into four categories: Luminal A, Luminal B, HER2 positive, Basal-like. These four categories will provide us with the benchmark to validate our results.

Furthermore, we use SVM (Support Vector Machines) to predict the three breast cancer categories (we remove the HER2 Positive patients' sample because it is too small), because SVM is a nonlinear classifier and robust to outliers as well, it performed better in classification tasks and gave a better accuracy rate. The overall accuracy rate in classifying the three breast cancers groups utilized by SVM is about 90%, it again indicates the dependence between our predictors (proteomes) and the categories of breast cancer.

After using supervised machine learning and understanding the dependence between proteomes and the four categories of breast cancer, we moved on to unsupervised machine learning to further validate our research question about how many clusters of patients we can obtain based on the proteomes information.

We first introduce K-means and K-medoids methods to perform clustering of our data. K-means and K-medoids methods provided us very similar outputs, the only difference was the center of each cluster. In this case, we will mainly discuss the result from K-means. Having known the dependence between proteomes and hormone types, we first use K-means to validate our results from Logistic regression.

In order to confirm how many clusters we should choose, we used Elbow methods, Silhouette and NbClust function to help decide k's value, despite we already know there should be four clusters in our patients' data. Instead of suggesting k=4, these three methods suggested different k values: k=2 and k=3. After comparing all outputs for k=2, k=3, and k=4, we found that k-mean clustering can precisely separate the four types of breast cancer: Luminal A, Luminal B, HER2 Positive and Basal-like patients.

Moreover, we perform hierarchical clustering methods to help us validate the clustering we get from the K-means approach. In this case, only the Ward Linkage Method can provide significant output, but it again proves that it makes more sense to have four clusters, and it gives us a good clustering for four groups, but the accuracy is slightly off compared to K-mean clustering.

Additionally, In order to compare Supervised and Unsupervised methods, we ran Ward Linkage and K-means clustering methods without the data points belonging to the HER2 category and chose 3 clusters. Below tables compare the k-mean, ward linkage and SVM results. For SVM, we used only the first two Principal components otherwise it would over fit our data and give 100 % accuracy.

SVM

	Actual		
Predicted	Basal-like	Luminal A	Luminal B
Basal-like	18	1	0
Luminal A	1	20	5
Luminal B	0	2	18

Ward Linkage clustering

Clusters	Basal-like	Luminal A	Luminal B
1	18	1	0
2	1	18	4
3	0	4	19

K-mean clustering

	Basal-like	Luminal A	Luminal B
1	0	3	16
2	18	1	0
3	1	19	7

We can see from the above results that unsupervised methods performed as good as supervised methods in categorizing the breast cancer patients.

When we do the dimension reduction, we realize we can't run dimensional reduction algorithms like PCA and clustering algorithms in the dataset with 80 rows and 11542 columns. So, we have to figure other ways to reduce the dimension of the data, so we were looking for references in the PAM_50 dataset to help us filter out the majority of the columns. We have only 78 observations after cleaning and merging our data. It is a very small dataset, but it still is able to give us a relatively good accuracy for supervised learning and good clustering results. The combination of using supervised learning first before unsupervised learning allowed us to verify the importance and significance of PR, ER, HER2 status. We then used this information in our unsupervised learning so that we can label our clusters correctly and recommend treatments as suggested by doctors.

In our case, it was really helpful for us to try multiple supervised machine learning methods to find out more about the data. At first, we noticed that there are many variables in the patients' demographic information columns, some of them are numerical variables such as OS time, and more of them are categorical variables such as tumor stages, nodes, and the three hormones used to categorize breast cancer types. Then we applied logistic regression, KNN method, SVM, neural

networks just to see the results. Luckily, we got good results from logistic regression on these three variables. Furthermore, successfully predicting these three variables proved that there is dependence between proteins and the hormones and gene status. From here we did more research on these three hormones and we were able to find that we can actually categorize breast cancer into four clusters. It gives us the ground truth and a benchmark to validate our analysis and clustering results. It was an important breakthrough for our project.

Our group faced several constraints in this project. First of all, none of the group members had a biology or medical background. The limits on knowledge about genes and proteomes cost us more time to interpret the dataset and do scientific research about breast cancers. We have 78 observations (including HER2 positive patients) which is not enough for clustering, furthermore, our data is not balanced for these four types of breast cancer. We have more luminal A & B patients than HER2 positive patients which only has 13 patients, it is not enough for the model training nor clustering. Time is another constraint, cleaning and understanding the dataset took us a lot of time, plus the time we spent on the research of breast cancers. We don't have enough time to fully validate our results.

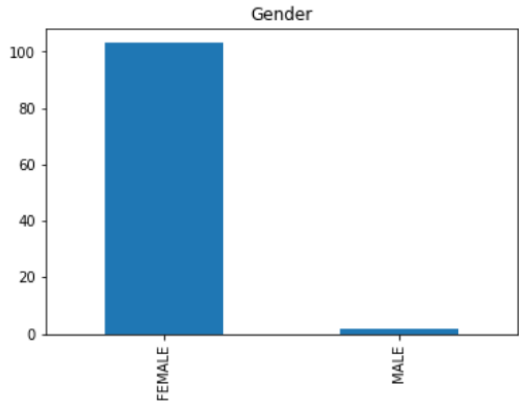
Future study on this project will mainly focus on finding more similar data to cross-validate our findings utilizing supervised/unsupervised machine learning techniques. We believe that with more observations, we would be able to have more confidence with our models and clustering results. Moreover, these four types of breast cancers are characterized by hormone status, and each type of breast cancer is recommended with certain types of treatment. We can use our current findings to further predict the type of cancers and give doctors medical suggestions about individualized treatment.

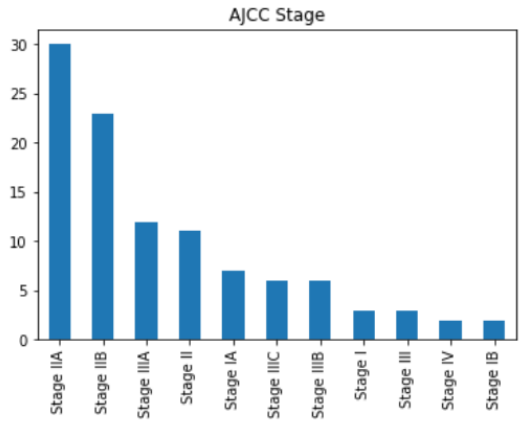
References and Acknowledgements

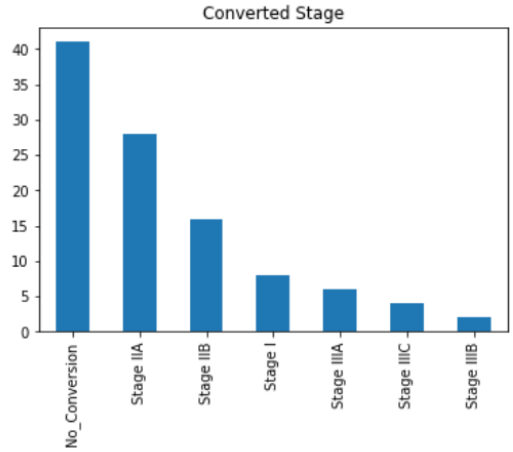
- Johansson, H., Socciarelli, F. and Vacanti, N., 2019. Breast cancer quantitative proteome and proteogenomic landscape. *Nature Communications*, 10(1). <https://www.nature.com/articles/s41467-019-09018-y>
- Baskin, Y. and Yigitbasi, T., 2010. Clinical Proteomics of Breast Cancer. *Current Genomics*, 11(7), pp.528-536. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3048315/>
- Brunoro, G., Carvalho, P. and Barbosa, V., 2019. Differential proteomic comparison of breast cancer secretome using a quantitative paired analysis workflow. *BMC Cancer*, 19(1). <https://bmccancer.biomedcentral.com/articles/10.1186/s12885-019-5547-y>
- Breast - Datasets - PLCO - The Cancer Data Access System. (2021). Retrieved 5 April 2021, from https://cdas.cancer.gov/datasets/plco/19/
- Anderson, C., & Bartee, L. (2021). How do genes direct the production of proteins?. Retrieved 6 April 2021, from https://openoregon.pressbooks.pub/mhccbiology102/chapter/how-do-genes-direct-the-production-of-proteins/
- What your breast cancer type means. (2020, February 05). Retrieved from https://www.mayoclinic.org/diseases-conditions/breast-cancer/in-depth/breast-cancer/art-20045654
- Molecular Subtypes of Breast Cancer. (2021). Retrieved 5 April 2021, from https://www.breastcancer.org/symptoms/types/molecular-subtypes

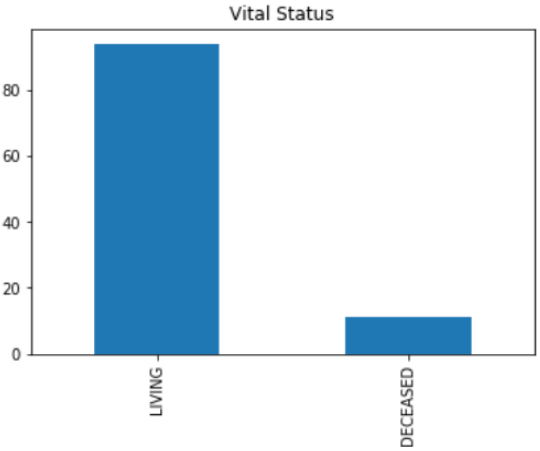
Appendix

Few visualization on important columns

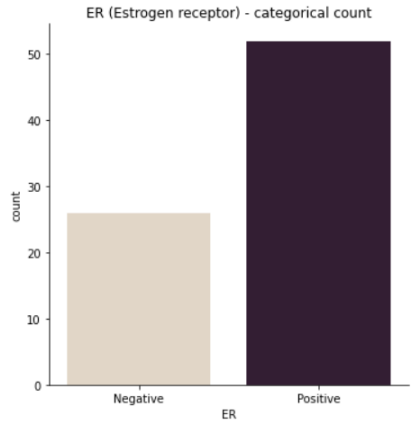




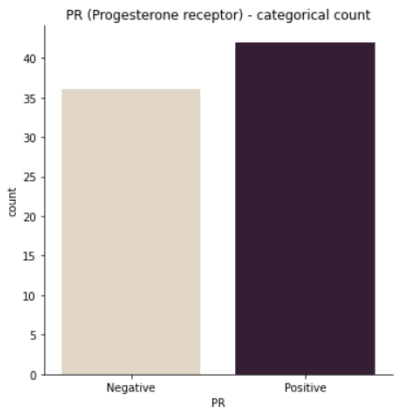




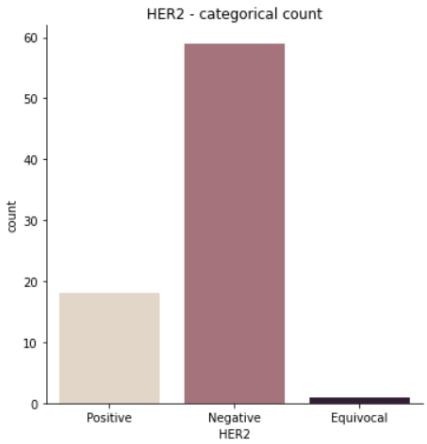
ER - Bar Plot



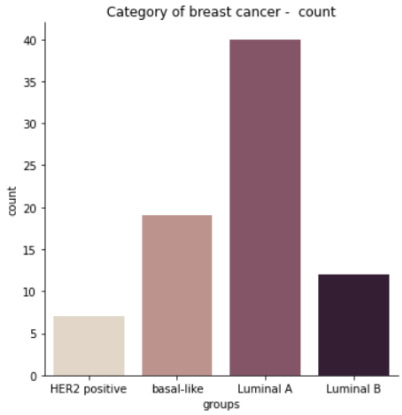
PR - Bar Plot



HER2 - Bar Plot



Category of Breast Cancer - Bar Plot




```
library(tidyverse)
library(stringr)
library(dplyr)
library(ggplot2)
library(factoextra)
library(NbClust)
library(caret)
library(caTools)
library(corrplot)
library(cluster)
library(e1071)
library(lattice)
##### read files
```

```
bcancer_raw <- read.csv("~/Desktop/DANA 4840/project/77_cancer_proteomes_CPTAC_itraq.csv")
patient <- read.csv("~/Desktop/DANA 4840/project/clinical_data_breast_cancer.csv")
pam50 <- read.csv("~/Desktop/DANA 4840/project/PAM50_proteins.csv")


################ Combining Datasets ###################
bcancer <- data.frame(t(bcancer_raw[-1]))
colnames(bcancer) <- bcancer_raw[,1]
bcancer <- bcancer[3:(nrow(bcancer)-3), ]
TCGA <- rownames(bcancer)
bcancer$TCGA <- TCGA
bcancer <- tibble::rowid_to_column(bcancer, "ID")

#write.csv(bcancer,"~/Desktop/DANA 4840/project/bcancer.csv")

# cleaning for TCGA in two csv file

bcancer$TCGA<-str_sub(bcancer$TCGA, end=-8)
bcancer$TCGA<-gsub('[[:punct:] ]+','',bcancer$TCGA)
patient$Complete.TCGA.ID <- str_sub(patient$Complete.TCGA.ID, start = 5)
patient$Complete.TCGA.ID <- gsub('[[:punct:] ]+','',patient$Complete.TCGA.ID)
colnames(patient)[1]<-'TCGA'

# Drop columns has more than 50% NAs
bcancer <- bcancer[, which(colMeans(!is.na(bcancer)) > 0.5)]
#mice_data<-mice(bcancer,m=1,maxit=2,seed=333)
#bcancer<-complete(mice_data,1)



#install.packages('varhandle')
library('varhandle')
bcancer[,1:11512]<- unfactor(bcancer[,1:11512])
for(i in 2:11512){
  bcancer[is.na(bcancer[,i]), i] <- median(bcancer[,i], na.rm = TRUE)
}
View(bcancer)

# Left-join two dataset
combined_bcancer <- left_join(bcancer, patient, by = 'TCGA', copy = FALSE)

View(combined_bcancer)
write.csv(combined_bcancer,"~/Desktop/DANA 4840/project/combined_bcancer.csv")
```

```
########################## Start From Here ######################


df <- read.csv('../Group Project/combined_bcancer (1).csv')
PAM50 <- read.csv('../Group Project/PAM50_proteins.csv')

df <- df[which(df$Gender != 'MALE'), ] #Removing Males

df$HER2.Final.Status <- ifelse(df$HER2.Final.Status == 'Equivocal'|df$HER2.Final.Status == 'Positive', 'Positive', 'Negative')
### Matching ids with PAM50 dataset

id_PAM50 <- PAM50$RefSeqProteinID

id_bcancer <- colnames(df)

new_df_feature_important <- df[which(id_bcancer %in% id_PAM50)] %>% scale()

dim(new_df_feature_important)

head(new_df_feature_important)



########################## Predicting Hormone status of Breast Cancers ##############################



######### ER (Estrogen receptor) (Positive or Negative) ###############

df_with_nine_dimensions <- data.frame(new_df_feature_important)

pca <- prcomp(df_with_nine_dimensions)

df_with_nine_dimensions <- pca$x[,1:9] #Selecting first 9 components
df_with_nine_dimensions <- data.frame(df_with_nine_dimensions)

# 1 means Positive ER  and 0 means Negative
df_with_nine_dimensions['ER'] <- ifelse(df$ER.Status == 'Positive', 1, 0)
```

```
#Creating a model by taking random samples multiple times
set.seed(123133456)
trainResult <- rep(0, 5)
testResult <- rep(0, 5)
for (i in c(1:5)) {
  sample_size <- sample.split(df_with_nine_dimensions, SplitRatio = 8 / 10)
  train <- subset(df_with_nine_dimensions, sample_size == T)
  test <- subset(df_with_nine_dimensions, sample_size == F)

  table(train$ER)

  logistic_model_train <-
    glm(formula =  ER ~ .,
        family = "binomial",
        data = train)
  summary(logistic_model_train)

  #Train set Predictions
  tainset_predicted_values <-
    predict(logistic_model_train, train, type = 'response')
  predicted <- ifelse(tainset_predicted_values > 0.5, 1, 0)
  confusion_matrix_train <-
    table(Predicted = predicted, Actual = train$ER)

  trainResult[i] <-
    sum(diag(confusion_matrix_train)) / sum(confusion_matrix_train)

  #Test Set Predictions
  testset_predicted_values <-
    predict(logistic_model_train, test, type = 'response')
  predicted <- ifelse(testset_predicted_values > 0.5, 1, 0)
  confusion_matrix_test <-
    table(Predicted = predicted, Actual = test$ER)

  testResult[i] <-
    sum(diag(confusion_matrix_test)) / sum(confusion_matrix_test)

}
Actual <- ifelse(test$ER == 0, 'ER Negative', 'ER Positive')
Predicted <- ifelse(predicted == 0, 'ER Negative', 'ER Positive')
table(Actual, Predicted)

cat("Average Train Accuracy = ",mean(trainResult))
```

```
cat("Average Test Accuracy = ",mean(testResult))



######### PR (Progestrone receptor) (Positive or Negative) ###############

df_with_nine_dimensions['ER'] <- NULL

df_with_nine_dimensions['PR'] <- ifelse(df$PR.Status == 'Positive', 1, 0)


df_with_nine_dimensions$PR%>% table()

trainResult <- rep(0, 5)
testResult <- rep(0, 5)
for (i in c(1:5)) {
  sample_size <- sample.split(df_with_nine_dimensions, SplitRatio = 9 / 10)
  train <- subset(df_with_nine_dimensions, sample_size == T)
  test <- subset(df_with_nine_dimensions, sample_size == F)


  logistic_model_train <-
    glm(formula =  PR ~ .,
        family = "binomial",
        data = train)
  summary(logistic_model_train)

  #Train set Predictions
  tainset_predicted_values <-
    predict(logistic_model_train, train, type = 'response')
  predicted <- ifelse(tainset_predicted_values > 0.5, 1, 0)
  confusion_matrix_train <-
    table(Predicted = predicted, Actual = train$PR)

  trainResult[i] <-
    sum(diag(confusion_matrix_train)) / sum(confusion_matrix_train)

  #Test Set Predictions
  testset_predicted_values <-
    predict(logistic_model_train, test, type = 'response')
  predicted <- ifelse(testset_predicted_values > 0.5, 1, 0)
  confusion_matrix_test <-
    table(Predicted = predicted, Actual = test$PR)
```

```
  testResult[i] <-
    sum(diag(confusion_matrix_test)) / sum(confusion_matrix_test)

}

Actual <- ifelse(test$PR == 0, 'PR Negative', 'PR Positive')
Predicted <- ifelse(predicted == 0, 'PR Negative', 'PR Positive')
cat("Average train Accuracy = ",mean(trainResult))
cat("Average Test Accuracy = ",mean(testResult))




################### HER2 ########################

df_with_nine_dimensions$PR <- NULL
df_with_nine_dimensions['HER2'] <- ifelse(df$HER2.Final.Status == 'Negative', 0, 1)

set.seed(123123)
trainResult <- rep(0, 5)
testResult <- rep(0, 5)

for (i in c(1:5)) {
  sample_size <- sample.split(df_with_nine_dimensions, SplitRatio = 8 / 10)
  train <- subset(df_with_nine_dimensions, sample_size == T)
  test <- subset(df_with_nine_dimensions, sample_size == F)

  logistic_model_train <-
    glm(formula =  HER2 ~ .,
        family = "binomial",
        data = train)
  summary(logistic_model_train)

  #Train set Predictions
  tainset_predicted_values <-
    predict(logistic_model_train, train, type = 'response')
  predicted <- ifelse(tainset_predicted_values > 0.5, 1, 0)
  confusion_matrix_train <-
    table(Predicted = predicted, Actual = train$HER2)

  trainResult[i] <-
    sum(diag(confusion_matrix_train)) / sum(confusion_matrix_train)
```

```
  #Test Set Predictions
  testset_predicted_values <-
    predict(logistic_model_train, test, type = 'response')
  predicted <- ifelse(testset_predicted_values > 0.5, 1, 0)
  confusion_matrix_test <-
    table(Predicted = predicted, Actual = test$HER2)

  testResult[i] <-
    sum(diag(confusion_matrix_test)) / sum(confusion_matrix_test)

}

Actual <- ifelse(test$HER2 == 0, 'HER2 Negative', 'HER2 Positive')
Predicted <- ifelse(predicted == 0, 'HER2 Negative', 'HER2 Positive')

cat("Average Train accuracy = ",mean(trainResult))


cat("Average Test accuracy = ",mean(testResult))




######## k-mean for ER, PR, HER2 positive #####

##### For K = 2 #########

set.seed(778513)
model <- kmeans(new_df_feature_important, centers = 2, iter.max = 500, nstart = 5)
model$betweenss/model$totss

fviz_cluster(model, new_df_feature_important)

table(model$cluster)

df_with_nine_dimensions %>% ggplot(aes(x = PC1, y = PC2)) + geom_point(color = "peru")


table(Clusters = model$cluster, df$ER)
table(Clusters = model$cluster, df$PR.Status)
table(Clusters = model$cluster, df$HER2.Final.Status)
```

```
######## K-mean for Breast Cancer Categories for K = 4 ########

original_df <- new_df_feature_important
set.seed(778513)
model <- kmeans(original_df, centers = 4, iter.max = 500, nstart = 5)
model$betweenss/model$totss
fviz_cluster(model, original_df)
table(model$cluster, df$PAM50.mRNA)
treatement <- new_df_feature_important


######## Model to predict breast cancer group (SVM) ##########

library(e1071)

breast_cancer_patients <- new_df_feature_important
breast_cancer_patients <- data.frame(breast_cancer_patients)
breast_cancer_patients$cancerType <- df$PAM50.mRNA

#This data set contain data about proteins and breast cancer group
head(breast_cancer_patients)
table(breast_cancer_patients$cancerType)

#This model has overfiting problem
set.seed(778)
sample_size <-sample.split(breast_cancer_patients, SplitRatio = 8 / 10)
train <- subset(breast_cancer_patients, sample_size == T)
test <- subset(breast_cancer_patients, sample_size == F)

svmfit = svm(
  cancerType ~ .,
  data = train,
  kernel = "linear",
  cost = 1,
  scale = TRUE
)

#train predictions
predicted_values <- predict(svmfit, train)
confusion_matrix_test <-
  table(Predicted = predicted_values, Actual = train$cancerType)
print(confusion_matrix_test)
sum(diag(confusion_matrix_test)) / sum(confusion_matrix_test)
```

```
#test predictions
predicted_values <- predict(svmfit, test)
confusion_matrix_test <-
  table(Predicted = predicted_values, Actual = test$cancerType)
print(confusion_matrix_test)
sum(diag(confusion_matrix_test)) / sum(confusion_matrix_test)



##### SVM model for Comparison with k-mean  #######

breast_cancer_patients$cancerType <- as.character(breast_cancer_patients$cancerType)
breast_cancer_patients <- breast_cancer_patients[which(breast_cancer_patients$cancerType !=
'HER2-enriched'),]
breast_cancer_patients$cancerType <- as.factor(breast_cancer_patients$cancerType)

pca <- prcomp(breast_cancer_patients[-length(breast_cancer_patients)])
features <- pca$x[,1:2]
features <- data.frame(features)
features$cancerType <- breast_cancer_patients$cancerType



svmfit <- svm(
  cancerType ~ .,
  data = features,
  kernel = "radial",
  cost = 1,
  scale = TRUE
)


predictions <- predict(svmfit, features)
confusion_matrix_test <-
  table(Predicted = predictions, Actual = features$cancerType)
print(confusion_matrix_test)
sum(diag(confusion_matrix_test)) / sum(confusion_matrix_test)



##### SVM model for making predictions ########
set.seed(99881)
sample_size <-sample.split(features, SplitRatio = 8 / 10)
train <- subset(features, sample_size == T)
```

```
test <- subset(features, sample_size == F)

svmfit <- svm(
  cancerType ~ .,
  data = train,
  kernel = "radial",
  cost = 1,
  scale = TRUE
)


plot(svmfit , train)

#train set predictions
predictions <- predict(svmfit, train)
confusion_matrix_test <-
  table(Predicted = predictions, Actual = train$cancerType)
print(confusion_matrix_test)
sum(diag(confusion_matrix_test)) / sum(confusion_matrix_test)

#test set predictions
predictions <- predict(svmfit, test)
confusion_matrix_test <-
  table(Predicted = predictions, Actual = test$cancerType)
print(confusion_matrix_test)

sum(diag(confusion_matrix_test)) / sum(confusion_matrix_test)


###################### K-mean (without HER2) ######################

fviz_nbclust(breast_cancer_patients[-length(breast_cancer_patients)], kmeans, method = "wss")
+
  geom_vline(xintercept = 2, linetype = 2) + # add line for better visualization
  labs(subtitle = "Elbow method") # add subtitle

# Silhouette method
fviz_nbclust(breast_cancer_patients[-length(breast_cancer_patients)], kmeans, method =
"silhouette") +
  labs(subtitle = "Silhouette method")


# NbClust function
nbclust_out <- NbClust(breast_cancer_patients[-length(breast_cancer_patients)],
```

```
                distance = "euclidean",
                min.nc = 2, max.nc = 4,
                method = "kmeans")


model <- kmeans(breast_cancer_patients[-length(breast_cancer_patients)], centers = 3)
model$betweenss/model$totss
fviz_cluster(model, breast_cancer_patients[-length(breast_cancer_patients)])
table(model$cluster, breast_cancer_patients$cancerType)




############################################################ Hierarchical
##########################################


######## Ward Method #######

#K = 4
dis_matrix <- dist(treatement, method = "euclidean")
d.ward <- hclust(dis_matrix, method = "ward.D2")
plot(d.ward, main = "Ward Linkage Clustering")
res <- cutree(d.ward, k = 4)
table(Clusters = res, df$PAM50.mRNA)

#K = 3 and without HER2
dis_matrix <- dist(breast_cancer_patients[-length(breast_cancer_patients)], method =
"euclidean")
d.ward <- hclust(dis_matrix, method = "ward.D2")
plot(d.ward, main = "Ward Linkage Clustering")
res <- cutree(d.ward, k = 3)
table(Clusters = res, breast_cancer_patients$cancerType)



##### Single Linkage #####
dis_matrix <- dist(treatement, method = "euclidean")
d.single <- hclust(dis_matrix, method = "single")
plot(d.single, main = "Single Linkage Clustering")
res <- cutree(d.single, k = 4)
```

```
table(Clusters = res, df$PAM50.mRNA)


##### Average Linkage #####
dis_matrix <- dist(treatement, method = "euclidean")
d.average <- hclust(dis_matrix, method = "average")
plot(d.average, main = "Average Linkage Clustering")
res <- cutree(d.average, k = 4)
table(Clusters = res, df$PAM50.mRNA)


###### Complete Linkage ############
dis_matrix <- dist(treatement, method = "euclidean")
d.complete <- hclust(dis_matrix, method = "complete")
plot(d.complete, main = "Complete Linkage Clustering")
res <- cutree(d.complete, k = 4)
table(Clusters = res, df$PAM50.mRNA)
```