Breast Cancer Prediction Using K-Nearest Neighbors (KNN)

Understanding the Algorithm and Its Practical Application

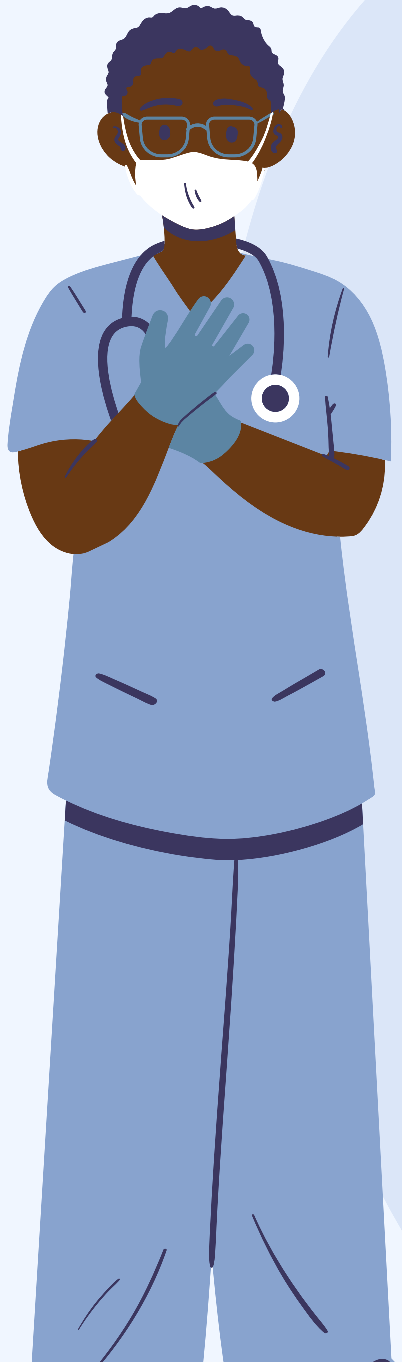
What is KNN ?

The K-Nearest Neighbors (KNN) algorithm is a supervised machine learning method employed to tackle classification and regression problems.KNN is one of the most basic yet essential classification algorithms in machine learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection.

Key Characteristics:

- Instance-based learning
- No explicit training phase
- Classification based on majority vote of neighbors

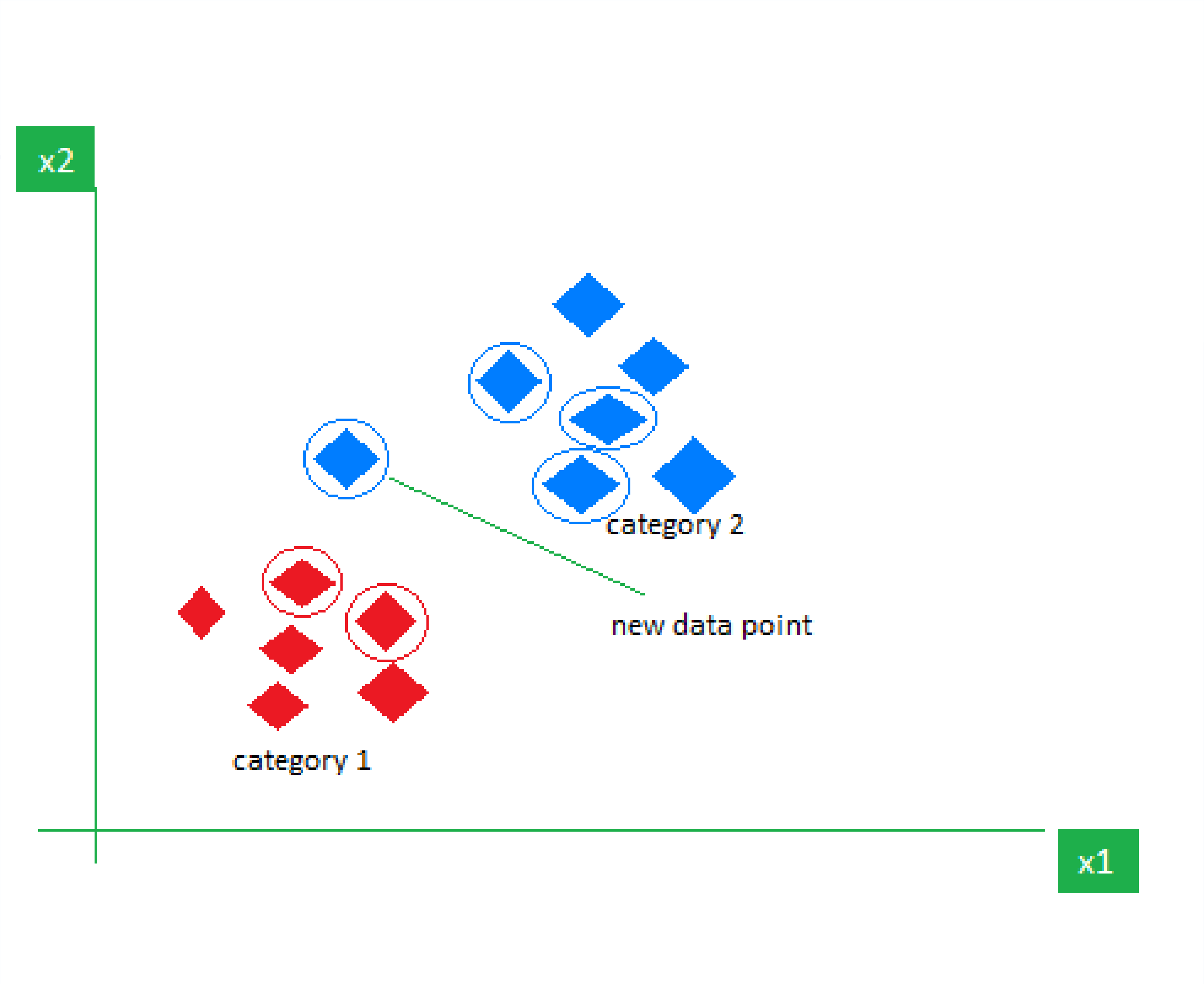
Brief history of the kNN algorithm



kNN was first developed by Evelyn Fix and Joseph Hodges in 1951 in the context of research performed for the US militaryl. They published a paper explaining discriminant analysis, which is a non-parametric classification method. In 1967, Thomas Cover and Peter Hart expanded on the non-parametric classification method and published their "Nearest Neighbor Pattern Classification" paper2. Almost 20 years later, the algorithm was refined by James Keller, who developed a "fuzzy KNN" that produces lower error rates3.

Today, the kNN algorithm is the most widely used algorithm due to its adaptability to most fields – from genetics to finance and customer service.

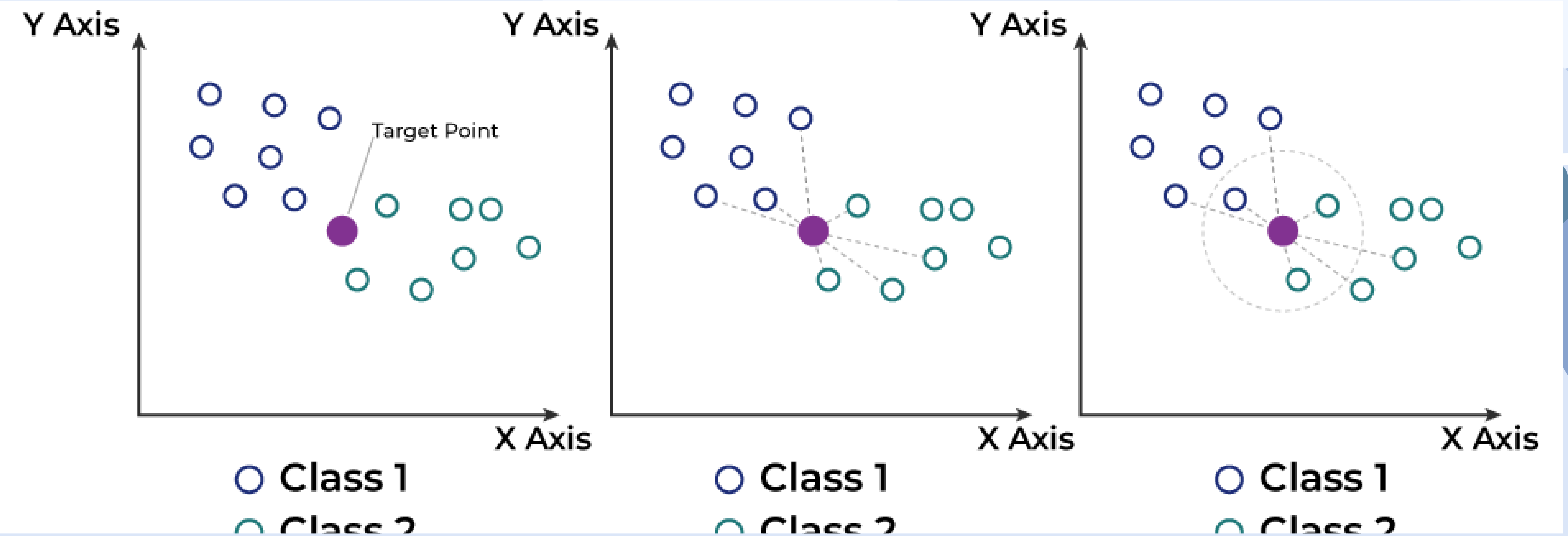
How KNN works?



1. Select K: Choose the number of neighbors (k).
2. Calculate Distance: Compute the distance between the new data point and all training points (e.g., Euclidean distance).
3. Identify Neighbors: Select the k nearest neighbors to the new data point.
4. Vote: For classification, each neighbor votes for their class, and the most common class is assigned. For regression, the average of the neighbors' values is taken.

Workings of KNN algorithm

The K-Nearest Neighbors (KNN) algorithm operates on the principle of similarity, where it predicts the label or value of a new data point by considering the labels or values of its K nearest neighbors in the training dataset.



4 types of computing kNN distance metrics

The key to the kNN algorithm is determining the distance between the query point and the other data points.

- **Euclidean distance** is the most common distance measure, which measures a straight line between the query point and the other point being measured.
- **Manhattan distance** is also a popular distance measure, which measures the absolute value between two points. It is represented on a grid, and often referred to as taxicab geometry
- **Minkowski distance** is a generalization of Euclidean and Manhattan distance metrics, which enables the creation of other distance metrics.
- **Hamming distance**, also referred to as the overlap metric, is a technique used with Boolean or string vectors to identify where vectors do not match. In other words, it measures the distance between two strings of equal length. It is especially useful for error detection and error correction codes.

Why do we need a KNN algorithm?

(K-NN) algorithm is a versatile and widely used machine learning algorithm that is primarily used for its simplicity and ease of implementation. It does not require any assumptions about the underlying data distribution. It can also handle both numerical and categorical data, making it a flexible choice for various types of datasets in classification and regression tasks. It is a non-parametric method that makes predictions based on the similarity of data points in a given dataset. K-NN is less sensitive to outliers compared to other algorithms.

Breast Cancer Prediction Using K-Nearest Neighbors (KNN)

Introduction

Breast cancer is one of the most common types of cancer in the world and the breast cancer causes death. To reduce the mortality rate caused by breast cancer, machine learning plays great role in breast cancer identification process. Machine learning algorithms are applied to develop an intelligent system which can identify breast cancer in the early stage as possible in order to reduce the complications and increase survival rate of the patients

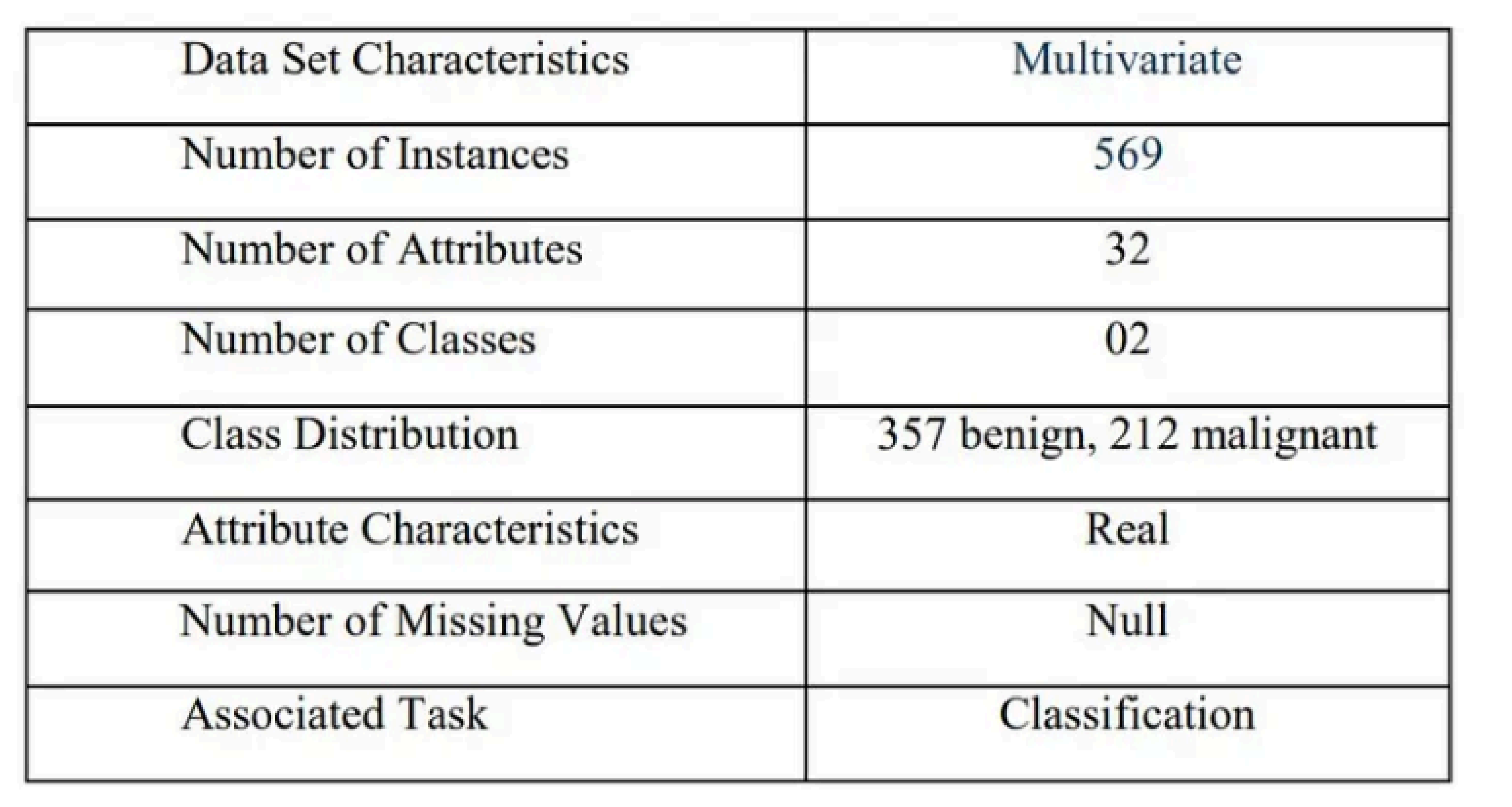
Step-by-Step Process:

- Data Collection: Breast Cancer (Diagnostic) Dataset.
- Data Preprocessing: Cleaning, normalizing, and splitting data into training and test sets.
- Selecting K: Using cross-validation to find the optimal k.
- Model Training: Using training data to train the KNN model.
- Prediction: Predicting the class (benign or malignant) for new data points.
- Evaluation: Assessing the model's performance using metrics like accuracy, precision, recall, and F1-score.

DETAILED EXPLANATION

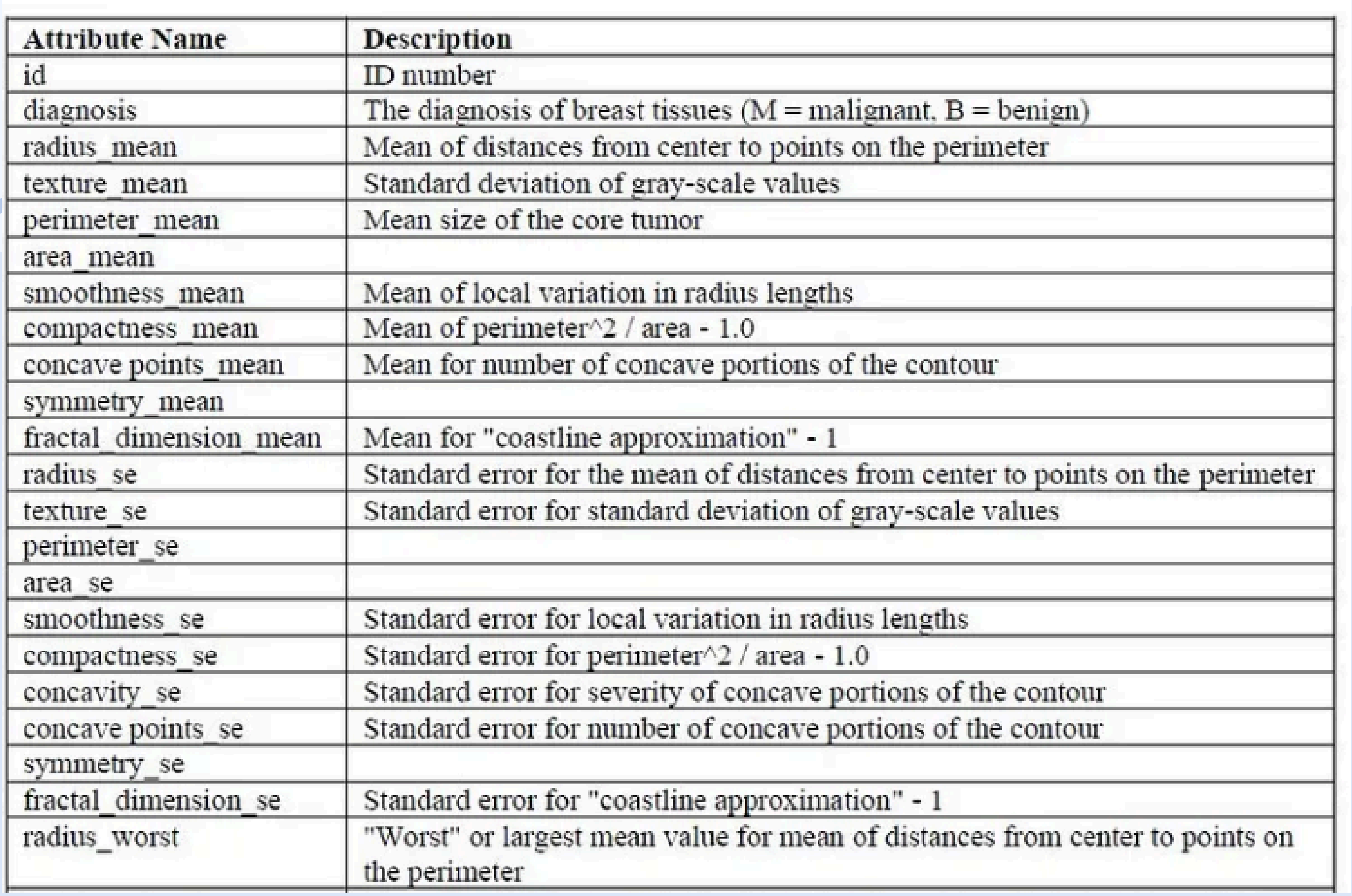
DATA COLLECTION

To create the classification of breast cancer stages and to train the model using the KNN algorithm for predict breast cancers, as the initial step we need to find a dataset. This database is posted on the Kaggle.com web site using the UCI machine learning repository and the database is obtained from the University of Wisconsin Hospitals.



Data Set Characteristics	Multivariate
Number of Instances	569
Number of Attributes	32
Number of Classes	02
Class Distribution	357 benign, 212 malignant
Attribute Characteristics	Real
Number of Missing Values	Null
Associated Task	Classification

Description of the Attributes



Attribute Name	Description
id	ID number
diagnosis	The diagnosis of breast tissues (M = malignant, B = benign)
radius_mean	Mean of distances from center to points on the perimeter
texture_mean	Standard deviation of gray-scale values
perimeter_mean	Mean size of the core tumor
area_mean	
smoothness_mean	Mean of local variation in radius lengths
compactness_mean	Mean of perimeter^2 / area - 1.0
concave points_mean	Mean for number of concave portions of the contour
symmetry_mean	
fractal_dimension_mean	Mean for "coastline approximation" - 1
radius_se	Standard error for the mean of distances from center to points on the perimeter
texture_se	Standard error for standard deviation of gray-scale values
perimeter_se	
area_se	
smoothness_se	Standard error for local variation in radius lengths
compactness_se	Standard error for perimeter^2 / area - 1.0
concavity_se	Standard error for severity of concave portions of the contour
concave points_se	Standard error for number of concave portions of the contour
symmetry_se	
fractal_dimension_se	Standard error for "coastline approximation" - 1
radius_worst	"Worst" or largest mean value for mean of distances from center to points on the perimeter

When considering the description of the dataset attributes “Malignant (M)” and “Benign (B)” are the two classes in this dataset which use to predict breast cancer. The alternate features represent different attributes of breast cancer risk that may be used to classify the given situation which causes breast cancer or not.

Data Preprocessing

Data preprocessing is extremely important because it allows improving the quality of the raw experimental data. (From Dataset)

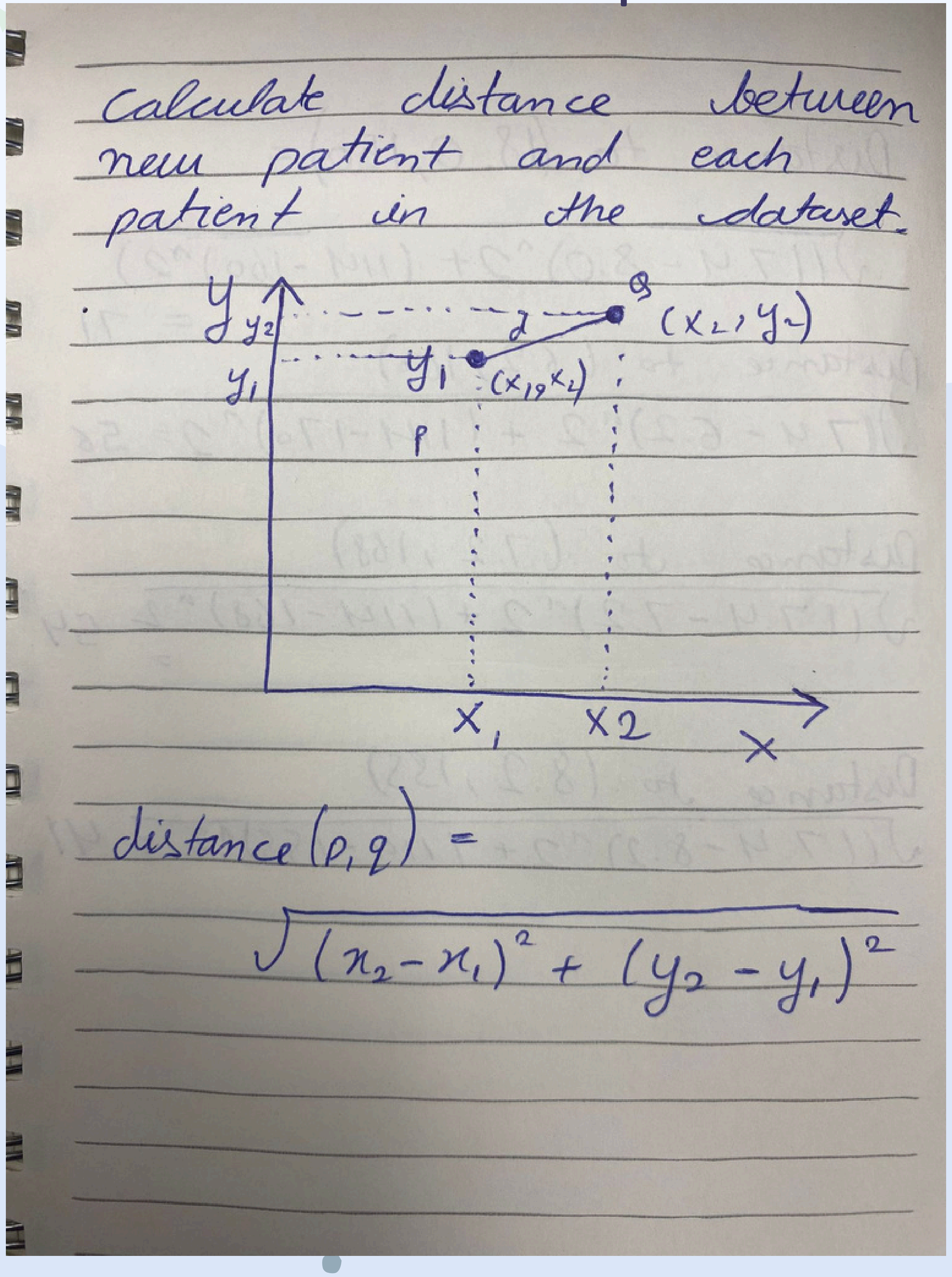
Split the data set into a testing set and training set

As the next step, we need to split the data into a training set and testing set. The training data will be used to create the KNN classifier model and the testing data will be used to test the accuracy of the classifier. Because splitting data into training and testing sets will avoid the overfitting and optimize the KNN classifier model.

Build the predictive model by implementing the K-Nearest Neighbors (KNN) algorithm

Euclidean Distance

kNN relies on measuring distances between data points to classify new instances. The Euclidean distance is a common metric used to measure the straight-line distance between two points in a Euclidean space.

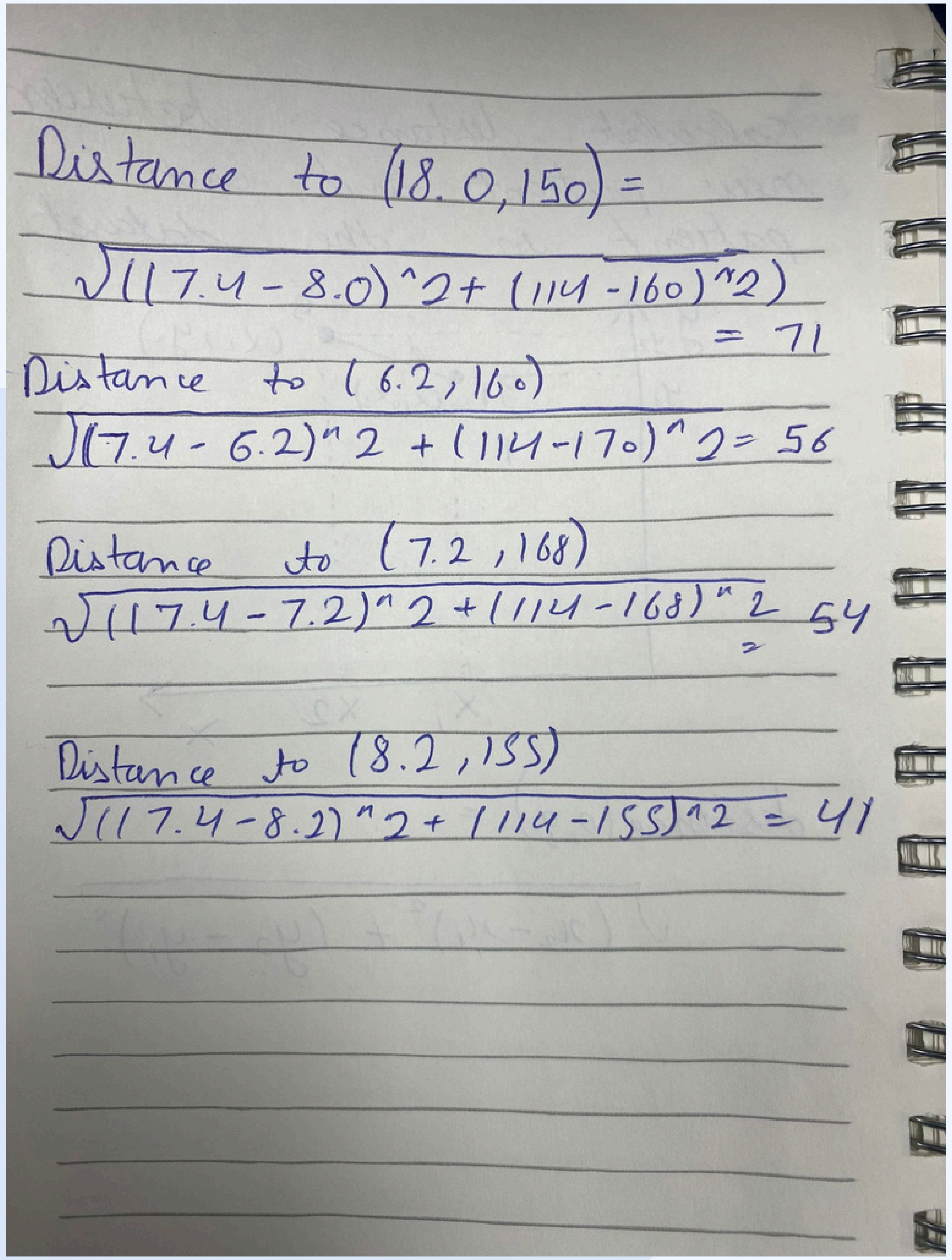


Calculate distance between new patient and each patient in the dataset.

y
y2
y1
y1
q
(x_2, y_2)
d
(x_1, x_2)
p
X1
X2
X

distance(p, q) =

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



Distance to (18.0, 150) =

$\sqrt{((17.4 - 8.0)^\wedge 2 + (114 - 160)^\wedge 2)}$
= 71

Distance to (6.2, 160)

$\sqrt{(7.4 - 6.2)^\wedge 2 + (114 - 170)^\wedge 2} = 56$

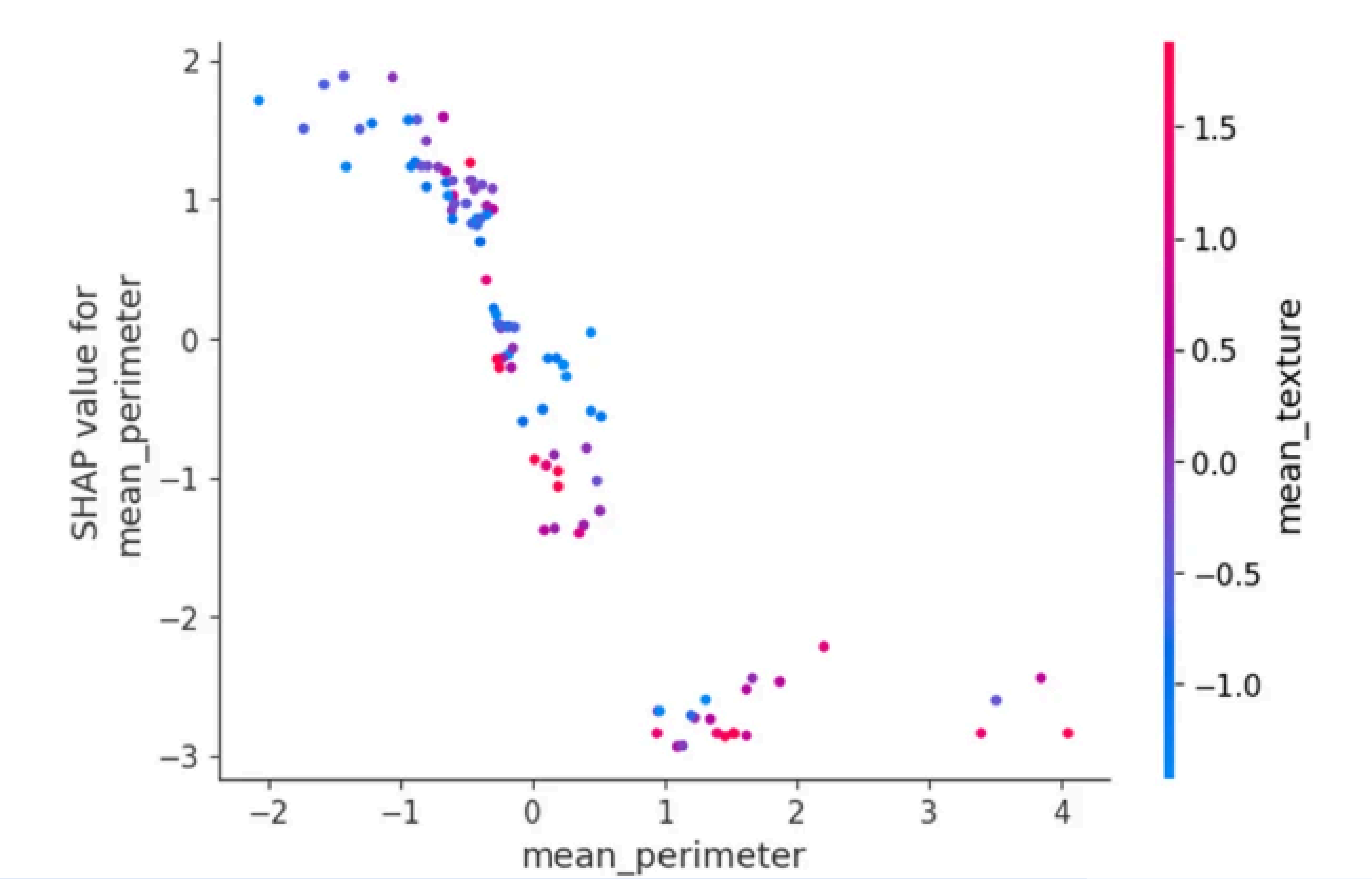
Distance to (7.2, 168)

$\sqrt{(17.4 - 7.2)^\wedge 2 + (114 - 168)^\wedge 2} = 54$

Distance to (8.2, 155)

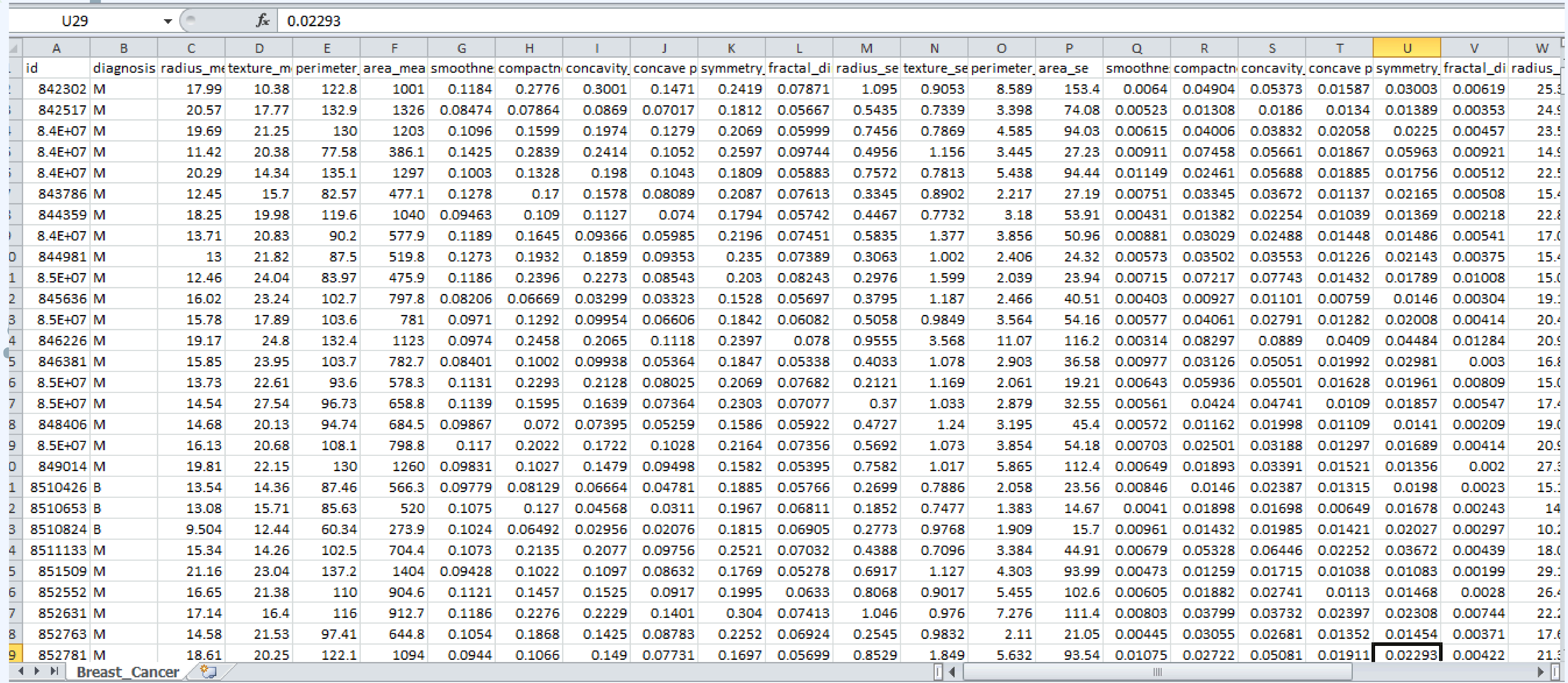
$\sqrt{(17.4 - 8.2)^\wedge 2 + (114 - 155)^\wedge 2} = 41$

Te dependence plot for mean_perimeter shows a weak positive correlation with the SHAP value, indicating that higher mean_perimeter values contribute to a higher probability of predicting early-stage breast cancer.



Our ML Project Using KNN

BREAST CANCER DATASET

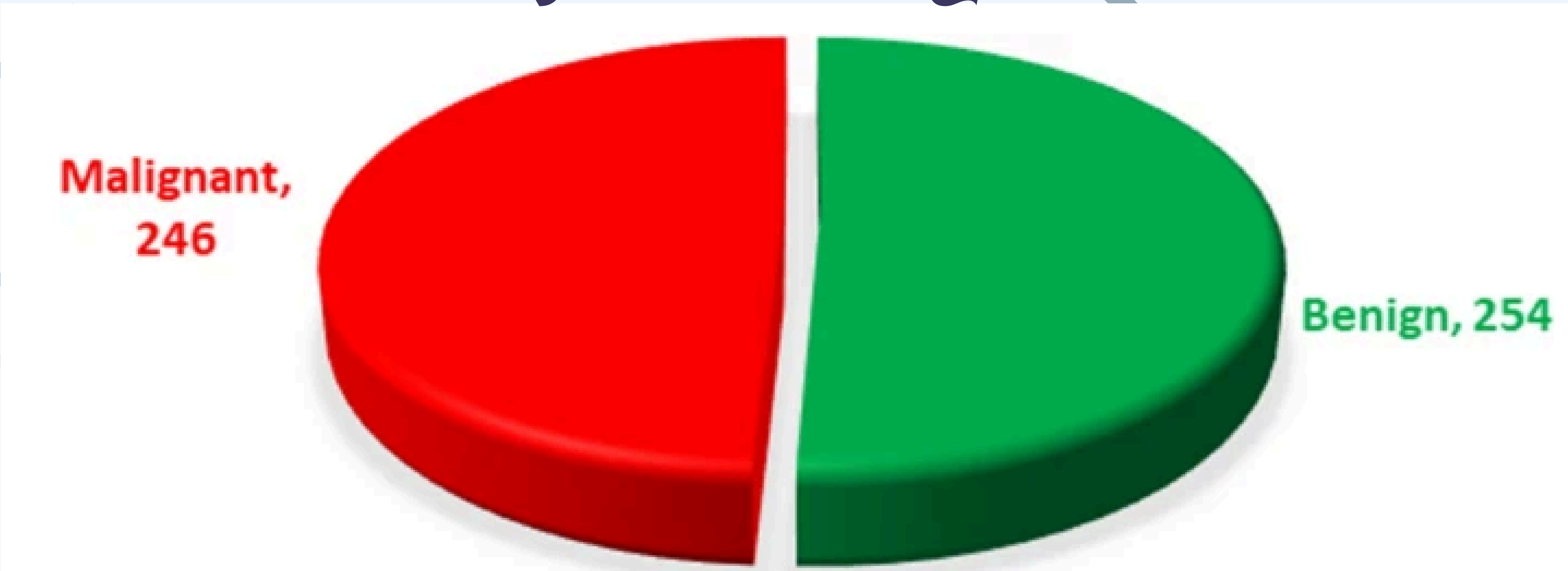


U29 0.02293

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
id	diagnosis	radius_me	texture_m	perimeter_	area_mea	smoothne	compactn	concavity_	concave p	symmetry_	fractal_di	radius_se	texture_se	perimeter_	area_se	smoothne	compactn	concavity_	concave p	symmetry_	fractal_di	radius_
842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871	1.095	0.9053	8.589	153.4	0.0064	0.04904	0.05373	0.01587	0.03003	0.00619	25.3
842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398	74.08	0.00523	0.01308	0.0186	0.0134	0.01389	0.00353	24.9
8.4E+07	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999	0.7456	0.7869	4.585	94.03	0.00615	0.04006	0.03832	0.02058	0.0225	0.00457	23.5
8.4E+07	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744	0.4956	1.156	3.445	27.23	0.00911	0.07458	0.05661	0.01867	0.05963	0.00921	14.9
8.4E+07	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883	0.7572	0.7813	5.438	94.44	0.01149	0.02461	0.05688	0.01885	0.01756	0.00512	22.5
843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613	0.3345	0.8902	2.217	27.19	0.00751	0.03345	0.03672	0.01137	0.02165	0.00508	15.4
844359	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742	0.4467	0.7732	3.18	53.91	0.00431	0.01382	0.02254	0.01039	0.01369	0.00218	22.8
8.4E+07	M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.07451	0.5835	1.377	3.856	50.96	0.00881	0.03029	0.02488	0.01448	0.01486	0.00541	17.0
844981	M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389	0.3063	1.002	2.406	24.32	0.00573	0.03502	0.03553	0.01226	0.02143	0.00375	15.4
8.5E+07	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243	0.2976	1.599	2.039	23.94	0.00715	0.07217	0.07743	0.01432	0.01789	0.01008	15.0
845636	M	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.05697	0.3795	1.187	2.466	40.51	0.00403	0.00927	0.01101	0.00759	0.0146	0.00304	19.1
8.5E+07	M	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606	0.1842	0.06082	0.5058	0.9849	3.564	54.16	0.00577	0.04061	0.02791	0.01282	0.02008	0.00414	20.4
846226	M	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	0.2397	0.078	0.9555	3.568	11.07	116.2	0.00314	0.08297	0.0889	0.0409	0.04484	0.01284	20.9
846381	M	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847	0.05338	0.4033	1.078	2.903	36.58	0.00977	0.03126	0.05051	0.01992	0.02981	0.003	16.8
8.5E+07	M	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069	0.07682	0.2121	1.169	2.061	19.21	0.00643	0.05936	0.05501	0.01628	0.01961	0.00809	15.0
8.5E+07	M	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303	0.07077	0.37	1.033	2.879	32.55	0.00561	0.0424	0.04741	0.0109	0.01857	0.00547	17.4
848406	M	14.68	20.13	94.74	684.5	0.09867	0.072	0.07395	0.05259	0.1586	0.05922	0.4727	1.24	3.195	45.4	0.00572	0.01162	0.01998	0.01109	0.0141	0.00209	19.0
8.5E+07	M	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028	0.2164	0.07356	0.5692	1.073	3.854	54.18	0.00703	0.02501	0.03188	0.01297	0.01689	0.00414	20.9
849014	M	19.81	22.15	130	1260	0.09831	0.1027	0.1479	0.09498	0.1582	0.05395	0.7582	1.017	5.865	112.4	0.00649	0.01893	0.03391	0.01521	0.01356	0.002	27.3
8510426	B	13.54	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.04781	0.1885	0.05766	0.2699	0.7886	2.058	23.56	0.00846	0.0146	0.02387	0.01315	0.0198	0.0023	15.1
8510653	B	13.08	15.71	85.63	520	0.1075	0.127	0.04568	0.0311	0.1967	0.06811	0.1852	0.7477	1.383	14.67	0.0041	0.01898	0.01698	0.00649	0.01678	0.00243	14
8510824	B	9.504	12.44	60.34	273.9	0.1024	0.06492	0.02956	0.02076	0.1815	0.06905	0.2773	0.9768	1.909	15.7	0.00961	0.01432	0.01985	0.01421	0.02027	0.00297	10.2
8511133	M	15.34	14.26	102.5	704.4	0.1073	0.2135	0.2077	0.09756	0.2521	0.07032	0.4388	0.7096	3.384	44.91	0.00679	0.05328	0.06446	0.02252	0.03672	0.00439	18.0
851509	M	21.16	23.04	137.2	1404	0.09428	0.1022	0.1097	0.08632	0.1769	0.05278	0.6917	1.127	4.303	93.99	0.00473	0.01259	0.01715	0.01038	0.01083	0.00199	29.1
852552	M	16.65	21.38	110	904.6	0.1121	0.1457	0.1525	0.0917	0.1995	0.0633	0.8068	0.9017	5.455	102.6	0.00605	0.01882	0.02741	0.0113	0.01468	0.0028	26.4
852631	M	17.14	16.4	116	912.7	0.1186	0.2276	0.2229	0.1401	0.304	0.07413	1.046	0.976	7.276	111.4	0.00803	0.03799	0.03732	0.02397	0.02308	0.00744	22.2
852763	M	14.58	21.53	97.41	644.8	0.1054	0.1868	0.1425	0.08783	0.2252	0.06924	0.2545	0.9832	2.11	21.05	0.00445	0.03055	0.02681	0.01352	0.01454	0.00371	17.6
852781	M	18.61	20.25	122.1	1094	0.0944	0.1066	0.149	0.07731	0.1697	0.05699	0.8529	1.849	5.632	93.54	0.01075	0.02722	0.05081	0.01911	0.02293	0.00422	21.3

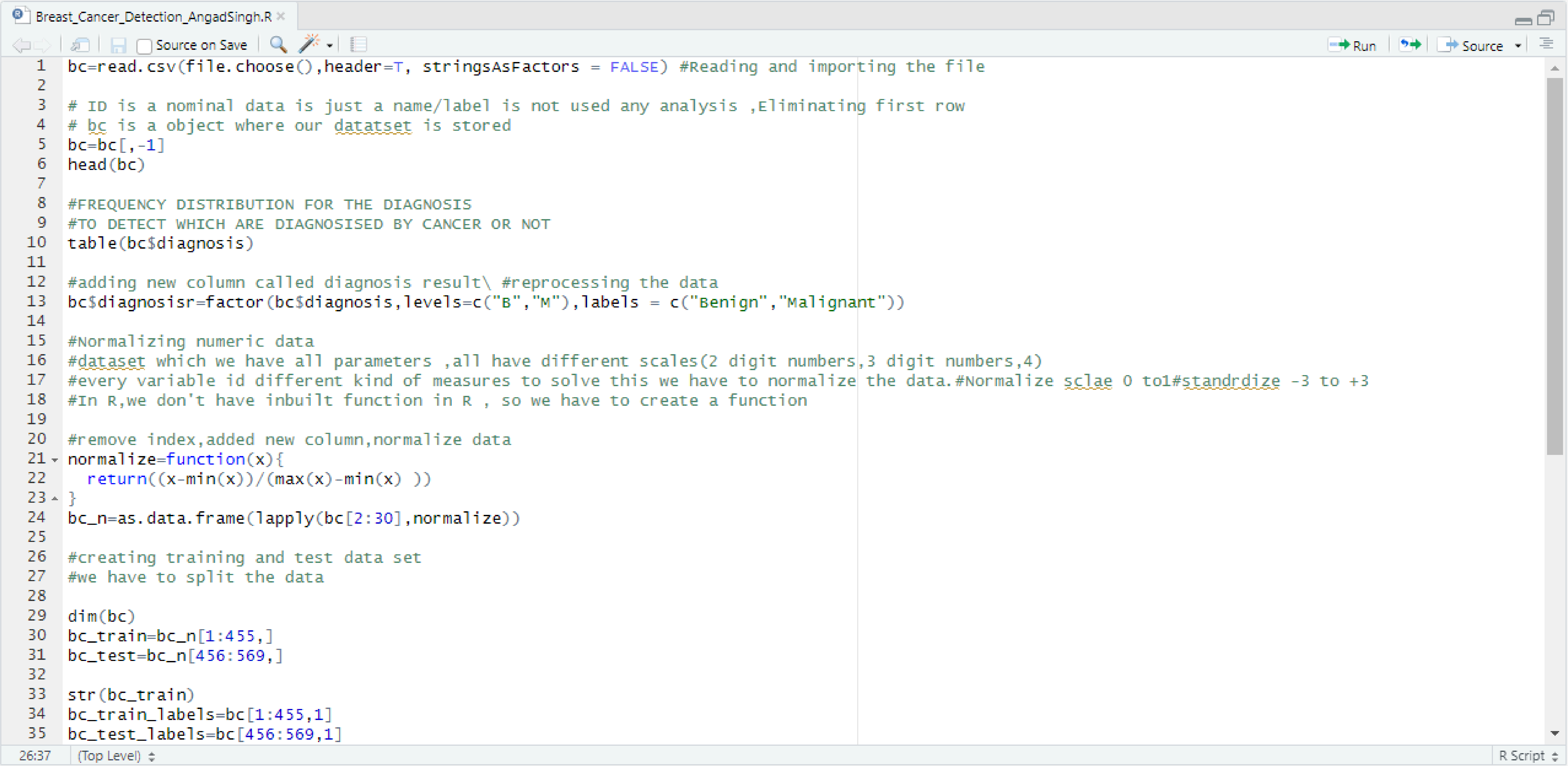
Breast_Cancer

Our ML Project Using KNN



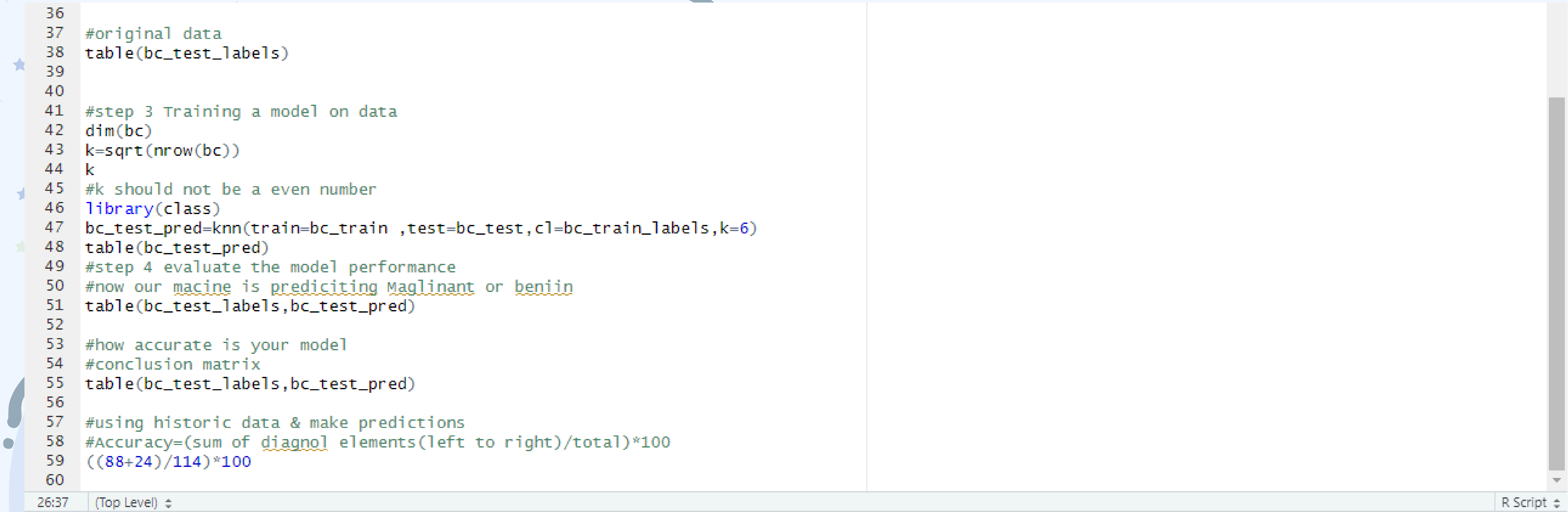
Ratio of malignant and benign data based on overall dataset.

CODE



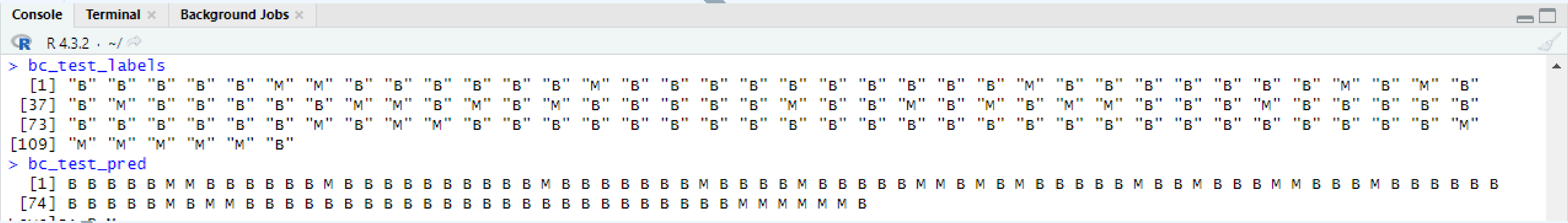
```r
bc=read.csv(file.choose(),header=T, stringsAsFactors = FALSE) #Reading and importing the file

# ID is a nominal data is just a name/label is not used any analysis ,Eliminating first row
# bc is a object where our datatset is stored
bc=bc[,-1]
head(bc)

#FREQUENCY DISTRIBUTION FOR THE DIAGNOSIS
#TO DETECT WHICH ARE DIAGNOSISED BY CANCER OR NOT
table(bc$diagnosis)

#adding new column called diagnosis result\ #reprocessing the data
bc$diagnosisr=factor(bc$diagnosis,levels=c("B","M"),labels = c("Benign","Malignant"))

#Normalizing numeric data
#dataset which we have all parameters ,all have different scales(2 digit numbers,3 digit numbers,4)
#every variable id different kind of measures to solve this we have to normalize the data.#Normalize sclae 0 to1#standrdize -3 to +3
#In R,we don't have inbuilt function in R , so we have to create a function

#remove index,added new column,normalize data
normalize=function(x){
  return((x-min(x))/(max(x)-min(x) ))
}
bc_n=as.data.frame(lapply(bc[2:30],normalize))

#creating training and test data set
#we have to split the data

dim(bc)
bc_train=bc_n[1:455,]
bc_test=bc_n[456:569,]

str(bc_train)
bc_train_labels=bc[1:455,1]
bc_test_labels=bc[456:569,1]
```

CODE



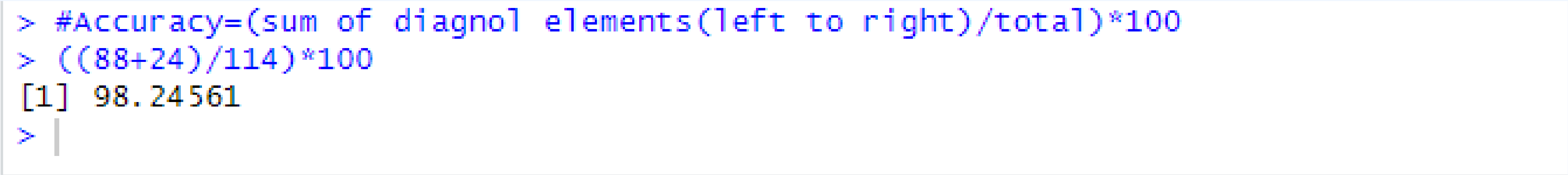
```

#original data
table(bc_test_labels)


#step 3 Training a model on data
dim(bc)
k=sqrt(nrow(bc))
k
#k should not be a even number
library(class)
bc_test_pred=knn(train=bc_train ,test=bc_test,cl=bc_train_labels,k=6)
table(bc_test_pred)
#step 4 evaluate the model performance
#now our macine is prediciting Maglinant or benijn
table(bc_test_labels,bc_test_pred)

#how accurate is your model
#conclusion matrix
table(bc_test_labels,bc_test_pred)

#using historic data & make predictions
#Accuracy=(sum of diagnol elements(left to right)/total)*100
((88+24)/114)*100

```

OUTPUT

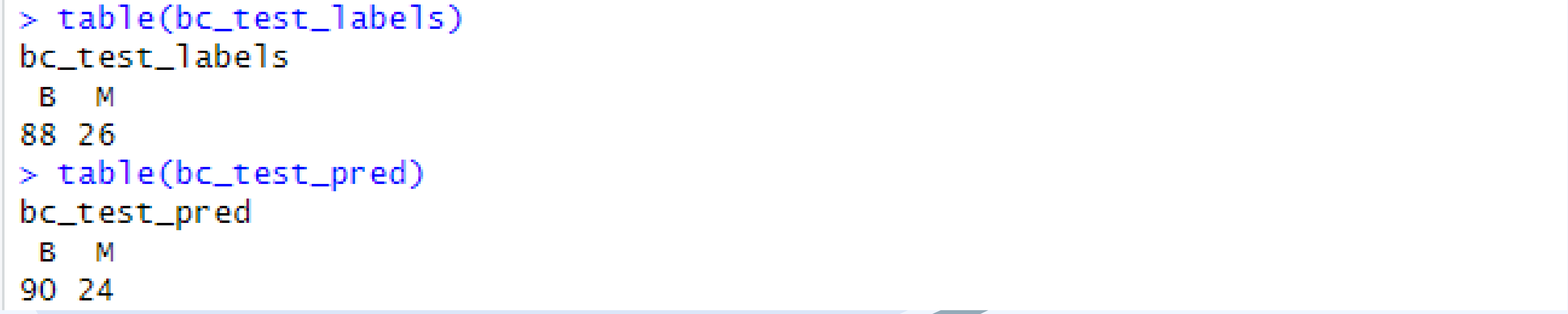


```
> bc_test_labels
  [1] "B" "B" "B" "B" "B" "M" "M" "B" "B" "B" "B" "B" "B" "M" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "M" "B" "B" "B" "B" "B" "B" "B" "M" "B" "M" "B"
 [37] "B" "M" "B" "B" "B" "B" "B" "M" "M" "B" "M" "B" "M" "B" "B" "B" "B" "B" "M" "B" "B" "M" "B" "M" "B" "M" "M" "B" "B" "B" "M" "B" "B" "B" "B" "B"
 [73] "B" "B" "B" "B" "B" "B" "M" "B" "M" "M" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "M"
[109] "M" "M" "M" "M" "M" "B"
> bc_test_pred
  [1] B B B B B M M B B B B B B M B B B B B B B B B B M B B B B B B B M B B B B M B B B B B M M B M B M B B B B B M B B M B B B M M B B B M B B B B B B
 [74] B B B B B M B M M B B B B B B B B B B B B B B B B B B B B B B B B B M M M M M M B
```

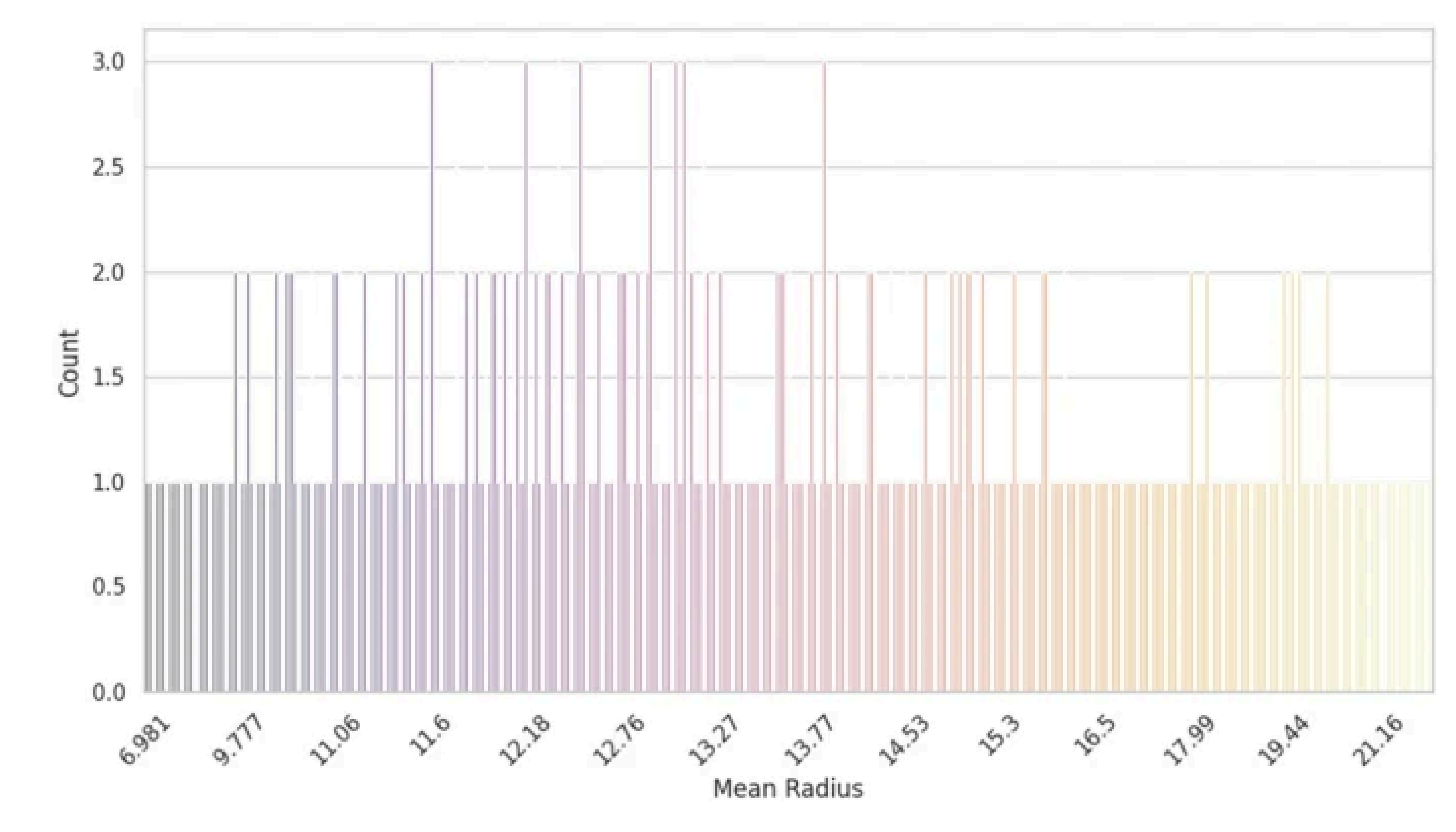
Accuracy



```
> #Accuracy=(sum of diagnol elements(left to right)/total)*100
> ((88+24)/114)*100
[1] 98.24561
> |
```



```
> table(bc_test_labels)
bc_test_labels
 B  M 
88 26 
> table(bc_test_pred)
bc_test_pred
 B  M 
90 24 
```



Patients believed to have cancer have a radius bigger than 1, whereas those without symptoms have a radius nearer to 1.

Advantages of the Application

- Easy to implement as the complexity of the algorithm is not that high.
- Adapts Easily – As per the working of the KNN algorithm it stores all the data in memory storage and hence whenever a new example or data point is added then the algorithm adjusts itself as per that new example and has its contribution to the future predictions as well.
- Few Hyperparameters – The only parameters which are required in the training of a KNN algorithm are the value of k and the choice of the distance metric which we would like to choose from our evaluation metric.

Disadvantages of the Application

- **Does not scale** : The kNN algorithm can be slow and demanding on resources because it needs to store all the training data. When it needs to make a prediction for a new data point, it compares it to every single data point in the training set to find the nearest neighbors. This can be computationally expensive, especially with large datasets.
- **Curse of Dimensionality:** This refers to a problem where as the number of features (dimensions) in your dataset increases, the amount of data needed to generalize effectively also increases exponentially. In practical terms, this means that when you have many features, the kNN algorithm might struggle to find the closest neighbors accurately because the distance between points becomes less meaningful in high-dimensional spaces.
- **Prone to Overfitting:** Overfitting happens when a model learns not just the general patterns in the data but also the noise and specifics of the training set.

Conclusion

Cancer was known to be the leading cause of death in people, especially female, which is essentially caused by the abnormal development of cancer cells. This paper presented a review of the classification algorithms used to breast cancer by implementing the KNN algorithm. Each algorithm used a different classification technique, and the dataset mostly used differed among various studies. To sum it up, the implementation of the using KNN is relatively easy. In terms of accuracy, KNN gave the most accurate prediction (98.68%). For breast cancer diagnostic by using various algorithms in the future offers a much more effective and reliable method for identifying the illness.'

Improvements

When applying the KNN classifier it offered various scores for the accuracy when the number of neighbors varied. The model gave this decent accuracy score when the optimal numbers of neighbors were 13, where the model was tested with the values in the range from 1 to 50 as the value of "K" or the number of neighbors.

References

- https://www.elastic.co/what-is/knn
- https://www.geeksforgeeks.org/k-nearest-neighbours/
- https://journal.umy.ac.id/index.php/jrc/article/view/8593/5755
- https://archives.palarch.nl/index.php/jae/article/view/6601/6403
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7173366/
- https://medium.com/@anuuz.soni/advantages-and-disadvantages-of-knn-ee06599b9336

Thank you