BREAST CANCER PREDICTION USING MACHINE LEARNING ALGORITHMS

Submitted By

Laxman Reddy Nalla,
USER ID LXN20710,
Department of Computer Science

Vinay Kumar Avirneni,
USER ID VXA45490,
Department of Computer Science

Uday Kumar Kuppam,
USER ID UXK97680
Department of Computer Science,

Gayathri Keshamoni,
USER ID GXK24880
Department of Computer Science,

Submitted To

Khan, Muhammad Zubair,
Course CS 5710
Department of Computer Science

UNIVERSITY OF CENTRAL MISSOURI
Lee Summit's, Missouri

Introduction:

Breast cancer, along with lung and bronchus cancer, prostate cancer, colon cancer, and pancreatic cancer among others, is the type of cancer that is most commonly known throughout the world. Breast cancer is the primary type of uncontrollable malignant development among women worldwide and may be a common cause of death. Breast cancer is caused by a variety of factors, such as family history, weight hormones, radiation therapy, and even reproductive factors. Every year, 2.1 million women are newly diagnosed with breast cancer, which also accounts for most women's cancer-related fatalities, according to data from the World Health Organization. According to estimates, 627,000 female cancer deaths in 2018 were attributable to breast cancer, or around 15% of all female cancer fatalities.

Algorithms for machine learning are commonly used in frameworks for intelligent human services. especially for the diagnosis and prognosis of breast cancer. In this paper, we compare different kinds of classification algorithms like k Nearest Neighbors, Support Vector Machine, Logistic Regression, and Gaussian Naive Bayes. There are many machine learning classifications and algorithms for the prediction of breast cancer outcomes. Additionally, evaluate and compare the accuracy, precision, recall, f1-Score, and Jaccard index performance of the various classifiers. The findings from this paper give an overview of the state of current machine-learning techniques for breast cancer detection.

Motivation:

- Breast Cancer has become the most common type of cancer in humans. The doctors classified cancer into 4 stages 1 to 4. If cancers are recognized in the early stages, there is a high possibility of recovery from it.
- There are new technologies where we can detect the type of cancer and its stage of it, but there are likely to be false positives and true negative results. This FP and TN can end up with the cost of human life. We can assess the FP, and TN using Machine Learning Classification algorithms with the technology.
- In this Machine Learning Project, we are proposing a comparison between ML classification algorithms that can accurately predict the type of cancer.

Significance:

- It is very important to get high prediction accuracy since a false prediction can result in the cost of human life.
- Objectives:
- Compare different classification algorithms which give high accuracy for the breast cancer prediction problem.
- Implementation of different ML classification algorithms against breast cancer datasets.

Objectives:

- Compare different classification algorithms which give high accuracy for the breast cancer prediction problem.

- Implementation of different ML classification algorithms against breast cancer datasets.

Increment:

Dataset: Breast Cancer Wisconsin (Diagnostic) Data Set | Kaggle

Dataset Description:

- Breast cancer is the most common cancer among women in the world. It accounts for 25% of all cancer cases and affected over 2.1 million people in 2015 alone. It starts when cells in the breast begin to grow out of control. These cells usually form tumors that can be seen via X-ray or felt as lumps in the breast area.
- The key challenge against its detection is how to classify tumors into malignant (cancerous) or benign(non-cancerous). We ask you to complete the analysis of classifying these tumors using machine learning (with SVMs) and the Breast Cancer Wisconsin (Diagnostic) Dataset.

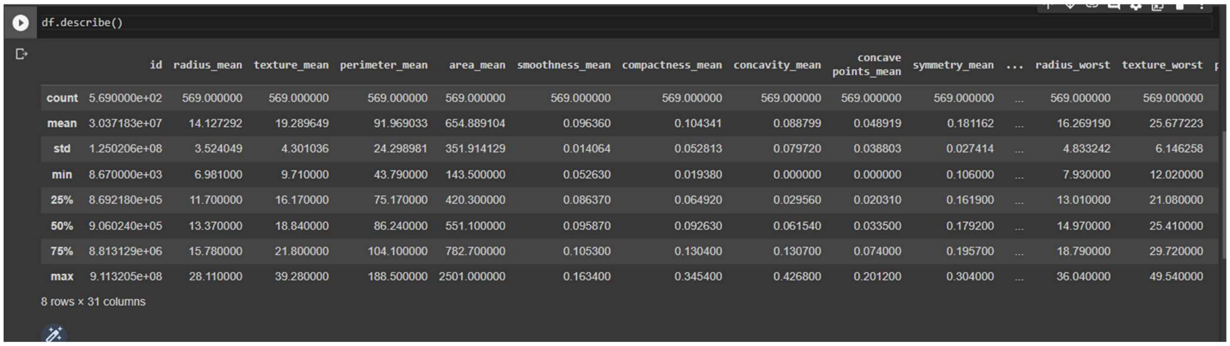
Detailed Description of Features:

1) ID number
2) Diagnosis (M = malignant, B = benign)
3) Ten real-valued features are computed for each cell nucleus:

   a) radius (mean of distances from the center to points on the perimeter)
   b) texture (standard deviation of gray-scale values)
   c) perimeter
   d) area
   e) smoothness (local variation in radius lengths)
   f) compactness (perimeter^2 / area - 1.0)
   g) concavity (severity of concave portions of the contour)
   h) concave points (number of concave portions of the contour)
   i) symmetry
   j) fractal dimension ("coastline approximation" - 1)

Features analysis:

Dataset statical information:



```
df.describe()
```

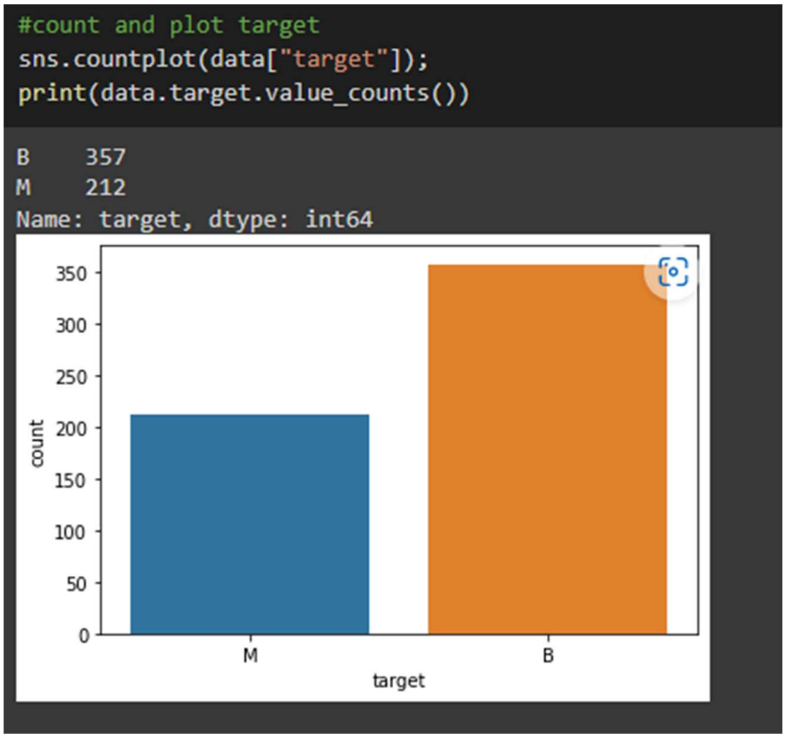
	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	...	radius_worst	texture_worst	p
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	...	569.000000	569.000000	
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162	...	16.269190	25.677223	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414	...	4.833242	6.146258	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000	...	7.930000	12.020000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900	...	13.010000	21.080000	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200	...	14.970000	25.410000	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700	...	18.790000	29.720000	
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000	...	36.040000	49.540000	

8 rows × 31 columns

Type of cancers using count plot:

Benign: 357

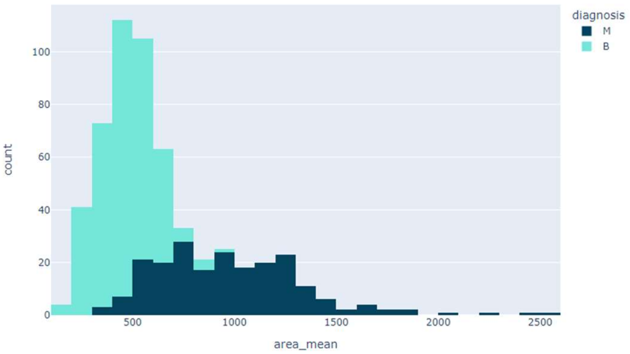
Malignant: 212



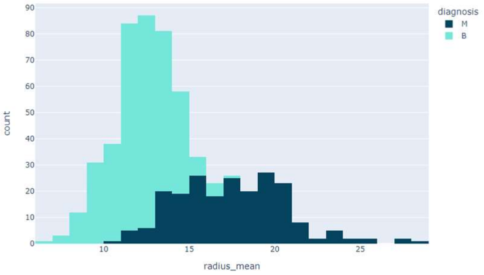
```
#count and plot target
sns.countplot(data["target"]);
print(data.target.value_counts())
```

```
B    357
M    212
Name: target, dtype: int64
```

Count Plot of Type of Cancers



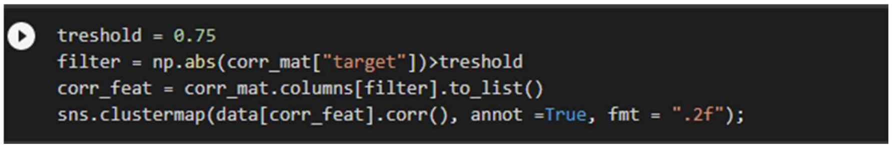
Area mean of breast cancer tumour



The radius of Cancer tumour

Correlation matrix:

The correlation matrix for the will shows the correlated features in dataset. The breast cancer dataset consists of 32 features, so it is difficult to see the correlation matrix so took the correlation matrix of features which has correlation thresholds above 0.75 with respect to Target.



```
treshold = 0.75
filter = np.abs(corr_mat["target"])>treshold
corr_feat = corr_mat.columns[filter].to_list()
sns.clustermap(data[corr_feat].corr(), annot =True, fmt = ".2f");
```

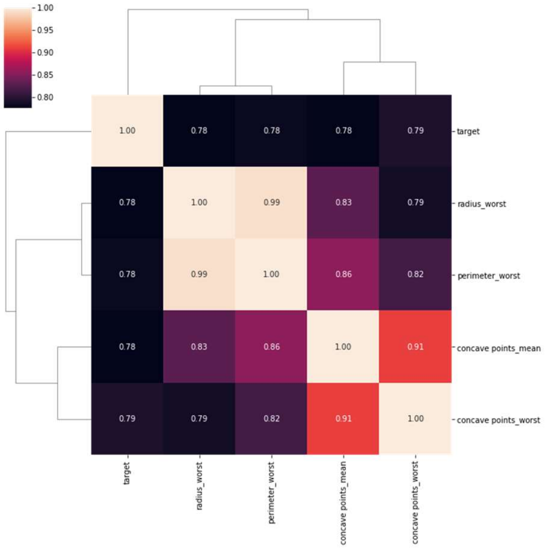
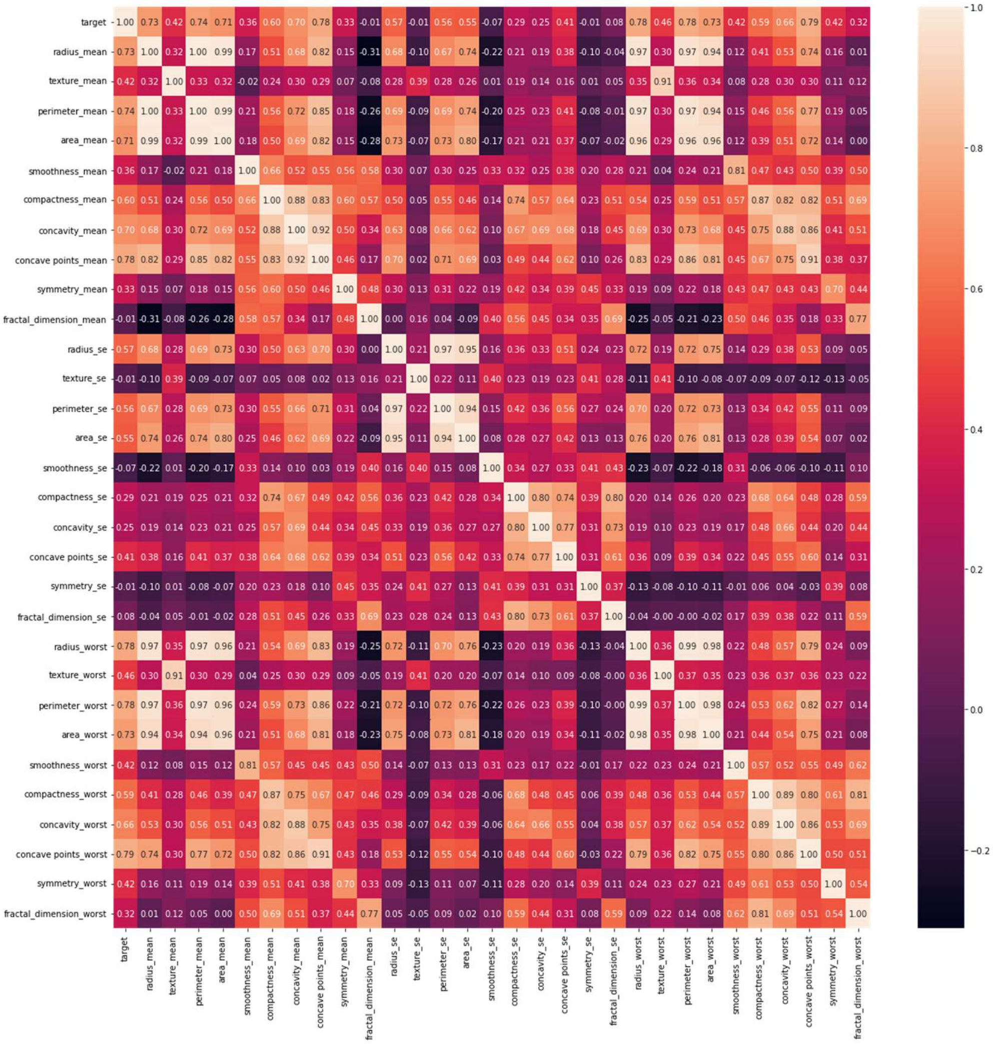


Fig: Correlation matrix of features



There is a high correlation of 0.75 and above between **radius worst, perimeter worst, concave points mean, and concave points worst** these features so we removed some features such as **Radius worst, perimeter worst, and concave points** mean these are highly related features. By removing we can reduce features in data without losing important features.

Implementation of Models:

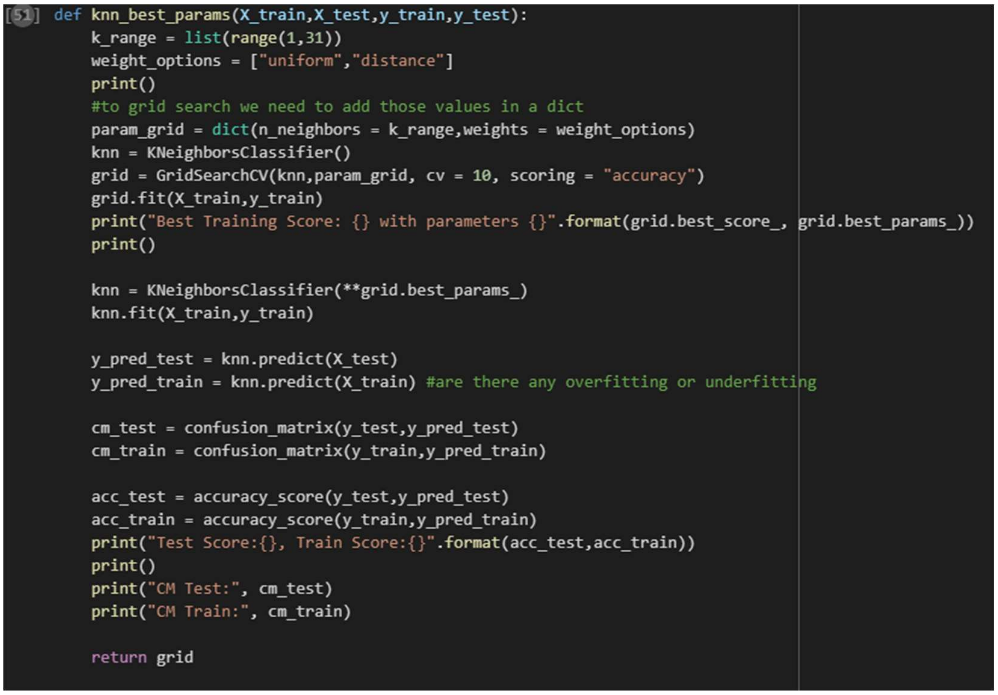
For this breast cancer data set we thought of using multiple machine-learning algorithms such as

1) PCA-KNN with n_negibours as 2.
2) NCA-KNN with n_negibours as 2
3) Logistic Regression
4) Decision Trees
5) SVM
6) XG Boost

KNN with PCA:

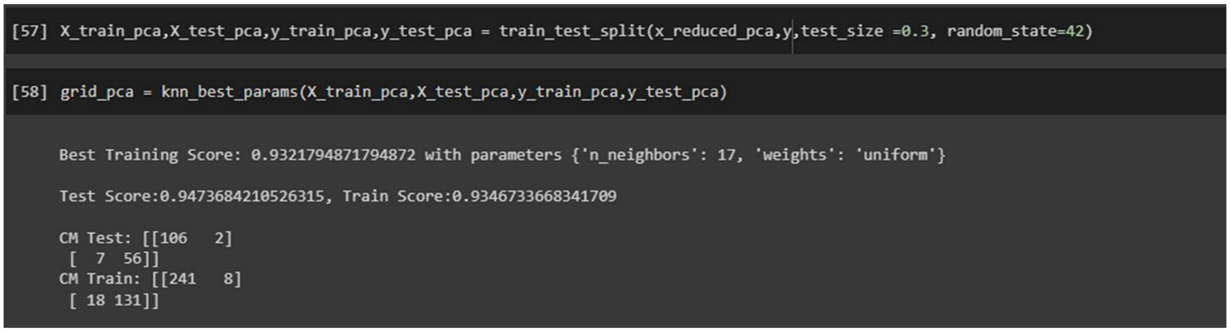
The k-nearest neighbor's algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

Principal Component Analysis or PCA is a widely used technique for the dimensionality reduction of a large data set. Reducing the number of components or features costs some accuracy and on the other hand, it makes the large data set simpler and easy to explore and visualize. Also, it reduces the computational complexity of the model which makes machine learning algorithms run faster.



```
[51] def knn_best_params(X_train,X_test,y_train,y_test):
         k_range = list(range(1,31))
         weight_options = ["uniform","distance"]
         print()
         #to grid search we need to add those values in a dict
         param_grid = dict(n_neighbors = k_range,weights = weight_options)
         knn = KNeighborsClassifier()
         grid = GridSearchCV(knn,param_grid, cv = 10, scoring = "accuracy")
         grid.fit(X_train,y_train)
         print("Best Training Score: {} with parameters {}".format(grid.best_score_, grid.best_params_))
         print()

         knn = KNeighborsClassifier(**grid.best_params_)
         knn.fit(X_train,y_train)

         y_pred_test = knn.predict(X_test)
         y_pred_train = knn.predict(X_train) #are there any overfitting or underfitting

         cm_test = confusion_matrix(y_test,y_pred_test)
         cm_train = confusion_matrix(y_train,y_pred_train)

         acc_test = accuracy_score(y_test,y_pred_test)
         acc_train = accuracy_score(y_train,y_pred_train)
         print("Test Score:{}, Train Score:{}".format(acc_test,acc_train))
         print()
         print("CM Test:", cm_test)
         print("CM Train:", cm_train)

         return grid
```

This is the sample implementation of the KNN algorithm using 2 neighbors because we are classifying data between whether the cancer is benign or malignant.



```
[57] X_train_pca,X_test_pca,y_train_pca,y_test_pca = train_test_split(x_reduced_pca,y,test_size =0.3, random_state=42)

[58] grid_pca = knn_best_params(X_train_pca,X_test_pca,y_train_pca,y_test_pca)

     Best Training Score: 0.9321794871794872 with parameters {'n_neighbors': 17, 'weights': 'uniform'}

     Test Score:0.9473684210526315, Train Score:0.9346733668341709

     CM Test: [[106   2]
      [  7  56]]
     CM Train: [[241   8]
      [ 18 131]]
```

By applying PCA with the KNN algorithm we got the

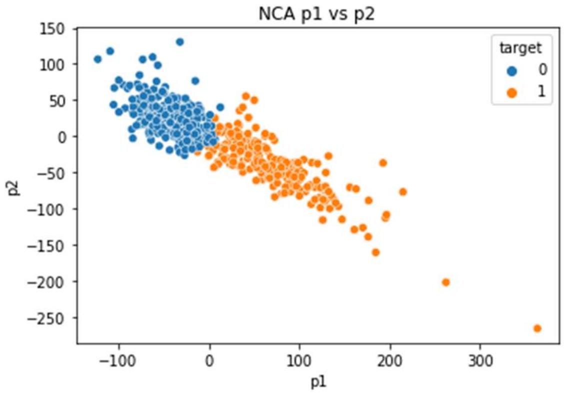
Accuracy: 0.9321794871794872

Test Scores: 0.947368421052631

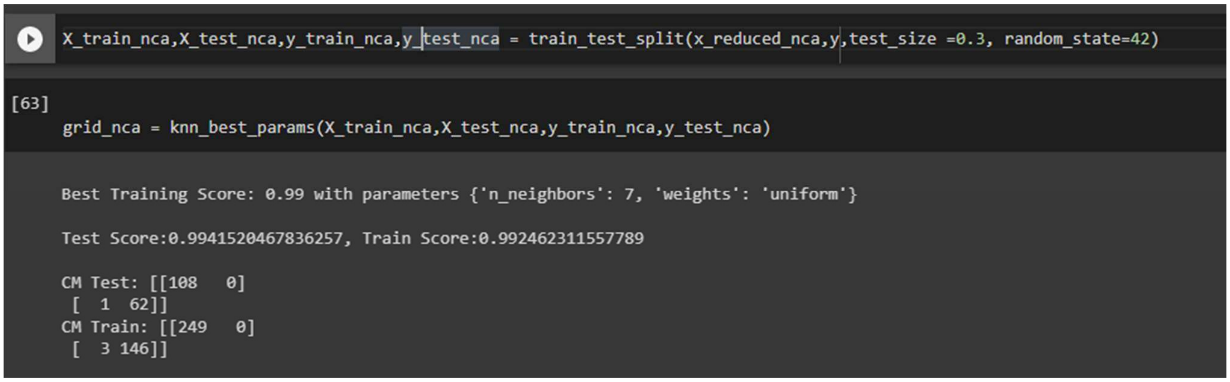
Train Score: 0.9346733668341709

KNN with NCA:

Neighborhood component analysis (NCA) is a non-parametric method for selecting features with the goal of maximizing the prediction accuracy of regression and classification algorithms. It learns linear transformation in a supervised fashion to improve the classification accuracy of a stochastic nearest neighbor’s rule in the transformed space.



As you can see above plt nca gives better solution than pca (dots are seperated).



```
X_train_nca,X_test_nca,y_train_nca,y_test_nca = train_test_split(x_reduced_nca,y,test_size =0.3, random_state=42)
```

```
[63]
     grid_nca = knn_best_params(X_train_nca,X_test_nca,y_train_nca,y_test_nca)

     Best Training Score: 0.99 with parameters {'n_neighbors': 7, 'weights': 'uniform'}

     Test Score:0.9941520467836257, Train Score:0.992462311557789

     CM Test: [[108   0]
      [  1  62]]
     CM Train: [[249   0]
      [  3 146]]
```

By Applying NCA with KNN we the

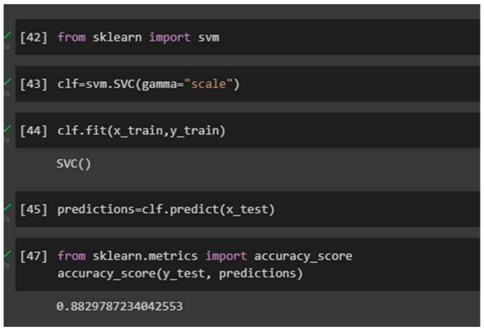
Accuracy: 0.99

Test Score: 0.99

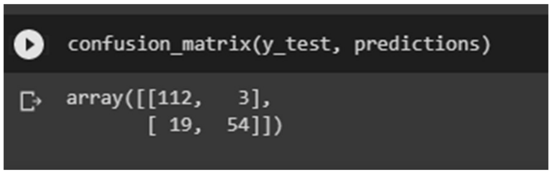
Train Accuracy: 0.99

SVM:

A Support Vector Machine (SVM) is a binary linear classification whose decision boundary is explicitly constructed to minimize generalization error. It is a very powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification, regression, and even outlier detection.
SVM is well suited for the classification of complex but small or medium-sized datasets.



```
[42] from sklearn import svm

[43] clf=svm.SVC(gamma="scale")

[44] clf.fit(x_train,y_train)

     SVC()

[45] predictions=clf.predict(x_test)

[47] from sklearn.metrics import accuracy_score
     accuracy_score(y_test, predictions)

     0.8829787234042553
```



```
confusion_matrix(y_test, predictions)

array([[112,   3],
       [ 19,  54]])
```

By implementing SVM we got

Accuracy: 0.8829787234042553

These are some of the advantages of SVM:

Effective in high-dimensional spaces.

Still effective in cases where the number of dimensions is greater than the number of samples.

Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

WORK COMPLETED:

As we proposed we decided to implement multiple machine learning algorithms we gathered the dataset from Kaggle and carried out some Exploratory data analysis on the dataset. After implementing EDA, we started implementing the KNN algorithm in which we implemented KNN with PCA (Principal component analysis) and KNN with NCA (Neighborhood component analysis). We have also implemented the SVM (Support vector machines) algorithm on the breast cancer dataset.

Responsibilities:

As a group we divided the tasks equally among us:

For each part of the project, we divided the work as follows:

- **Uday and Gayathri** have invested their time into referring to different papers and articles to assess and decide on the machine learning algorithms.
- **Laxma Reddy and Vinay** have handled the dataset collection, preprocessing, EDA, and implementation of algorithms and ML models.

Contribution:

- We four members worked together on this use case by discussing all the aspects that needs to be done.
- Overall, each person in the group contributed equally to the project i.e 25% of work per person.

WORK TO BE COMPLETED:

- At the end of part two of the project we left a few other Machine Learning Algorithms and the performance tuning part.

References

[1] Abien Fred Agarap. 2017. A Neural Network Architecture Combining Gated Recurrent Unit (GRU) and Support Vector Machine (SVM) for Intrusion Detection in Network Traffic Data. arXiv preprint arXiv:1709.03082 (2017).

[2] C. Cortes and V. Vapnik, (1995), “Support-vector Networks.Machine Learning 20.3”, (1995), 273– 297. https://doi.org/10.1007/BF00994018.

[3] Gönen, M.; Alpaydın, E. Multiple kernel learning algorithms. J. Mach. Learn. Res. 2011, 12, 2211– 2268.

[4] Keles, M. Kaya, "Breast Cancer Prediction and Detection Using Data Mining Classification Algorithms: A Comparative Study." Tehnicki Vjesnik - Technical Gazette, vol. 26, no. 1, 2019, p. 149+.

[5] Cristianini, N.; Shawe-Taylor, J. An Introduction to Support Vector Machines and other kernel- based learning methods.Ai Magazine 2000, 22, 190.

[6] Y. Sun, J. Liu, K. Yu, M. Alazab, K. Lin, “PMRSS: Privacy-preserving Medical Record Searching Scheme for Intelligent Diagnosis in IoT Healthcare”, IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2021.3070544