Machine Learning Applications in Healthcare: Case Studies and Analysis

The University of Texas at Austin

Department of Computer Science

Author:
Kevin Salazar

Supervisor:
Professor Junfeng Jiao

Case Studies in Machine Learning

November 30th, 2024

1. Introduction

As technology rises day to day, artificial intelligence and its subset of machine learning become a more-than-ever hot topic, because of this, it is essential to reflect on how this groundbreaking technology can be implemented to save lives. The focus of this study will be Machine Learning and its ability to learn from data, in the form of examples, definitions, behavior that automatically improves with experience [1]. By now, it is almost old news: big data will transform medicine [2], this is because of the enough information collected from all branches of medicine, and big data analytics can now be applied; originating new technologies, both in treatment of patients and health management [3]. Machine learning similarly approaches problems as a doctor progresses through residency: learning rules from data, making patient observations, considering variables and making a prediction in the form of a diagnosis [4].

The current machine learning advancements in healthcare have primarily served as support for a physician's ability to fulfill their roles, identify trends in healthcare, and develop disease prediction models [5]. Some of those specific machine learning accomplishments are: identification of irregularities in the blood samples [7], bones [10], organs [7-9], all of this done with the use of medical monitoring and imaging [6]. In addition, multiple studies have been performed and discuss AI and ML applications to respond to the global health crisis of COVID-19 [11, 12, 13 ,14, 15], which gave more insight on AI & ML methods based on their challenges experienced when implementing such a type of technology.

The primary purpose of this study is to identify and understand how machine learning can be applied to contribute to healthcare in society. This paper will focus on multiple features of machine learning in healthcare; such as:

- How beneficial can machine learning be during an extreme situation such as a pandemic?

- How functional machine learning models are when making a diagnosis?
- How effective is machine learning in predicting an outcome for a patient in a specific medical condition?
- Can more than one machine learning model be applied for a case study?

2. Research Background

As it is known, the Coronavirus disease 19 (COVID-19) affected humanity in almost every way possible, as a global health emergency, it disrupted the societal flow. Determining the next steps of a pandemic as it actively occurs is essential for humanity; thus, ever since, there has been a lot of research and study done to prepare for the future.

Some of the applied research includes Machine Learning methodologies to predict the immunogenic landscape of COVID-19 for universal vaccine design blueprints [20], where ensemble machine learning models were used; this study mapped the biology of the COVID-19 virus and the human body, using antigen presentation and immunogenicity predictors. The results identified different antigen presentations into distinct population groups. Therefore, to create the optimal vaccine, they needed to predict “epitope hotspots”, the binding of an antibody on a target antigen [20].

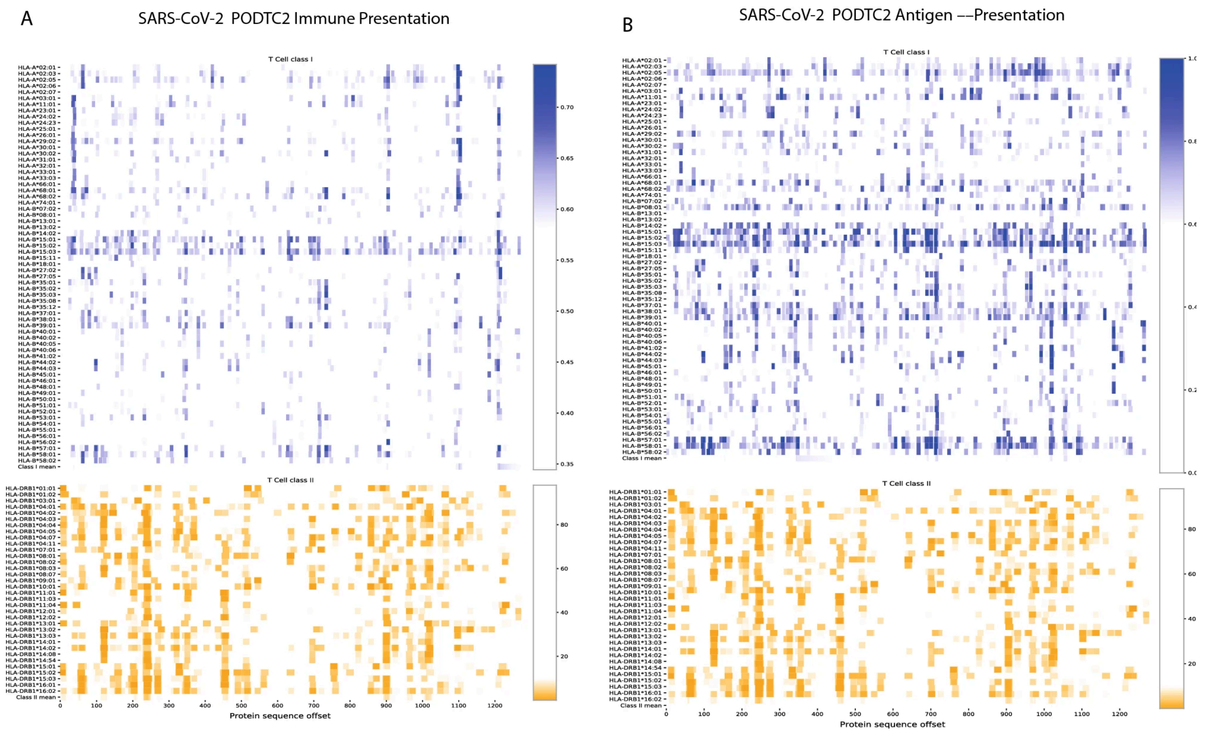


Figure 1. Mapping the S-Protein of the COVID-19 Virus [20]

The image represents the mapping of the S-Protein (spike protein of COVID-19 virus) across the most frequent alleles in the human population. Positive results are 0.7 and above and 0.1 and below for a Class II [20].

All of this thanks to an artificial intelligence engine for clinical trials: The NEC Immune Profiler, which involves ensemble models such as Random Forest. This is a prime example of how machine learning methodologies can revolutionize health sciences, the results offered a possible vaccine blueprint [20].

In addition, the use of machine learning in health sciences also affects the way we diagnose illnesses, another example comes from CheXNet: an algorithm that is capable of detecting pneumonia. As of now, the challenging evaluation of X-rays with the help of expert radiologists, is the best way to diagnose pneumonia, however, CheXNet can innovate. This image analyzing algorithm is a DenseNet, which is a variant of a convolutional neural network [21]. The architecture of this algorithm contains: 121 CNN layers, an optimized loss function, the weights are initialized with pre-trained model weights, with an initial learning rate of 0.001, decayed by a factor of 10, each time that the validation loss plateaus after an epoch. [21] .

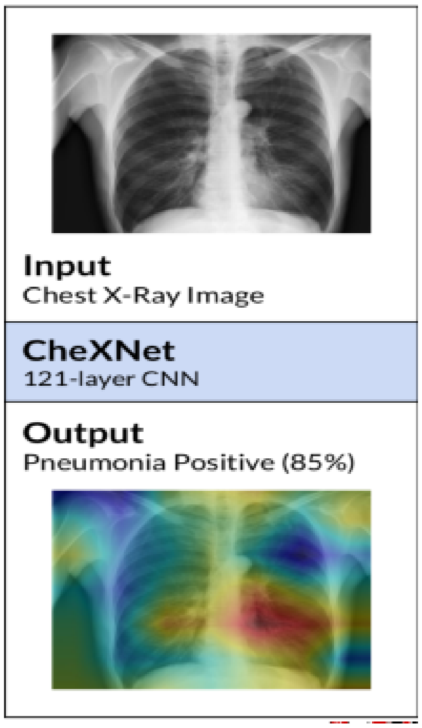
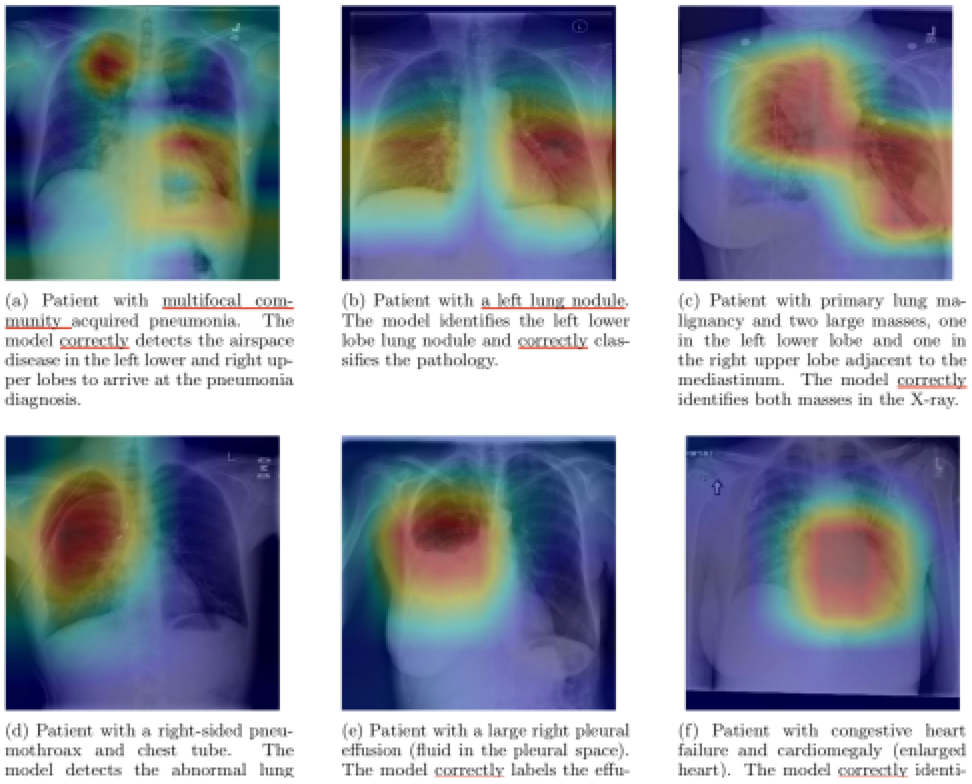


Figure 2. Visual overview of CheXNet [21]

As shown in the image above, the input of the algorithm is a X-Ray image, the output portrays a heatmap over the X-Ray image, representing the possible probability of pneumonia. The comparison of CheXNet to an average Radiologist was the following: F1 scores were 0.387, and 0.435, CheXNet is statistically higher. [21].



(a) Patient with multifocal community acquired pneumonia. The model correctly detects the airspace disease in the left lower and right upper lobes to arrive at the pneumonia diagnosis.

(b) Patient with a left lung nodule. The model identifies the left lower lobe lung nodule and correctly classifies the pathology.

(c) Patient with primary lung malignancy and two large masses, one in the left lower lobe and one in the right upper lobe adjacent to the mediastinum. The model correctly identifies both masses in the X-ray.

(d) Patient with a right-sided pneumothroax and chest tube. The model detects the abnormal lung

(e) Patient with a large right pleural effusion (fluid in the pleural space). The model correctly labels the effu-

(f) Patient with congestive heart failure and cardiomegaly (enlarged heart). The model correctly identi-

Figure 3. CheXNet application on multiple patients [21]

The image above represents the X-Rays of multiple patients after applying CheXNet, helping highlight areas of importance for further investigation, when making a classification for an X-Ray related disease [21].

With the two real life examples provided, we can see that some key applications of machine learning in health sciences are predictive analytics (the blueprint for a possible vaccine) and disease diagnosis (x-Ray algorithm).

It should not be forgotten that although these examples have been applied in real life scenarios, they still have limitations and are actively improving with day to day research. The CheXNet algorithm can only evaluate frontal X-ray images for training and testing, is limited to a certain number of doctors when labeling images, more doctors could be implemented, etc [21].

In the continuation of this work, we will be performing extended analysis on an existent data set dedicated to a specific medical condition.

3. Dataset Background

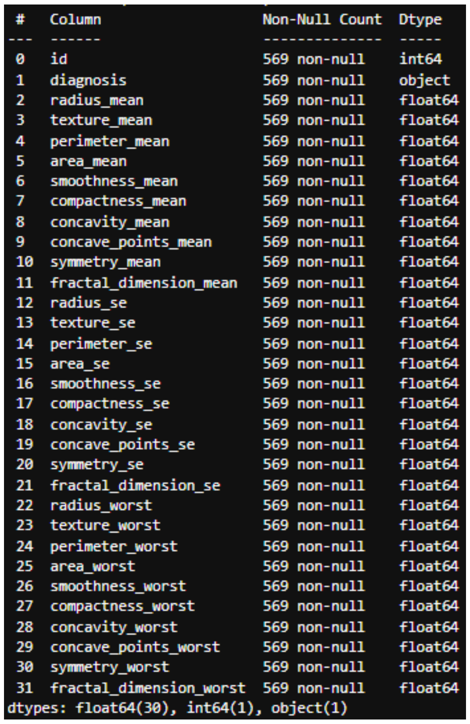
According to the CDC (Centers for Disease Control and Prevention), cancer is the second leading cause of death in the United States [17] . The most common cancer recorded is breast cancer, followed by prostate, lung and colorectal cancer [18, 19]. With this being mentioned, our dataset will contain diagnostic data related to breast cancer. This will demonstrate how machine learning can contribute to predicting a specific outcome for a patient in a specific condition and get more insight on the variety of machine learning models that can be applied to a single case.

This dataset first appeared in 1992, it was created by Dr. William H. Wolberg of the University of Wisconsin[23] . The dataset was initially compiled at the University of Wisconsin hospitals, where the working team used fine needle aspiration (FNA) biopsies to extract breast tumor cells for examination [22].

4. Data Attributes

To investigate this dataset, this work will be conducting a meta-analysis of some existing studies [24, 26], with some additions added, on the sole dataset previously mentioned. The execution of the studies will be done on Jupyter Notebooks.

The dataset contains 30 numerical features, these features describe multiple properties of the cell nuclei present in a breast tumor. The following image shows the data information with ‘data.info()’.



```
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave_points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave_points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave_points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30), int64(1), object(1)
```

Figure 4. Utilizing ‘.info()’ method on the dataset.

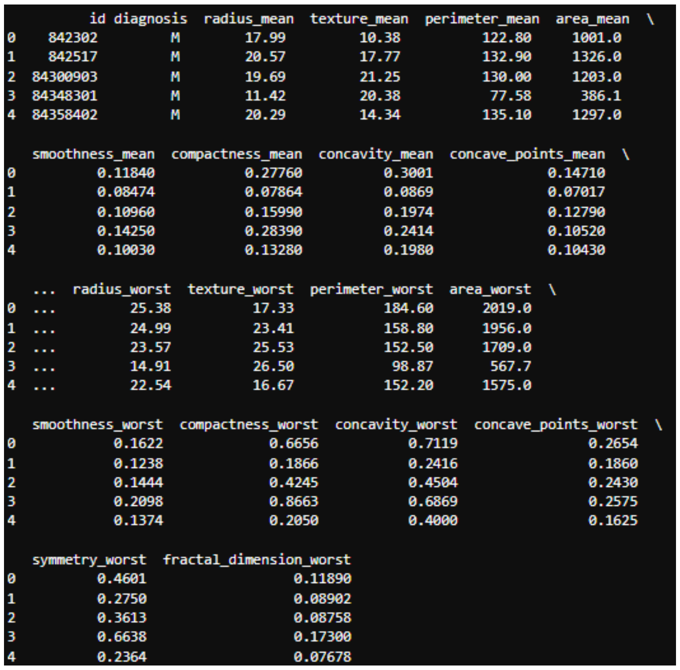
Here are some key features from the cell nuclei in a breast tumor to consider [23]:

- Radius: Distance from center to points on the nucleus perimeter.
- Texture: Standard deviation for grayscale values
- Perimeter: Distance of a shape
- Area
- Smoothness: The local variation in radius lengths
- Compactness: ((Squared perimeter / Area) - 1)
- Concavity: Intensity of the contour that can be concave

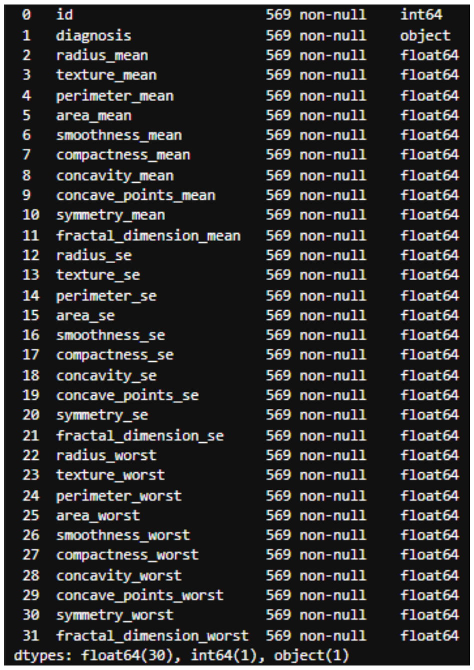
- Concave Points: Amount of concave portions
- Symmetry
- Fractal Dimension: (‘coastline approximation’ - 1)

5. Data Analysis, Preparation, Visualization & Problem Formulation

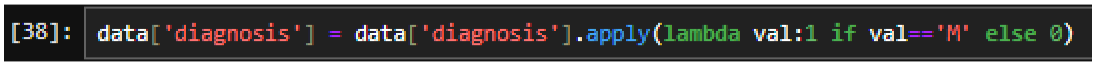
After applying several data analysis methods such as: ‘data.info’, ‘data.head()’, ‘data.isnull().sum()’. The analysis of the data is as follows:



```
         id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0    842302         M        17.99         10.38          122.80     1001.0
1    842517         M        20.57         17.77          132.90     1326.0
2  84300903         M        19.69         21.25          130.00     1203.0
3  84348301         M        11.42         20.38           77.58      386.1
4  84358402         M        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave_points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   ...  radius_worst  texture_worst  perimeter_worst  area_worst  \
0  ...         25.38          17.33           184.60      2019.0
1  ...         24.99          23.41           158.80      1956.0
2  ...         23.57          25.53           152.50      1709.0
3  ...         14.91          26.50            98.87       567.7
4  ...         22.54          16.67           152.20      1575.0

   smoothness_worst  compactness_worst  concavity_worst  concave_points_worst  \
0            0.1622             0.6656           0.7119                0.2654
1            0.1238             0.1866           0.2416                0.1860
2            0.1444             0.4245           0.4504                0.2430
3            0.2098             0.8663           0.6869                0.2575
4            0.1374             0.2050           0.4000                0.1625

   symmetry_worst  fractal_dimension_worst
0          0.4601                  0.11890
1          0.2750                  0.08902
2          0.3613                  0.08758
3          0.6638                  0.17300
4          0.2364                  0.07678
```



```
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave_points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave_points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave_points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30), int64(1), object(1)
```

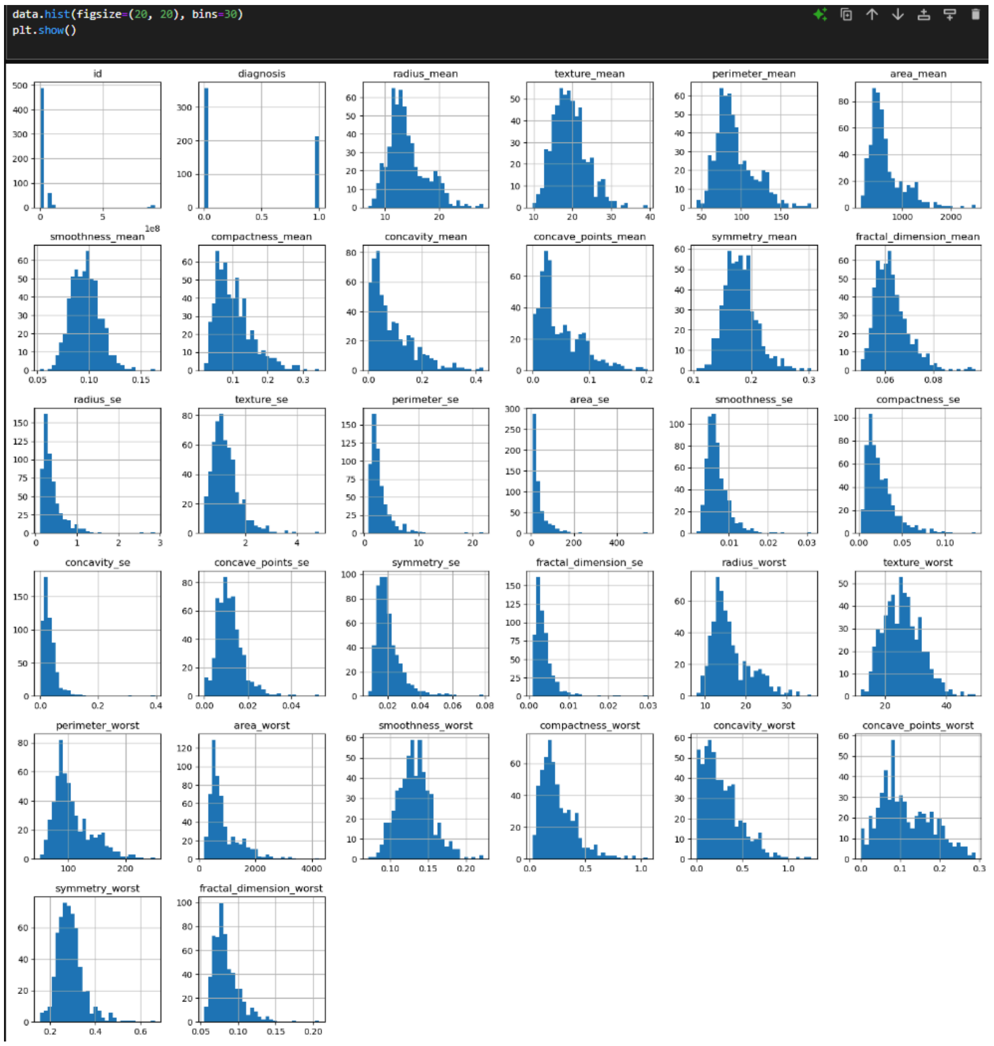


```
[38]: data['diagnosis'] = data['diagnosis'].apply(lambda val:1 if val=='M' else 0)
```

Figures 5, 6 and 7. VIsuals of the data analysis and modifying the object type feature to a binary classification.

The dataset is composed of 569 rows and 32 columns, there are no missing values, and we have an 'object' data type for the 'diagnosis' feature whose values are M (Malignant) or B (Benign). The goal is to classify according to the diagnose feature. Therefore the values M and B will be encoded for binary classification, M will equal 1 and B will equal 0.

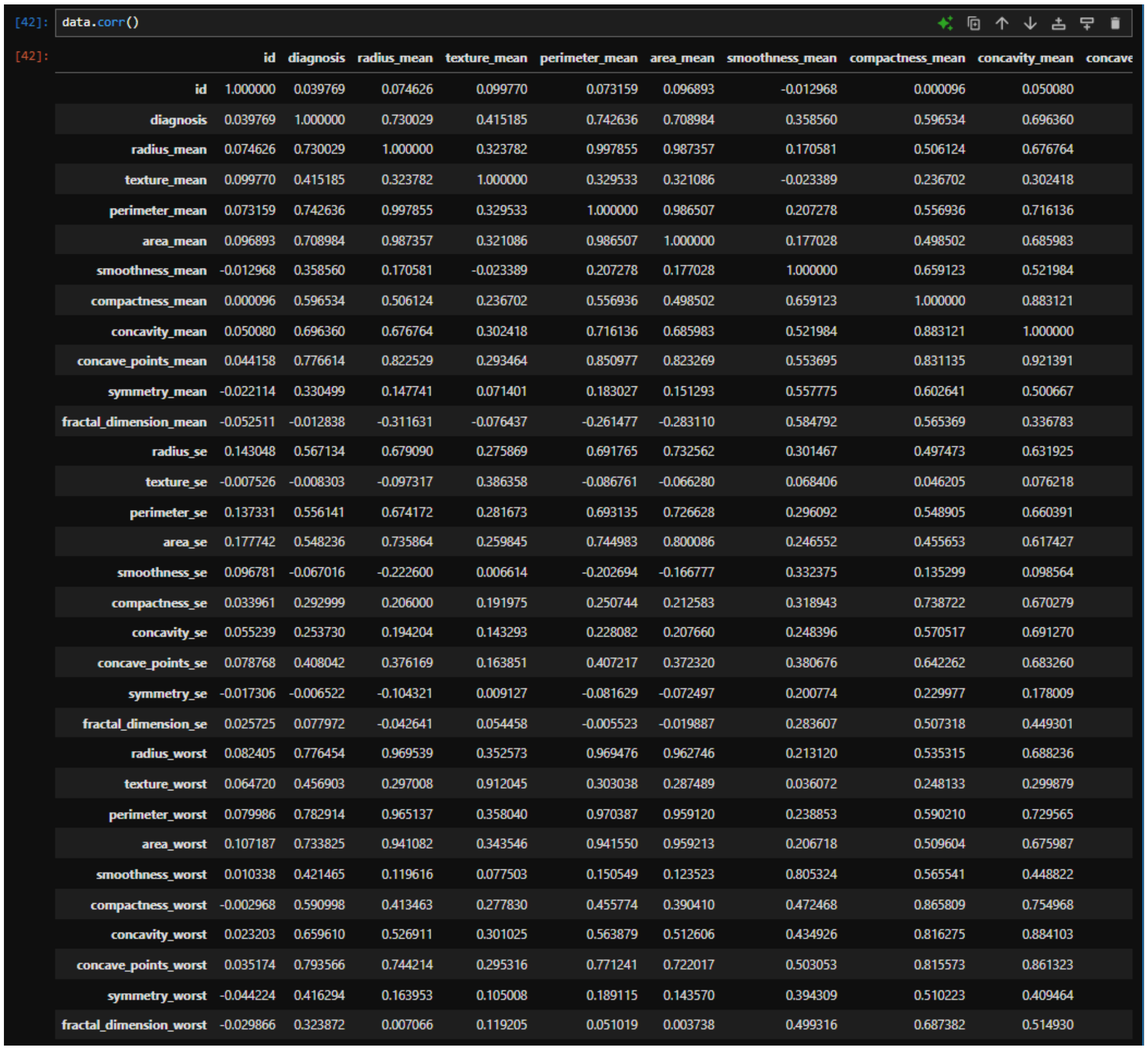
6. Exploratory Data Analysis, Data Correlation



```
data.hist(figsize=(20, 20), bins=30)
plt.show()
```

Figure 8. The image above represents the distribution of values for each specific feature on the dataset.

From a glance, it can be seen that the majority of the features have a right-skewed distribution. The diagnosis feature was transformed from an object data type to binary values, majority of the data recorded is Benign or 0. There are some features such as ‘texture_mean’ and ‘smoothness_mean’ that have a normal looking distribution. It can be seen that some features like ‘radius_mean’, ‘radius_worst’ and ‘radius_se’ have similar distributions because they are metrics of the same property; a further analysis for correlated data will be done.



```
[42]: data.corr()
```

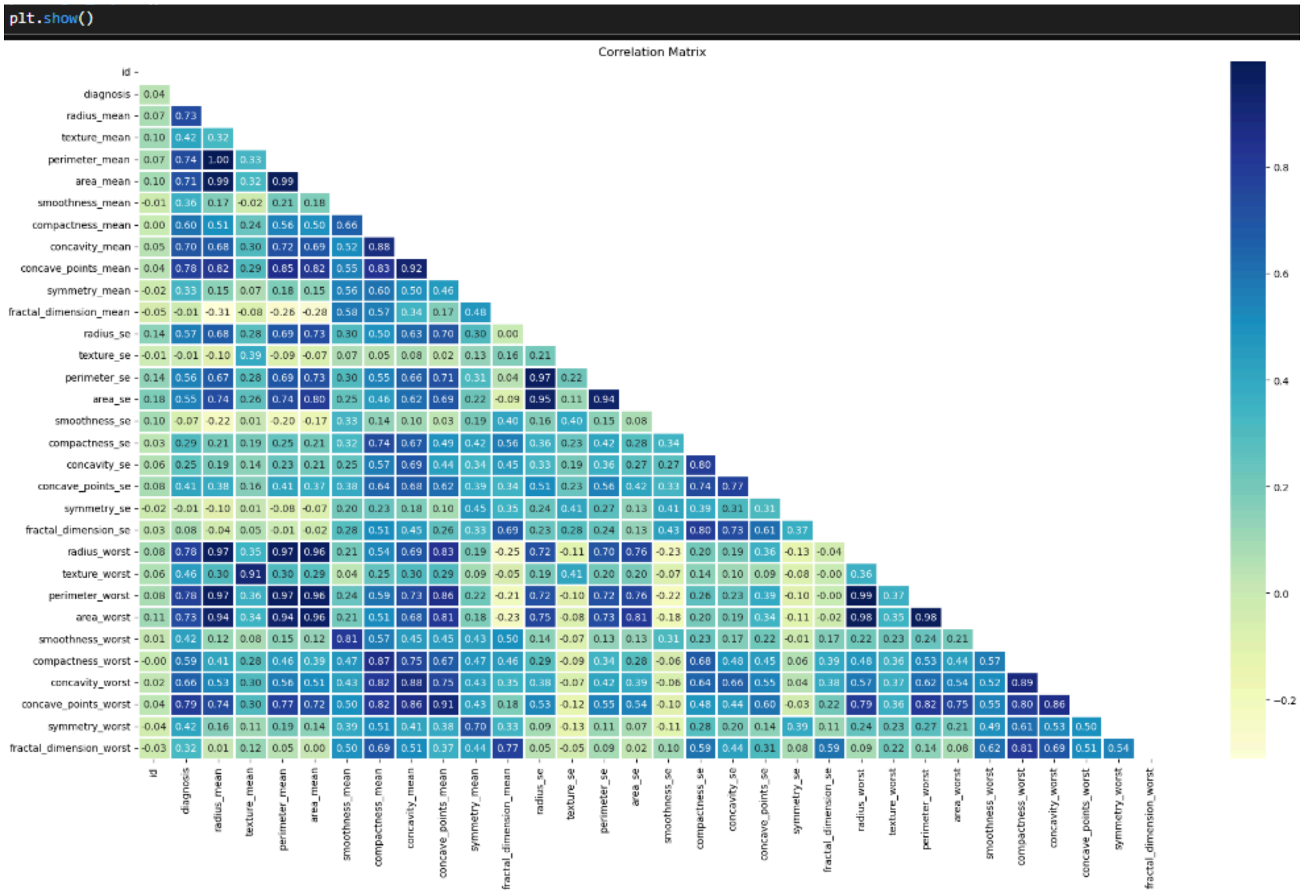
[42]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave
id	1.000000	0.039769	0.074626	0.099770	0.073159	0.096893	-0.012968	0.000096	0.050080	
diagnosis	0.039769	1.000000	0.730029	0.415185	0.742636	0.708984	0.358560	0.596534	0.696360	
radius_mean	0.074626	0.730029	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764	
texture_mean	0.099770	0.415185	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	0.302418	
perimeter_mean	0.073159	0.742636	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716136	
area_mean	0.096893	0.708984	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	0.685983	
smoothness_mean	-0.012968	0.358560	0.170581	-0.023389	0.207278	0.177028	1.000000	0.659123	0.521984	
compactness_mean	0.000096	0.596534	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	0.883121	
concavity_mean	0.050080	0.696360	0.676764	0.302418	0.716136	0.685983	0.521984	0.883121	1.000000	
concave_points_mean	0.044158	0.776614	0.822529	0.293464	0.850977	0.823269	0.553695	0.831135	0.921391	
symmetry_mean	-0.022114	0.330499	0.147741	0.071401	0.183027	0.151293	0.557775	0.602641	0.500667	
fractal_dimension_mean	-0.052511	-0.012838	-0.311631	-0.076437	-0.261477	-0.283110	0.584792	0.565369	0.336783	
radius_se	0.143048	0.567134	0.679090	0.275869	0.691765	0.732562	0.301467	0.497473	0.631925	
texture_se	-0.007526	-0.008303	-0.097317	0.386358	-0.086761	-0.066280	0.068406	0.046205	0.076218	
perimeter_se	0.137331	0.556141	0.674172	0.281673	0.693135	0.726628	0.296092	0.548905	0.660391	
area_se	0.177742	0.548236	0.735864	0.259845	0.744983	0.800086	0.246552	0.455653	0.617427	
smoothness_se	0.096781	-0.067016	-0.222600	0.006614	-0.202694	-0.166777	0.332375	0.135299	0.098564	
compactness_se	0.033961	0.292999	0.206000	0.191975	0.250744	0.212583	0.318943	0.738722	0.670279	
concavity_se	0.055239	0.253730	0.194204	0.143293	0.228082	0.207660	0.248396	0.570517	0.691270	
concave_points_se	0.078768	0.408042	0.376169	0.163851	0.407217	0.372320	0.380676	0.642262	0.683260	
symmetry_se	-0.017306	-0.006522	-0.104321	0.009127	-0.081629	-0.072497	0.200774	0.229977	0.178009	
fractal_dimension_se	0.025725	0.077972	-0.042641	0.054458	-0.005523	-0.019887	0.283607	0.507318	0.449301	
radius_worst	0.082405	0.776454	0.969539	0.352573	0.969476	0.962746	0.213120	0.535315	0.688236	
texture_worst	0.064720	0.456903	0.297008	0.912045	0.303038	0.287489	0.036072	0.248133	0.299879	
perimeter_worst	0.079986	0.782914	0.965137	0.358040	0.970387	0.959120	0.238853	0.590210	0.729565	
area_worst	0.107187	0.733825	0.941082	0.343546	0.941550	0.959213	0.206718	0.509604	0.675987	
smoothness_worst	0.010338	0.421465	0.119616	0.077503	0.150549	0.123523	0.805324	0.565541	0.448822	
compactness_worst	-0.002968	0.590998	0.413463	0.277830	0.455774	0.390410	0.472468	0.865809	0.754968	
concavity_worst	0.023203	0.659610	0.526911	0.301025	0.563879	0.512606	0.434926	0.816275	0.884103	
concave_points_worst	0.035174	0.793566	0.744214	0.295316	0.771241	0.722017	0.503053	0.815573	0.861323	
symmetry_worst	-0.044224	0.416294	0.163953	0.105008	0.189115	0.143570	0.394309	0.510223	0.409464	
fractal_dimension_worst	-0.029866	0.323872	0.007066	0.119205	0.051019	0.003738	0.499316	0.687382	0.514930	

Figure 9. Usage of corr() function on the dataset.

After running ‘data.corr()’ the matrix displays the Pearson correlation coefficient for each pair of variables. As it can be seen there are some values that are very correlated to each other, one example is ‘area_mean’ and ‘radius_mean’, with a correlation of ‘0.987357’. Some other values such as radius and perimeter also have high correlation.

A better representation of the correlation will be done with a heat map of the correlation matrix [24] :



```
plt.show()
```

Figure 10. Generating a Correlation Matrix heatmap [24].

The darker values on the heatmap demonstrate high correlation between two features, at this glance, we can see that values range from -1 to 1. Features like ‘radius_mean’, ‘area_mean’, and ‘concave_points_mean’ appear to have high correlations with each other.. There are some features related with the ‘diagnosis’ feature, such as ‘concave_points_mean’ and ‘area_worst’, from this glance, these features can be highly predictive on the malignant or benign classification. Overall, this map helps identify the feature importance to the target. Correlation matrix is very useful for regression techniques such as linear regression, multiple linear regression and lasso regression, which is ideal for this study [25].

7. Methodology

For the methodology, we will be implementing multiple machine learning models, and debate which one is the best for the case. Parts of the analysis will be based on existing studies by Vika Sukani and possibly other cited works, in order to answer our research questions [26].

8. Machine Learning Models

In this study, a numerous amount of machine learning models will be used, a list of them as follows:

- Logistic Regression: The reason why logistic regression will be chosen as a model for the dataset is because, the model has its basis in the odds of a two level outcome, and in this dataset, it is possible to have an assumption of a linear relationship between the features and the natural logarithm of the odds of the target variable (Malignant/Benign). This model also tends to perform well on a dataset with continuous and correlated features (we previously showed the correlation matrix for the features in the dataset) [27].

- Decision Tree Classifier: This model is a perfect candidate for the study of this dataset because the problem revolves in a classification (benign or malign). There are two main operations for the decision tree: Evaluation of splits for each attribute with the selection of the best split, and the creation of partitions using the best split [28]. The analysis of the dataset involves looking at the features and the correlation between them, decision trees demonstrate a clear ranking of feature importance, which can help with the ultimate classification goal. [28]

- Random Forest Classifier: This model consists of a set of decision trees, each of which is generated by the bagging algorithm with no pruning, this forms a "forest" of classifiers voting for a particular class[29]. The basis of its ranking goes by feature importance (as previously mentioned for the decision tree) which is convenient with datasets that contain a lot of features, such as the one presented in this study[29]. The uniqueness about Random Forests is that they tend to generalize better than decision trees due to the high randomization involved when selecting subsets of data and features. [29]

- K Neighbor Classifier: This model classifies an unlabelled test sample based on the majority of similar samples among the k-nearest neighbors that are the closest to the test sample[30]. It is a baseline classifier in multiple pattern classification problems such as pattern recognition, text categorization, ranking models, object recognition, and event recognition[30]. As a non-parametric algorithm, because the parameters are determined by the size of the training dataset. There are no assumptions about the underlying data distribution[30]. Thus, KNN classifier could be the best choice for any classification study that involves little to no prior knowledge about data distribution. Making it a great fit for this study, which is a classification problem [30].

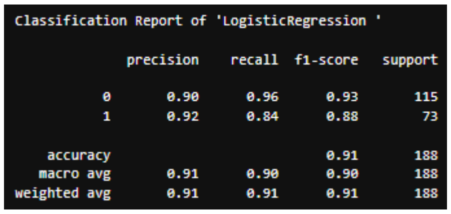
- Support Vector Classifier: This model offers classification for both linear and non linear data, they are a set of related supervised learning methods used for classification and regression[31]. The reason why this model is a great fit for the presented data set is that, with the amount of features that the dataset has, SVC can clear the class separation of malignant and benign cases. SVC also seeks an optimal hyperplane, which can maximize the margin between benign and malignant classes [31].

9. Further Analysis

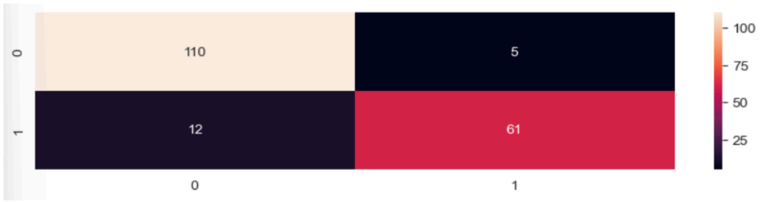
There will be a further analysis with the created models, using the grid search algorithm, the reason for this is to apply some hyper tuning, some of the models (such as SVM or Random Forest) can have their accuracy increased with the use of grid search. Grid search offers manual optimization, through the predefined set of hyperparameter values; thus, ideal for this case study. [32]

10. Results and Findings

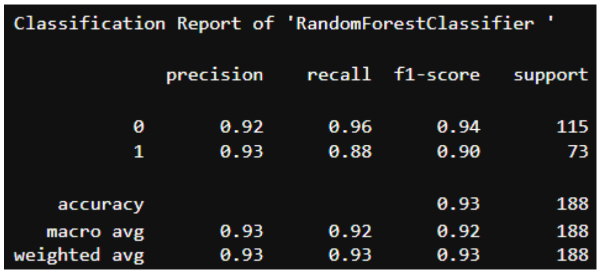
After training and creating the previously mentioned models, classification reports for each of the models were created, along with its confusion matrices. As presented in the following figures:

I 

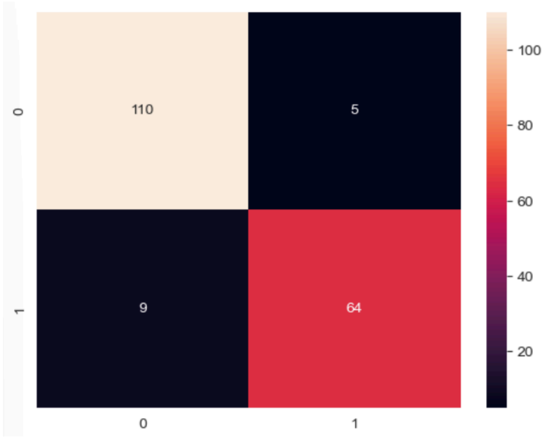
```
Classification Report of 'LogisticRegression '

              precision    recall  f1-score   support

           0       0.90      0.96      0.93       115
           1       0.92      0.84      0.88        73

    accuracy                           0.91       188
   macro avg       0.91      0.90      0.90       188
weighted avg       0.91      0.91      0.91       188
```



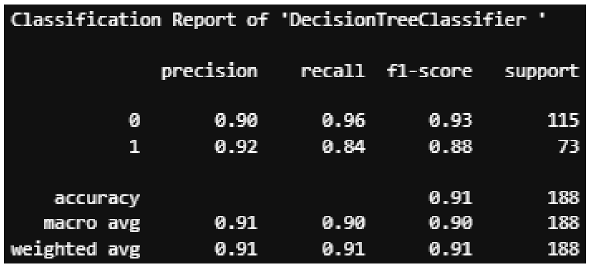
Figures 11 and 12. Classification report and Confusion Matrix for LogisticRegression.



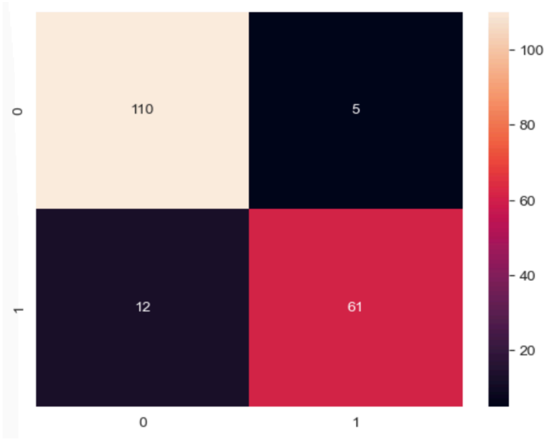
```
Classification Report of 'RandomForestClassifier '

              precision    recall  f1-score   support

           0       0.92      0.96      0.94       115
           1       0.93      0.88      0.90        73

    accuracy                           0.93       188
   macro avg       0.93      0.92      0.92       188
weighted avg       0.93      0.93      0.93       188
```



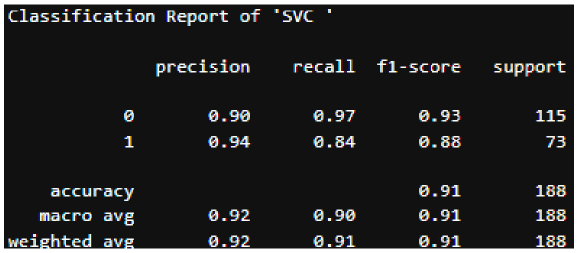
Figures 13 and 14. Classification report and Confusion Matrix for Random Forest Classifier



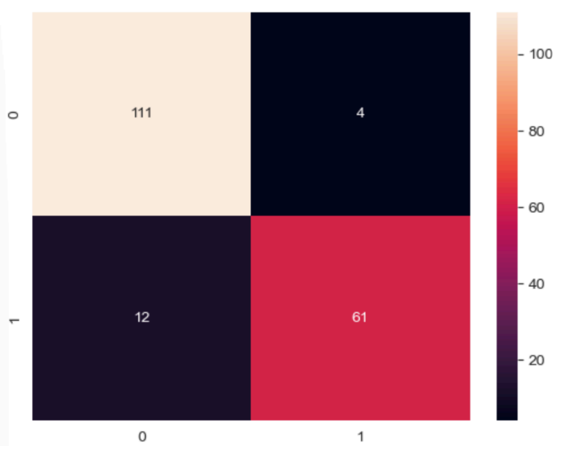
```
Classification Report of 'DecisionTreeClassifier '

              precision    recall  f1-score   support

           0       0.90      0.96      0.93       115
           1       0.92      0.84      0.88        73

    accuracy                           0.91       188
   macro avg       0.91      0.90      0.90       188
weighted avg       0.91      0.91      0.91       188
```



Figures 15 and 16. Classification report and Confusion Matrix for Decision Tree Classifier



```
Classification Report of 'SVC '

              precision    recall  f1-score   support

           0       0.90      0.97      0.93       115
           1       0.94      0.84      0.88        73

    accuracy                           0.91       188
   macro avg       0.92      0.90      0.91       188
weighted avg       0.92      0.91      0.91       188
```



Figures 17 and 18. Classification report and Confusion Matrix for SVC

11. Ranking of Models

Logistic Regression: The images 11 and 12 represent the results for the Logistic Regression model, they are announced by specific classification (Benign = 0, or Malign = 1). For Benign, we have a 0.90 Precision, 0.96 Recall, 0.93 F1 Score for 115 instances. For Malign, we have 0.92 Precision, 0.84 Recall, 0.88 F1 Score for 73 instances. With an overall 0.91 Accuracy score.

The confusion matrix represents the following: 61 True Positives, 110 True Negatives, 5 False Positives and 9 False Negatives. The Malign-Benign instances are the same amount for each model result.

Random Forest Classifier: Following images 13 and 14 display the results for the Random Forest Classifier model. For Benign, we have a 0.92 Precision, 0.96 Recall, 0.94 F1 Score. For Malign, we have 0.93 Precision, 0.88 Recall, 0.90 F1 Score. With an overall 0.91 Accuracy score.

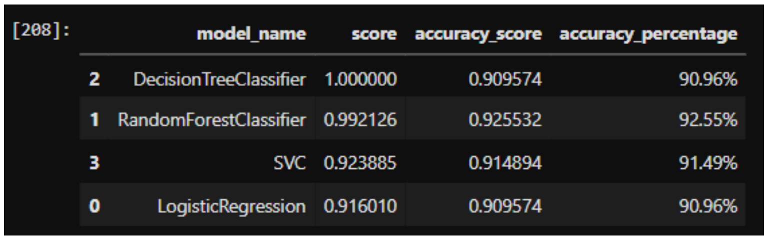
The confusion matrix displays 61 True Positives, 110 True Negatives, 5 False Positives and 12 False Negatives.

Decision Tree Classifier: The images 15 and 16 display the results for the Decision Tree Classifier. For Benign, we have a 0.90 Precision, 0.96 Recall, 0.93 F1 Score. For Malign, we have 0.92 Precision, 0.84 Recall, 0.88 F1 Score. With an overall 0.91 Accuracy score.

The confusion matrix represents 61 True Positives, 110 True Negatives, 5 False Positives and 12 False Negatives.

Support Vector Classifier: The images 17 and 18 display the results for the SVC model. For Benign, we have a 0.90 Precision, 0.97 Recall, 0.93 F1 Score. For Malign, we have 0.94 Precision, 0.84 Recall, 0.88 F1 Score. With an overall 0.91 Accuracy score.

The confusion matrix represents 61 True Positives, 110 True Negatives, 5 False Positives and 12 False Negatives.



[208]:

	model_name	score	accuracy_score	accuracy_percentage
2	DecisionTreeClassifier	1.000000	0.909574	90.96%
1	RandomForestClassifier	0.992126	0.925532	92.55%
3	SVC	0.923885	0.914894	91.49%
0	LogisticRegression	0.916010	0.909574	90.96%

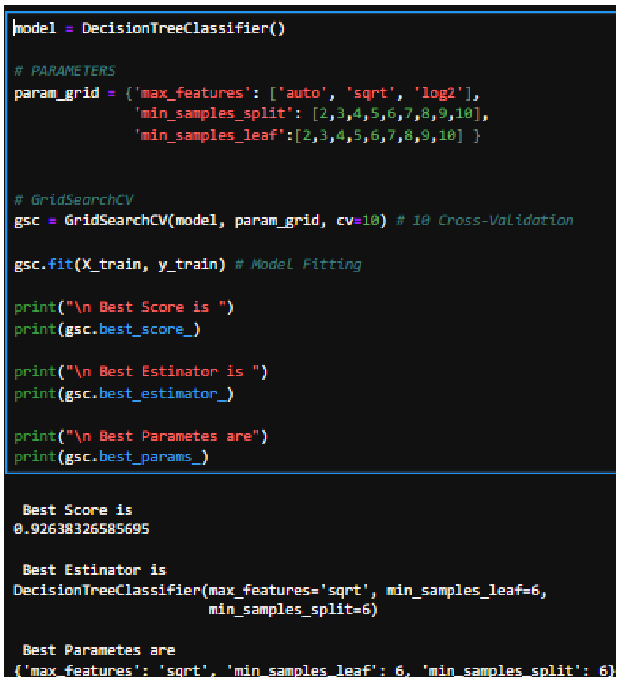
Figure 19. Results of the trained models.

Image 19 showcases the **overall scores and accuracy scores** for the 4 models. **The Decision Tree Classifier** got a score of 1, it perfectly fit the data during cross-validation. Test Accuracy score is 90.96%, it fit the data perfectly but it did not generalize well enough, it could be an overfitting issue. **Random Forest Classifier** performed somewhat worse with a score 0.992 but it has the highest accuracy score, which is 92.55%. **Support Vector Classifier** has a score of 0.92, and has a better testing percentage than the decision tree classifier, at 91.48%. At the final place of the ranking, **Logistic Regression** comes in with a 0.9160 score, the lowest of all four models, but it has a similar testing percentage to the **Decision Tree Classifier**, both at 90.96%, still a strong match for this case study.

12. Hyperparameter Tuning (Grid Search Algorithm)

By using ‘GridSearchCV’ we will hypertune the 4 presented models in this case study [24]. Each one of them will showcase the best score, best estimator and best parameters applied.

GSC DECISION TREE



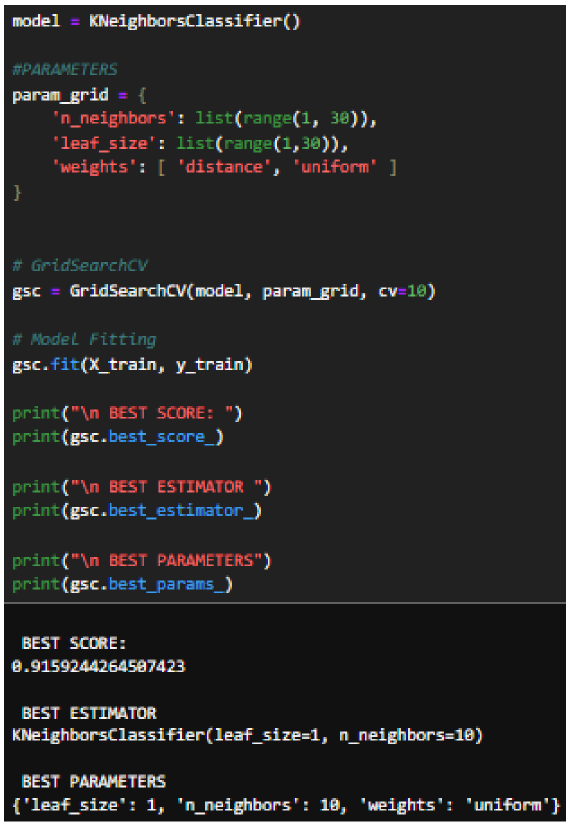
```
model = DecisionTreeClassifier()

# PARAMETERS
param_grid = {'max_features': ['auto', 'sqrt', 'log2'],
              'min_samples_split': [2,3,4,5,6,7,8,9,10],
              'min_samples_leaf':[2,3,4,5,6,7,8,9,10] }


# GridSearchCV
gsc = GridSearchCV(model, param_grid, cv=10) # 10 Cross-Validation

gsc.fit(X_train, y_train) # Model Fitting

print("\n Best Score is ")
print(gsc.best_score_)

print("\n Best Estinator is ")
print(gsc.best_estimator_)

print("\n Best Parametes are")
print(gsc.best_params_)
```

```
 Best Score is 
0.92638326585695

 Best Estinator is 
DecisionTreeClassifier(max_features='sqrt', min_samples_leaf=6,
                       min_samples_split=6)

 Best Parametes are
{'max_features': 'sqrt', 'min_samples_leaf': 6, 'min_samples_split': 6}
```

Figure 20. Grid Search Hypertuning on DecisionTreeClassifier

The hyperparameter tuning involved for the DecisionTreeClassifier involves controlling the maximum number of features considered when looking for the best split; they also include the minimum number of samples required to split an internal node and minimum samples required to be at a leaf node. Multiple optional parameters were applied in the ‘param_grid’ dictionary to choose the best hypertuning possible. The results gave a best score of 92.63%, the best estimators and parameters for the model were: Using the square root of the total number of features for the number of features for the best split, each leaf node must have at least 6 samples and the node will be best split is it contains a minimum of 6 samples.

GSC K NEIGHBOR CLASSIFIER



```
model = KNeighborsClassifier()

#PARAMETERS
param_grid = {
    'n_neighbors': list(range(1, 30)),
    'leaf_size': list(range(1,30)),
    'weights': [ 'distance', 'uniform' ]
}


# GridSearchCV
gsc = GridSearchCV(model, param_grid, cv=10)

# Model Fitting
gsc.fit(X_train, y_train)

print("\n BEST SCORE: ")
print(gsc.best_score_)

print("\n BEST ESTIMATOR ")
print(gsc.best_estimator_)

print("\n BEST PARAMETERS")
print(gsc.best_params_)
```

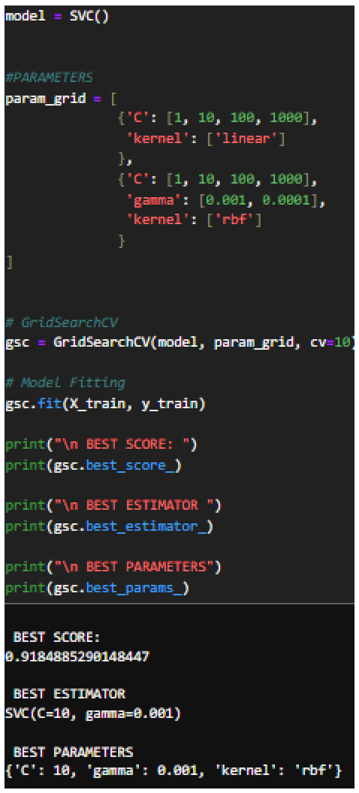
```
 BEST SCORE:
0.9159244264507423

 BEST ESTIMATOR
KNeighborsClassifier(leaf_size=1, n_neighbors=10)

 BEST PARAMETERS
{'leaf_size': 1, 'n_neighbors': 10, 'weights': 'uniform'}
```

Figure 21. Grid Search Hypertuning with K Neighbor Model

The hyperparameter tuning for the K Neighbor model involved the number of neighbors considered for classification, with a range of 1 to 30; the size of the leaf in the tree used for searching also with a range of 1 to 30; the use of the weights was used in two ways, distance based (closed neighbors have better influence) and uniform based (all neighbors have equal influence).

The model achieved a score of 91.59%, the best estimators and parameters for the model were: Having a leaf size of 1, having 10 numbers of neighbors and having a uniform based weighting.

GSC Support Vector Machine



```
model = SVC()


#PARAMETERS
param_grid = [
              {'C': [1, 10, 100, 1000],
               'kernel': ['linear']
              },
              {'C': [1, 10, 100, 1000],
               'gamma': [0.001, 0.0001],
               'kernel': ['rbf']
              }
]


# GridSearchCV
gsc = GridSearchCV(model, param_grid, cv=10)

# Model Fitting
gsc.fit(X_train, y_train)

print("\n BEST SCORE: ")
print(gsc.best_score_)

print("\n BEST ESTIMATOR ")
print(gsc.best_estimator_)

print("\n BEST PARAMETERS")
print(gsc.best_params_)
```

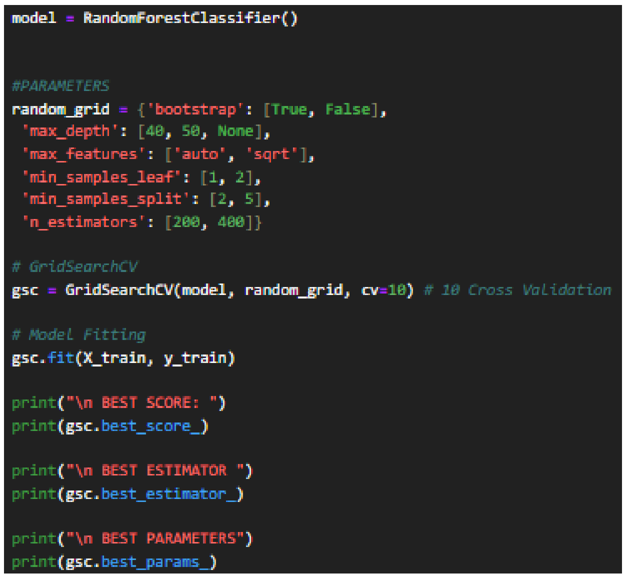
```
 BEST SCORE:
0.9184885290148447

 BEST ESTIMATOR
SVC(C=10, gamma=0.001)

 BEST PARAMETERS
{'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}
```

Figure 22. Grid Search Hypertuning with SVC Model

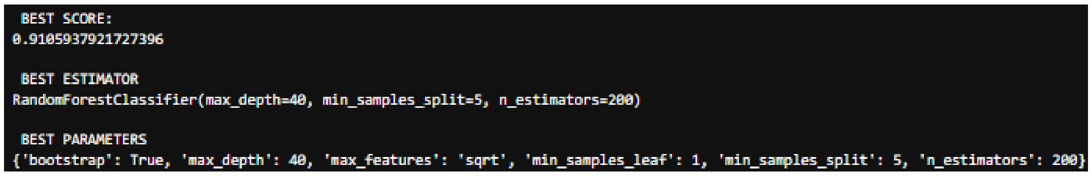
Hypertuning for the Support Vector Machine involved a regularization parameter (Value of 'C' for less regularization) with values 1, 10, 100 and 1000 tested, utilizing different types of kernel such as 'linear' or 'rbf'. The gamma parameter is for how far the influence of a single training can reach; the values tested were 0.001 and 0.0001.

The model achieved a score of 91.84%, having a C value of 10, a gamma value of 0.001 and a 'rbf' type of kernel were the best estimators and parameters.

GSC RANDOM FOREST CLASSIFIER



```
model = RandomForestClassifier()


#PARAMETERS
random_grid = {'bootstrap': [True, False],
 'max_depth': [40, 50, None],
 'max_features': ['auto', 'sqrt'],
 'min_samples_leaf': [1, 2],
 'min_samples_split': [2, 5],
 'n_estimators': [200, 400]}

# GridSearchCV
gsc = GridSearchCV(model, random_grid, cv=10) # 10 Cross Validation

# Model Fitting
gsc.fit(X_train, y_train)

print("\n BEST SCORE: ")
print(gsc.best_score_)

print("\n BEST ESTIMATOR ")
print(gsc.best_estimator_)

print("\n BEST PARAMETERS")
print(gsc.best_params_)
```



```
 BEST SCORE:
0.9105937921727396

 BEST ESTIMATOR
RandomForestClassifier(max_depth=40, min_samples_split=5, n_estimators=200)

 BEST PARAMETERS
{'bootstrap': True, 'max_depth': 40, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 200}
```

Figure 23 and 24. Grid Search Hypertuning with Random Forest Model

Hypertuning for the Random Forest Classifier model involved optional bootstrapping, a max depth of the tree of 40, 50 or none, maximum features for the best split, minimum number of samples at a leaf and ‘n’ estimators for the number of trees in the forest.
Achieving a score of 91.05% the best estimators and parameters were: Applying bootstrapping, a max tree depth of 40, square rooted max features, 1 minimum sample for the leaf, 5 minimum samples for split, and a number of 200 trees.

13. Discussion and Conclusion

This work presented the ways machine learning can impact the Healthcare section, overviewing multiple case studies. It provided answers to our research questions: It was proved that machine learning can be indeed of beneficial use when dealing with global issues like a pandemic, helping by studying the biology of a virus and its reaction with the human biology, for a possible blueprint for a vaccine [20]; machine learning can also be useful when making a diagnosis through X-ray analysis, achieving even a greater accuracy than humans could, but also with present limitations [21].

Additional further analysis on a case study was performed, demonstrating that machine learning can be of great use when predicting an outcome for a specific disease such as cancer, while providing a variety of models to the application, along with additional analysis such as hypertuning in order to receive the best results. Binary classification, feature selection and model validation were some of the machine learning applications discussed in this specific case study. The top ranked model in the case study ended up being the Decision Tree Classifier, hypertuning helped improve the model, but it also had some limitations and incidents such as overfitting, future studies could be done to improve such.

As this study comes to a close, it is believed that the future of machine learning will enable innovative and spectacular changes for the greater good of overall human development.

14. References

[1] Kersting K. Machine Learning and Artificial Intelligence: Two Fellow Travelers on the Quest for Intelligent Behavior in Machines. Front Big Data. 2018 Nov 19;1:6. doi: 10.3389/fdata.2018.00006. PMID: 33693322; PMCID: PMC7931929.

[2] Implementing Machine Learning in Health Care — Addressing Ethical Challenges

Authors: Danton S. Char, M.D., Nigam H. Shah, M.B., B.S., Ph.D., and David Magnus, Ph.D.Author

https://www.nejm.org/doi/full/10.1056/NEJMp1714229

[3] The use of Big Data Analytics in healthcare

Kornelia Batko and Andrzej Ślęzak

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8733917/

[4] Predicting the Future - Big Data, Machine Learning, and Clinical Medicine. Ziad Obermeyer, M.D., M.Phil. and Ezekiel J. Emanuel, M.D., Ph.D.

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5070532/

[5] Machine Learning in Healthcare

Hafsa Habehh and Suril Gohel

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8822225/

[6] Rao S.R., Desroches C.M., Donelan K., Campbell E.G., Miralles P.D., Jha A.K. Electronic health records in small physician practices: availability, use, and perceived benefits. J. Am. Med. Inform. Assoc. 2011;18(3):271–275. doi: 10.1136/amiajnl-2010-000010. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

[7] Woldaregay A.Z., Årsand E., Botsis T., Albers D., Mamykina L., Hartvigsen G. Data-driven blood glucose pattern classification and anomalies detection: Machine-learning applications in type 1 diabetes.

J. Med. Internet Res. 2019;**21**(5):e11030. doi: 10.2196/11030. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

[8] Esteva A., Kuprel B., Novoa R.A., Ko J., Swetter S.M., Blau H.M., Thrun S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature.* 2017;**542**(7639):115–118. doi: 10.1038/nature21056. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

[9] Rajpurkar P, Irvin J, Zhu K, Yang B, Mehta H, Duan T, et al. CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning. *arXiv.* 2017:3–9. [Google Scholar]

[10] Tian L., Zhang D., Bao S., Nie P., Hao D., Liu Y., et al. Radiomics-based machine-learning method for prediction of distant metastasis from soft-tissue sarcomas. Clin. Radiol. 2020;76(2):158.e19–158.e25. [PubMed] [Google Scholar] [Ref list]

[11] Lalmuanawma S, Hussain J, Chhakchhuak L. Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review. Chaos Solitons Fractals 2020 Oct;139:110059 [FREE Full text] [CrossRef] [Medline]

[12]Wynants L, Van Calster B, Collins GS, Riley RD, Heinze G, Schuit E, et al. Prediction models for diagnosis and prognosis of covid-19 infection: systematic review and critical appraisal. BMJ 2020 Apr 07;369:m1328 [FREE Full text] [CrossRef] [Medline]

[13] Ye J. The role of health technology and informatics in a global public health emergency: practices and implications from the COVID-19 pandemic. JMIR Med Inform 2020 Jul 14;8(7):e19866 [FREE Full text] [CrossRef] [Medline]

[14]Vaishya R, Javaid M, Khan IH, Haleem A. Artificial intelligence (AI) applications for COVID-19 pandemic. Diabetes Metab Syndr 2020;14(4):337-339 [FREE Full text] [CrossRef] [Medline]

[15]Shi F, Wang J, Shi J, Wu Z, Wang Q, Tang Z, et al. Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for COVID-19. IEEE Rev Biomed Eng 2020 Apr 16;PP. [CrossRef] [Medline]

[17] Leading Causes of DeathData are for the U.S
https://www.cdc.gov/nchs/fastats/leading-causes-of-death.htm

[18] Siegel RL, Giaquinto, AN, Jemal A. Cancer statistics, 2024. CA: A Cancer Journal for Clinicians 2024; 74(1):1–114. Last accessed May 8, 2024. [PubMed Abstract] .

[19] SEER Cancer Statistics Factsheets: Common Cancer Sites. National Cancer Institute. Bethesda, MD, https://seer.cancer.gov/statfacts/html/common.html. Last accessed May 10, 2024.

[20] Malone, B., Simovski, B., Moliné, C. et al. Artificial intelligence predicts the immunogenic landscape of SARS-CoV-2 leading to universal blueprints for vaccine designs. Sci Rep 10, 22375 (2020). https://doi.org/10.1038/s41598-020-78758-5

[21] Rajpurkar, Pranav & Irvin, Jeremy & Zhu, Kaylie & Yang, Brandon & Mehta, Hershel & Duan, Tony & Ding, Daisy & Bagul, Aarti & Langlotz, Curtis & Shpanskaya, Katie & Lungren, Matthew & Ng, Andrew. (2017). CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. 10.48550/arXiv.1711.05225.
https://web.njit.edu/~usman/courses/cs732_spring19/CheXNet_Yanan%20Yang.pdf

[22] Wolberg, W. H., & Mangasarian, O. L. (1990). "Multisurface method of pattern separation for medical diagnosis applied to breast cytology." Proceedings of the National Academy of Sciences, 87(23), 9193-9196.

[23] W. Wolberg, O. Mangasarian, N. Street, and W. Street. "Breast Cancer Wisconsin (Diagnostic)," UCI Machine Learning Repository, 1993. [Online]. Available: https://doi.org/10.24432/C5DW2B.

[24] Abdeltawab Mahmoud Breast Cancer Wisconsin Dataset
https://www.kaggle.com/code/abdeltawabmahmoud/breast-cancer-detection-accuracy-96

[25] Sanskar Wagavkar Introduction to the Correlation Matrix

https://builtin.com/data-science/correlation-matrix

[26] Vikas Ukani Breast Cancer Wisconsin Data Set

https://www.kaggle.com/code/vikasukani/breast-cancer-prediction-using-machine-learning

[27] Journal Article LaValley, Michael P. Logistic Regression 2008 Circulation 2395-2399 117 18

doi:10.1161/CIRCULATIONAHA.106.682658

https://www.ahajournals.org/doi/abs/10.1161/CIRCULATIONAHA.106.682658

[28] Du, Wenliang and Zhan, Zhijun, "Building Decision Tree Classifier on Private Data" (2002).

Electrical Engineering and Computer Science. 8.

https://surface.syr.edu/eecs/8

[29] Petkovic, D., Altman, R., Wong, M., & Vigil, A. (2018). Improving the explainability of Random Forest classifier - user centered approach. Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, 23, 204–215.

[30] Abu Alfeilat, H. A., Hassanat, A. B. A., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., & Prasath, V. B. S. (2019). Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. Big data, 7(4), 221–248. https://doi.org/10.1089/big.2018.0175

[31] Bhavsar, Himani P. and Mahesh Panchal. “A Review on Support Vector Machine for Data Classification.” (2012).

[32] James Bergstra and Yoshua Bengui. “Random Search for Hyper-Parameter Optimization” Journal of Machine Learning Research (2012).