Machine Learning Engineer Nanodegree (2019)
Capstone Report

Sameh Adel
June 4, 2019

Breast Cancer Detection Using Machine Learning

I. Definition

Project Overview

This project collects historical measurements for people who have been tested against breast cancer, some of them thankfully have no cancer, but some carries a breast cancer which the most common cancer in the world.

In this project we used these historical measurements in order to make the future diagnosis more accurate.

Imagine this scenario, someone in early stage of the Breast Cancer have been sent to two or three Doctors and they have never seen that he/she carries a cancer, so he/she thanks the god and went home. Can you imagine the chance of getting treated in this early stage that have been wasted by these Doctors! Imagine if these Doctors have a tool that could asset them to diagnose the cancer with high accuracy, would you think this would lead to a better cancer treatment?

This is what "Breast Cancer Detection Using Machine Learning" project all about. It uses these historical measurements to train a Machine Learning model to diagnose the future possible patients. This model takes inputs of some measurements that measured by the Doctors and accurately this model will make classification according to this data.

Problem Statement

The goal of this problem is to use the power of Machine Learning algorithms to take the dataset of past measurements and understand which most features that lead to the Breast Cancer? Also, to predict the likelihood of future patients to be diagnosed as sick.

So, given important measurements of a future patient we can train a ML algorithm to predict if he/she carries a Breast Cancer easily and accurately.

Evaluation Metrics

I going to use some metrics and techniques to evaluate the selected models and the final model, these metrics are:

1) Learning Curve to help in overfitting and underfitting detection
2) F1 score based on Precision and Recall
3) Confusion Matrix to help in detecting **False Negatives** and **False Positives.**

II. Analysis

Data Exploration

For this problem I have downloaded a dataset called ''Breast Cancer Wisconsin (Diagnostic) '' from Kaggle datasets which is originally loaded from UCI Machine Learning datasets.

This dataset contains 569 rows each row represents one observation, each observation has 32 features from which I dropped one unnecessary column from it (id column).

Features of this dataset are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

These features come as following:

Number of instances: **569**
Number of attributes: **32** (ID, diagnosis, 30 real-valued input features):

1. **id** (patient id)
2. **diagnosis** (M: Malignant, B: Benign)
3. **radius_mean** (mean of distances from center to points on the perimeter)
4. **texture_mean** (standard deviation of gray-scale values)
5. **perimeter_ mean** (This is the measure around the breast from the outer side to the inner side going through the nipple)
6. **area_mean** (area of the breast)
7. **smoothness_mean** (local variation in radius lengths)
8. **compactness_mean** (perimeter^2 / area)
9. **concavity_mean** (severity of concave portions of the contour)
10. **concave points_mean** (number of concave portions of the contour)
11. **symmetry_mean**
12. **fractal_dimension_mean** (mean for "coastline approximation" - 1)
13. **radius_se** (standard error for the mean of distances from center to points on the perimeter)
14. **texture_se** (standard error for standard deviation of gray-scale value)
15. **perimeter_se** (perimeter standard error)
16. **area_se** (area standard error)
17. **smoothness_se** (standard error for local variation in radius lengths)
18. **compactness_se** (standard error for perimeter^2 / area - 1.0)
19. **concavity_se** (standard error for severity of concave portions of the contour)

20. **concave points_se** (standard error for number of concave portions of the contour)
21. **symmetry_se** (symmetry standard error)
22. **fractal_dimension_se** (standard error for "coastline approximation")

The remaining features hold the same attributes but calculating the worst for examp le area worst.
In this this project I have applied some basic statistics to get intuition about the data set, it's shown below: (Note table shown here applied for most important features, but I applied statistics for all features in the project.)

	concave points_worst	perimeter_worst	concave points_mean	radius_worst	perimeter_mean	area_worst	radius_mean	area_mean	concavity_mean
count	569	569	569	569	569	569	569	569	569
mean	0.114606223	107.2612127	0.048919146	16.26918981	91.96903339	880.5831283	14.1272917	654.889104	0.088799316
std	0.065732341	33.60254227	0.038802845	4.83324158	24.29898104	569.3569927	3.52404883	351.914129	0.079719809
min	0	50.41	0	7.93	43.79	185.2	6.981	143.5	0
25%	0.06493	84.11	0.02031	13.01	75.17	515.3	11.7	420.3	0.02956
50%	0.09993	97.66	0.0335	14.97	86.24	686.5	13.37	551.1	0.06154
75%	0.1614	125.4	0.074	18.79	104.1	1084	15.78	782.7	0.1307
max	0.291	251.2	0.2012	36.04	188.5	4254	28.11	2501	0.4268

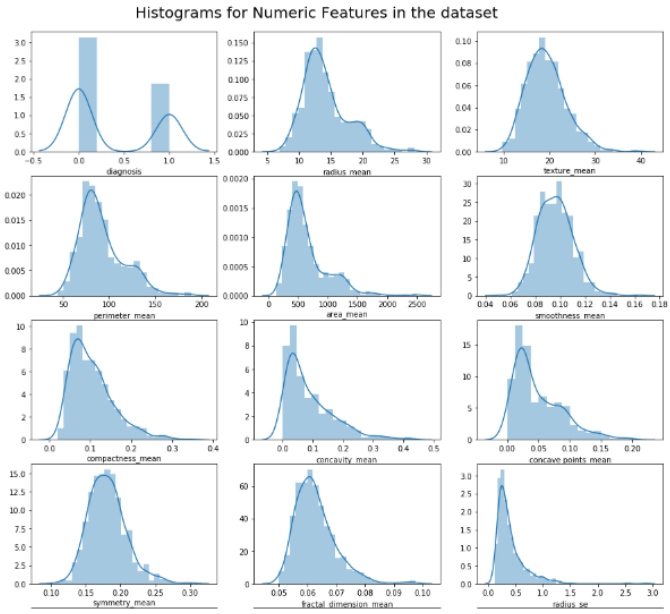
	compactness_mean	compactness_worst	smoothness_worst	perimeter_se	symmetry_worst	texture_worst	area_se	radius_se	texture_mean
count	569	569	569	569	569	569	569	569	569
mean	0.104340984	0.254265044	0.132368594	2.866059227	0.290075571	25.6772232	40.3370791	0.40517206	19.28964851
std	0.052812758	0.157336489	0.022832429	2.021854554	0.061867468	6.146257623	45.4910055	0.27731273	4.301035768
min	0.01938	0.02729	0.07117	0.757	0.1565	12.02	6.802	0.1115	9.71
25%	0.06492	0.1472	0.1166	1.606	0.2504	21.08	17.85	0.2324	16.17
50%	0.09263	0.2119	0.1313	2.287	0.2822	25.41	24.53	0.3242	18.84
75%	0.1304	0.3391	0.146	3.357	0.3179	29.72	45.19	0.4789	21.8
max	0.3454	1.058	0.2226	21.98	0.6638	49.54	542.2	2.873	39.28

After applying data exploratory analysis, I found that the data doesn't contain NULL values it's tota lly clean and ready for further steps in the project. But I found **21** outliers in the dataset, so I have r emoved them from the dataset for the safety of the model's performance.

Exploratory Visualization

In this Project I have applied some visualizations in order to get intuition about the d ataset and its distribution and how its featur es corelated with each other.
So, the visualizations used and their purpose s are:

1. **Histograms**: To see the distribution of the dataset and detect the skewness of the fe atures.
2. **Heatmap**: Used to visualize the correlation between each pair of variables in the data set (including the target variable).
3. **Bar plots**: Used to get intuition visually ab out the correlations between the target va riable and the most important features in t he dataset.



After applying these visualizations, I have a clear picture about the most corelated f eatures that highly affect the target variable. So, I have picked the best 19 featur

es and stored them to be checked if they enough to produce high accuracy later i n the project.

Algorithms and Techniques

In this project I have selected 4 Algorithms that I think best for this project

1. Support Vector Machine
2. Random Forest Classifier
3. Logistic Regression
4. MLPClassifier

Further I have filtered them using Learning Curves. After applying learning curves, I have selected two best models. Later I have picked one winner model (Support Vector Machine Classifier), so I'm going to discuss the SVC Algorithm.
SVC is the most Algorithms suitable for this dataset because I have a small dataset with medium number of features, and as known about SVM Algorithms they have high time complexity working with high dimension dataset and also SVC best choice when working with such small dataset and gives good result like what happened in this project.
SVC doesn't have many hyper-parameters to be tuned and this is one of its advantages, so I used GridSearchCV to tune two of its hyper-parameters (kernel, C). After applying GridSearchCV the model achieved high accuracy, so this is the technique I have followed to work with SVC.

Benchmark

Below I listed the selected Algorithms and their performance before filtering them.

Algorithms	Cross Validation (f1-score)
Support Vector Machine	0.96
Logistic Regression	0.97
Random Forest Classifier	0.92
MLPClassifier	0.96

All the above algorithms tested on Learning Curves and Cross-Validation. Both Cross-Validation and Learning Curves agreed that the best to Algorithms that will generalize well are Support Vector Machine and Logistic Regression.

III. Methodology

Data Preprocessing

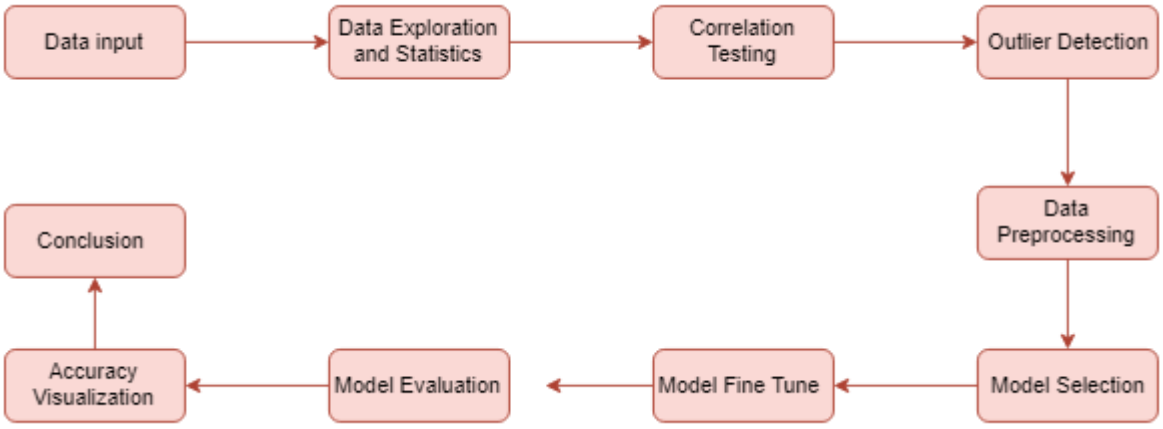
This section of the project applied the following steps:

1. Delete the outliers
2. Fix the skewness of the dataset.
3. Apply transformation on the dataset.
4. Split the dataset into trainset and testset.

The dataset that used for this project is clean and not having NULL values. But the dataset contains outliers in some features and I found it better to remove them because they are small. After that I have fixed the skewness of some features using Log transformation. Finally, I applied StandardScaler on the dataset and split it into two sets (Train, Test), trainset contains 80% of the whole dataset.

Implementation

The following diagram explains the implementation of the project lifecycle:



The implementation goes in order as following:

1. **Data Input**: Loaded the dataset and stored it in *Dataframe* using Pandas.
2. **Data Exploration and Statistics**: Applied statistics and data types explorations using Pandas built-in functions, after that I have used *Seaborn* library to plot histograms to see the distributions of the variables in the dataset.
3. **Correlation Testing**: In this section of the project I used Pearson's Coefficient to test the correlations between each pair of variables against each other and

against the target variable. I used *Seaborn* library to plot Heatmap and Bar plots.
4. **Outlier Detection**: Detected outliers in this step and after that I have removed them.
5. **Data Preprocessing**: In this section I have applied transformation on the dataset using StandardScaler, after that I have split the dataset into trainset and testset.
6. **Model Selection**: In this section I have applied Learning Curves and Cross-Validation in order to filter the four selected models.
7. **Model Fine Tune**: After I have filtered the models I applied in this section GridSearchCV to fine Tune the model hyper-parameters in order to get the best model to be our final model.
8. **Model Evaluation**: In this section I have tested the model using Cross-Validation and also on testset after applying GridSearchCV.
9. **Accuracy Visualization**: I have calculated the confusion matrix elements (True Positives, etc.) and finally plotted the confusion matrix using Heatmap.
10. **Conclusion**: Concluded the final model performance according to its result on trainset and testset.

Refinement

In order to refine my model, I have applied two approaches:

1. **Refine the model's hyper-parameter**: this approached aims to get high accuracy when getting to classification. In this approach I have used GridSearchCV to fine tune the model's hyper-parameters.

SVC before Refinement	**SVC after Refinement**
C = 1	C = 3
Kernel = ''rbf''	Kernel = ''rbf''
F1_score = 0.96	F1_score = 0.97

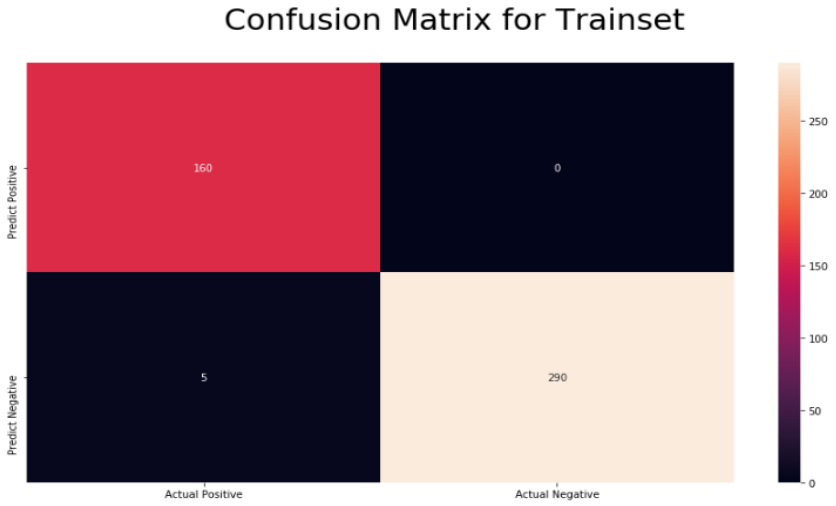
2. **Refine the model's input features**: this approach aims to reduce the number of features that used as input to the model, that's would put us in trade-off situation where we can keep the most important features sacrificing with a little accuracy just to speed-up the model. I have used the most corelated features in the "Refine the model input features" section and tested the result after removing the least important features, and I found that the accuracy not affected a little bit, but still giving great result.

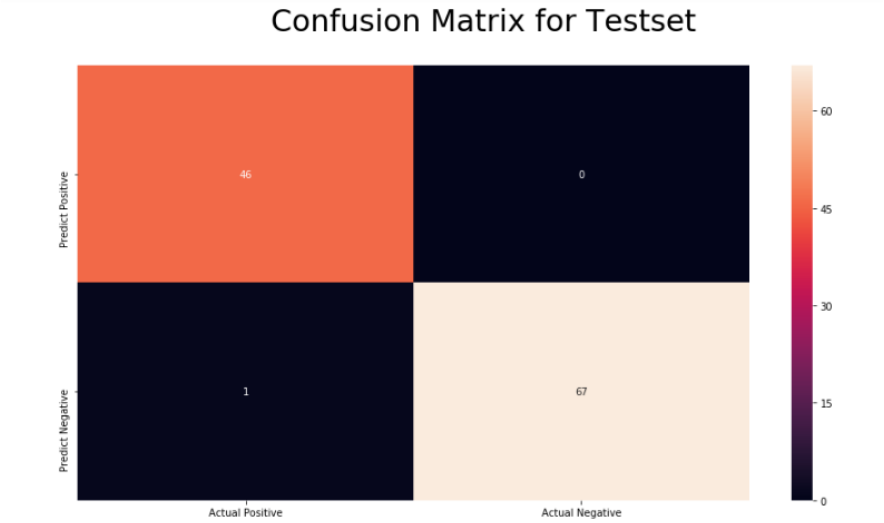
IV. Results

Model Evaluation and Validation

In order to evaluate the final model, I have used Cross-Validation to make sure the model not overfitting the dataset, and tested the model on the testset ,also I applied Confusion Matrix to get to know the True Positives and False Positives, etc.

After Applying Cross-Validation using 10 K-Folds on the final model (SVC), and the average score is **0.97** so this is very good result that indicates that the model not overfitting. Also, the model got score of **0.98** when tested on testset.
Finally, I have plotted the Confusion Matrix to visualize each of the matrix elements on both trainset and testset. These Metrices indicates **ZERO** False Negative in total, but **six** False Positives.





Justification

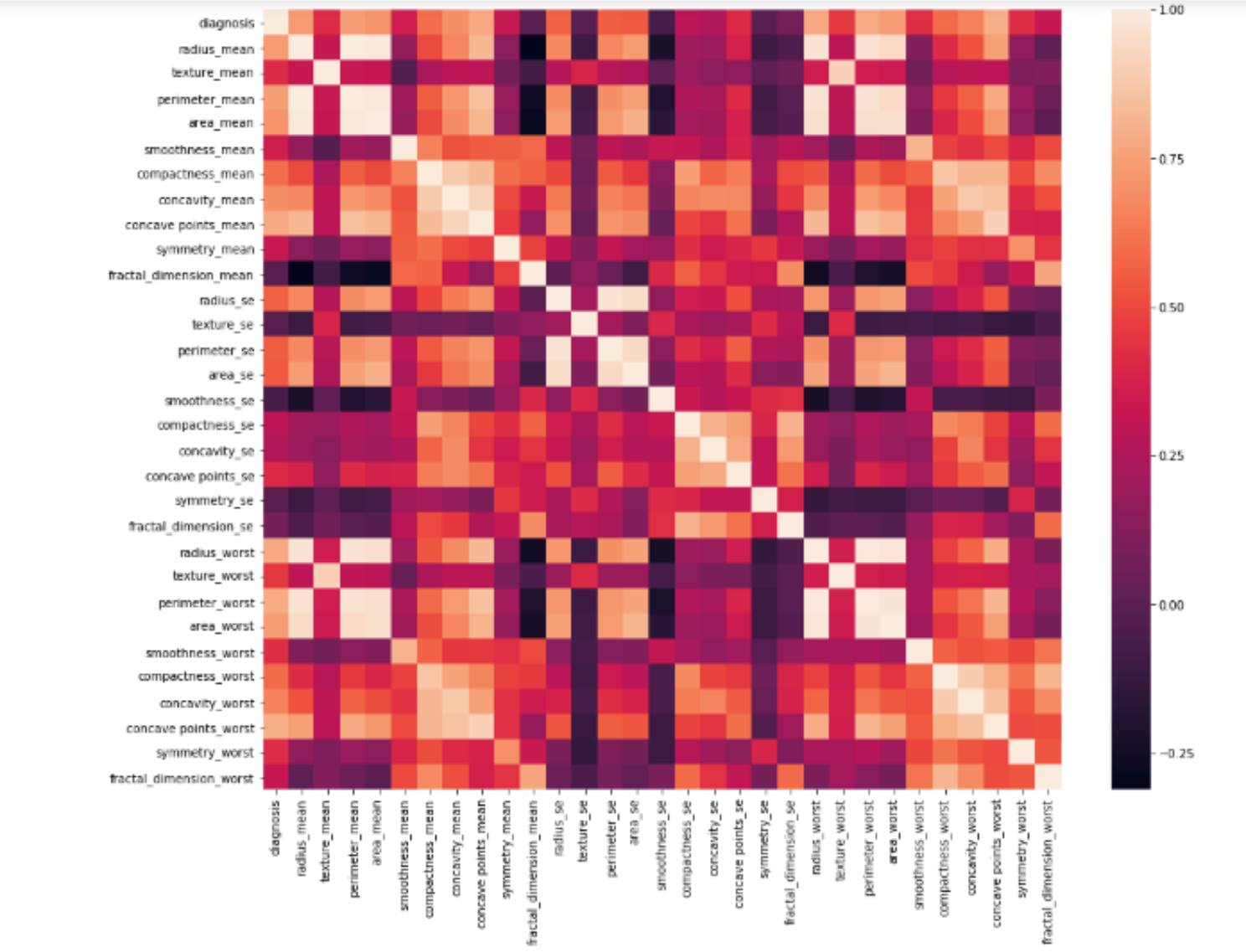
The result obtained at final stage in the project indicates significant improvement on the model accuracy. As benchmark section shows SVC scored **0.96** on 10-Fold Cross-Validation before the refinement step, but after refinement of the model the accuracy jumped to **0.97** this indicates better hyper-parameter of the model.

The final result on testset shows great solution to the problem as it scored **ZERO** in total of False Negative and only **6** in total for False Positive. So these results indicate good solution to the problem.

V. Conclusion

Free-Form Visualization

For the seek of determining how relevant the features are? I have plotted Heatmap and Bar plots to test the correlation of each pair of variables including the target variable. The more the cell corresponding to pair of variables get whiter or darker the more they are corelated with each other.



Reflection

This problem has been gone through some processes to reach the final solution. First of all, the data have been loaded and explored using Statistics and Histograms, this showed the distribution of the dataset and helped interpreting the dataset and detect the skewness of some features which have been treated using Logarithm transformation. After that outliers have been detected and removed.

Now the dataset ready to go to preprocessing step, in this step the dataset has been transformed using StandardScaler and split into trainset and testset. The next step is filtering some picked models to best one and make it ready to get to refinement stage of the project.

Finally, the model has been tested using Cross-Validation and further tested on the testset. After achieving great results, a Confusion Matrix have been calculated and plotted using Heatmap to explore False Positives and False Negatives, then giving conclusion about these results.

The interesting part of the project is selecting the best model among variety of good ones, but to talk about difficulties I have no difficulties working in this project the dataset is small and no time consuming tasks exist in the project.

But to be fair I didn't think a Machine Learning model would achieve this great result, I thought it may just reach score of **0.9**.

Improvement

The improvement that I see is important to this implementation is to increase the data points as much as possible, increasing the number of data points are best in almost all cases. I see that if we are going to increase the number of data points we will need to apply the feature selection step that applied later in this project to achieve a model that has good time performance and generalize well in testset(s).

I learned how to apply **XGBoost** Algorithms as ensemble method, but unfortunately, I have a software issue working with this library. This Algorithms proved its ability to achieve very good results in many problems so I'm sorry for not using it.