Machine Learning Engineer Nanodegree

Capstone Project

Nourhan Mohamed Fathy

May 20th ,2019

I. Definition

Project Overview

Breast cancer (BC) is the most common invasive cancer in women, and it continues to be a worldwide medical problem since the number of cases has significantly increased over the past decades (~ 4.4 million), with a highly yearly incidence (~1.15 million new diagnosed).

In this project, I trained and tested different classifiers capable of predicting whether the tumor is benign or malignant , the best classifier was XGBoosting Classifier, and was trained on the data provided by UCI Machine Learning Repository on Kaggle.

Problem Statement

Breast tumor features are computed from digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image, using ML to predict whether the cancer is benign or malignant.

Metrics

The model is graded based on accuracy and F beta scores, measured on the test set.

$$accuracy = \frac{true\ positives + true\ negatives}{dataset\ size}$$

$$F_\beta = (1+\beta^2)\frac{precision \times recall}{\beta^2 \times precision + recall}$$

II. Analysis

Data Exploration

The dataset (Breast Cancer Wisconsin(Diagnostic) Data set) provided on Kaggle and UCI Machine Learning Repository, it includes 570 rows (357 benign, 212 malignant), the attribute information are ID number and Diagnosis (M= malignant, B= benign), and ten -features computed for each cell nucleus:
1) Radius. 2) Texture. 3) Perimeter. 4) Area. 5) Smoothness. 6)Compactness. 7) Concavity. 8) Concave points. 9) Symmetry. 10) Fractal dimensions.
The mean, standard error, and worst of these features are computed for each image, resulting in 30 features.

Files:

data.csv(122 KB): the dataset

Line counts:

data.csv:570 rows

Labels:

-id number

-diagnosis: (M = malignant, B = benign)

-Ten real-valued features are computed for each cell nucleus:

1.radius (mean of distances from center to points on the perimeter)

2.texture (standard deviation of gray-scale values)

3.perimeter

4.area

5.smoothness (local variation in radius lengths)

6.compactness (perimeter^2 / area - 1.0)

7.concavity (severity of concave portions of the contour)

8.concave points (number of concave portions of the contour)

9.symmetry

10.fractal dimension ("coastline approximation" - 1))

Observations:

The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

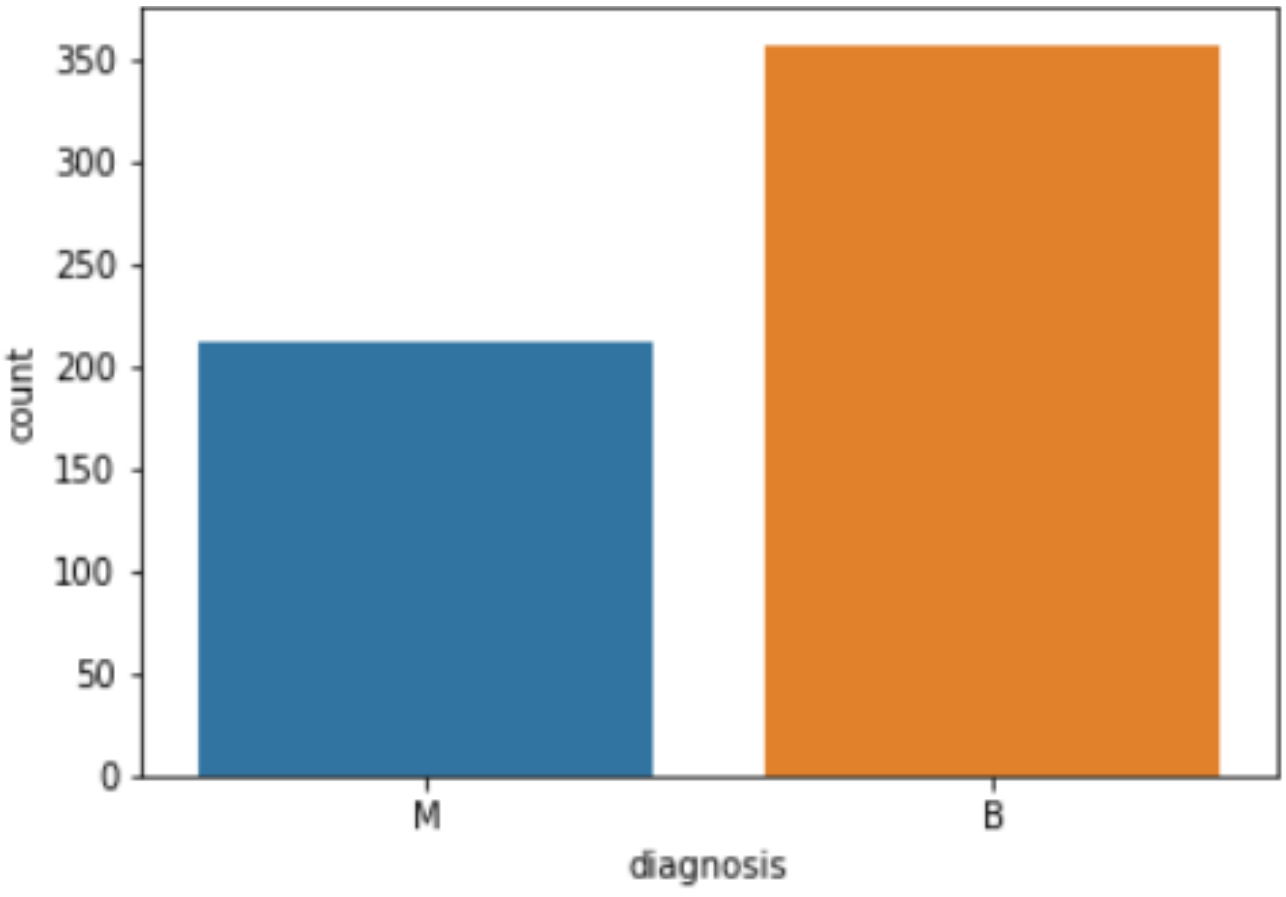
All feature values are recoded with four significant digits.

Missing attribute values: none.

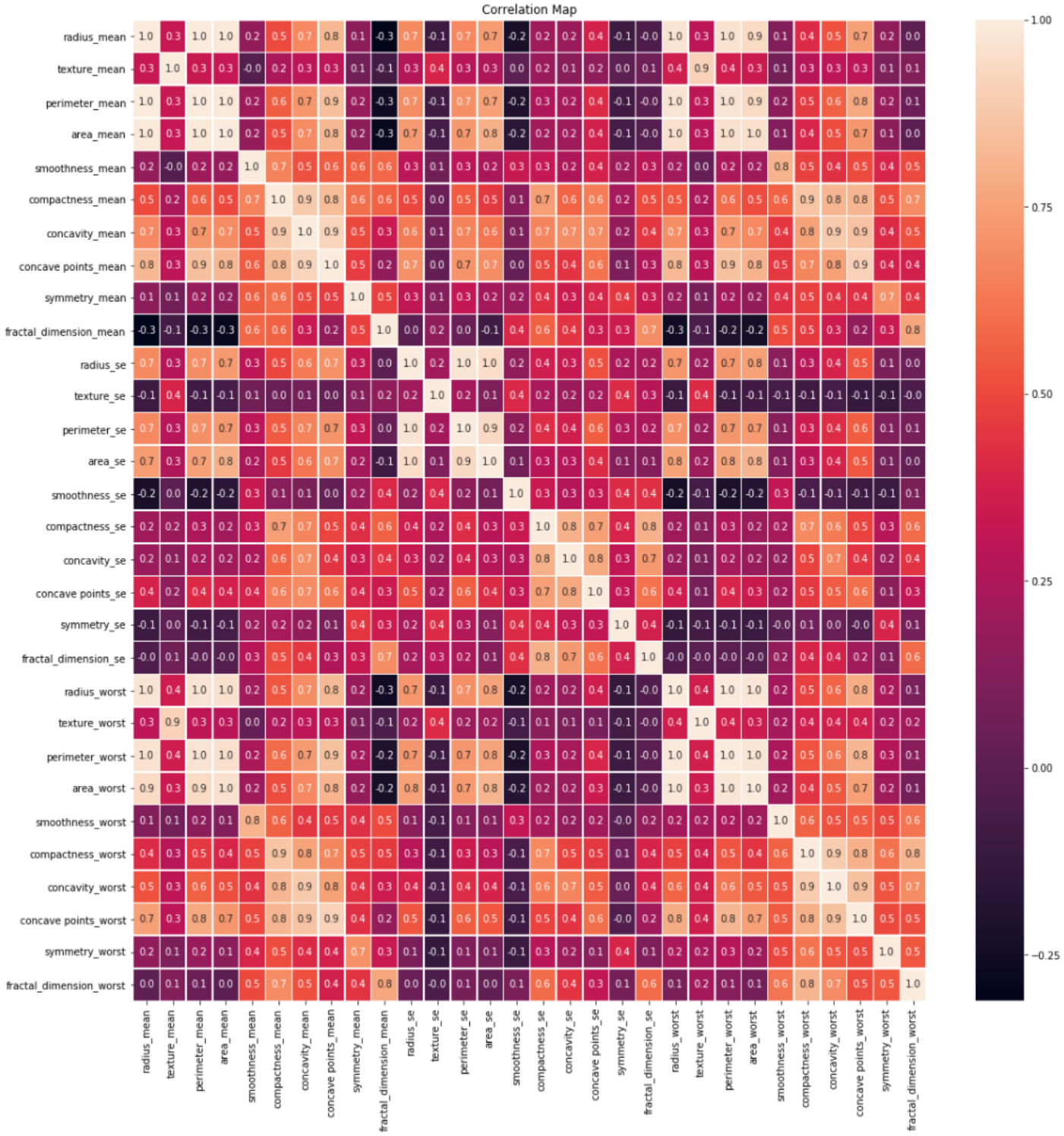
Class distribution: 357 benign, 212 malignant.

Exploratory Visualization

The plot below shows the two classes of the dataset



Visualize features correlations using heat map as shown below



From the heat map above we can figure out that (radius_mean, perimeter_mean, area_mean, radius_worst, perimeter_worst, area_worst) are highly correlated.

Algorithms and Techniques

In this solution to decides to use:

1) (XGBoost) is an implementation of the gradient boosting machines that is highly flexible and versatile while being scalable and fast. XGBoost works with most regression, classification, and ranking problems as well as other objective functions; the framework also gained its popularity in recent years because of its compatibility with most platforms and distributed solutions like Amazon AWS, Apache Hadoop, Spark among others.

2) (SVM) Powerful kernels Maximum margin classifiers, effective in high dimensional spaces Have higher accuracy than some of traditional classifiers and not easily influenced by overfitting.

3)(Random Forest Classifier) Scale quickly, have ability to deal with unbalanced and missing data Generates an internal unbiased estimate of generalization error as forest building progresses. Provides an experimental way to detect variable interactions.

Benchmark

The benchmark model used is Logistic regression model that predicts whether the tumor is benign or malignant, the model scored accuracy of (0.956140350877193), and F beta score of (0.9311740890688259).

III. Methodology

Data Preprocessing

The given dataset had already been well-prepared except for the ‘diagnosis’ column I had to encode it (M=1, B=0), and ‘Unnamed: 32’ column which contains NaN, I just dropped it from the dataset.

Implementation

The software requirement for the implementation is as follows:

- Python

- NumPy

- Pandas

- scikit-learn

- xgboost

-The dataset first was filtered from useless columns and then has been encoded.

-Correlation map was created to find the highly correlated features, after feature selection, dataset is divided into training and testing sets, and models are trained, then the scores are calculated.

-Then dataset is divided into training and testing sets with taking all variables (features) into considerations, and repeat training, and calculating the scores as previous.

-By comparing the results we can deduce the best model is XGBClassifier.

-The chosen model is then optimized using grid search.

Refinement

Hyperparameter tuning

The refinement process started with hyperparameter tuning. Thanks to XGBoost's versatility that its classifier object XGBClassifier is actually compatible with scikit-learn framework, I was able to make use of scikit-learn's GridSearchCV object to perform tuning.
The search space contains values for the following parameters:
-max_depth:

Maximum depth of the tree.
Used to prevent overfitting, as higher depth allows model to learn more specific relations.

-learning_rate:
Control how fast the model tries to converge.

-gamma:
Minimum loss reduce for each node split.

-min_child_weight:
Minimum sum of weights of all observations required in a child node. Used to avoid overfitting as higher values prevent model to learn realtions specific to the dataset.

-max_delta_step:
Control the update step of each tree's weight estimation. Important when classes are imbalanced.

-colsample_bytree:
Control the amount of features to be sampled by each tree.

-reg_lambda:
L2 regularization.

IV. Results

Model evaluation and validation

During development a training set used to evaluate the model.

The final hyperparameters were chosen because they performed the best among the tried combinations.

-max_depth = 6

-learning_rate = 0.1

-gamma = 0.01

-min_child_weight = 1

-max_delta_step = 1

-colsample_bytree = 1

-reg_lambda = 1

-random_state = 42

Justification

The final model after optimization scored 0.986 accuracy and 0.9811 F beta score.

Model	Accuracy Score	F-beta Score
Unoptimized	0.979	0.977
Optimized	0.986	0.9811

V. Conclusion

Free-Form Visualization

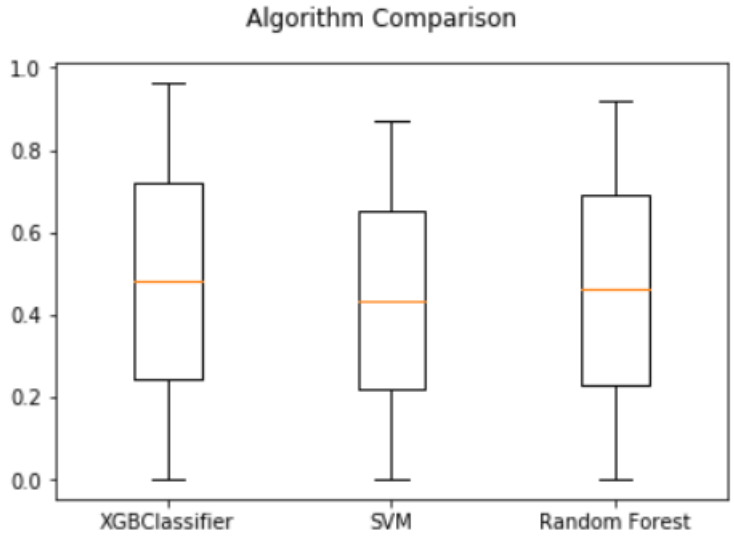


Figure1: Feature selection models. (ignore the features that are highly correlated and take only one feature of them instead of the others).

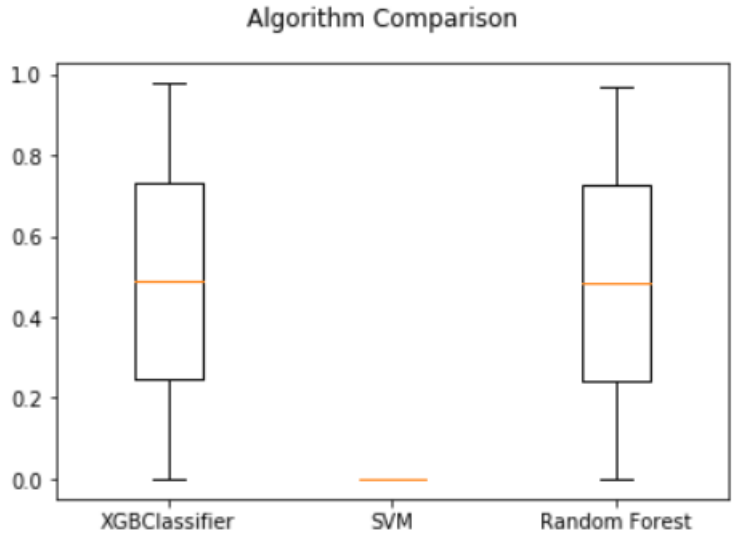


Figure2: All variables models. (include all the features in the dataset).

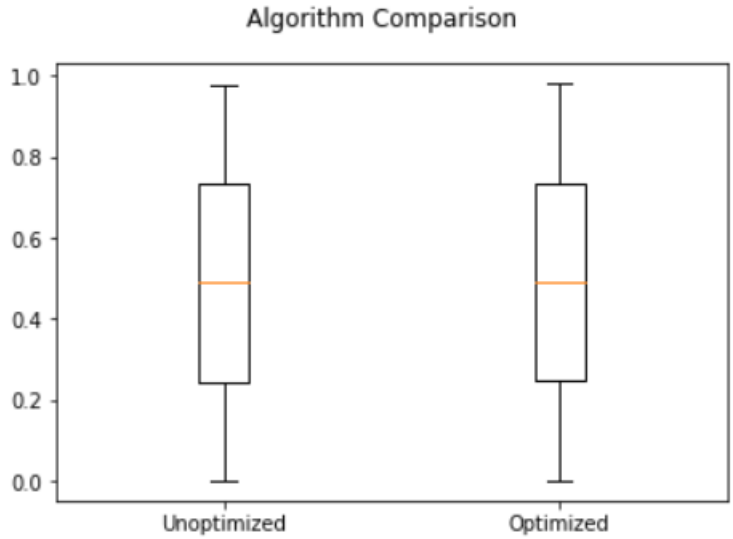


Figure3: final optimized model. (after applying grid search on the highest score model, which was **XGBClassifier**)

-from the previous figures, we can see the developing steps from start to the final optimized model.

Reflection

The process used for this project can be summarized using the following steps:

1.An initial problem and relevant, public datasets were found.
2.The data was downloaded and preprocessed.
3.A benchmark was created for the classifier (Logistic Regression model).
4.Correlations between features were included and highly correlated features were excluded.
4.The classifiers were trained using the data (SVM, XGBClassifier, and Random Forest).
5.Calculate Accuracy and F beta scores for each model.
6.Train the models with all dataset features without excluding any of them.
7.repeat step 5
8.Optimize the highest accuracy and f beta scores' model to get better results.

Improvement

There are not too much that can be improved upon for this project. For example:
-the same model design training on much training set (large dataset).
-take more features, as we saw in earlier figures it affects the results.