Machine Learning Engineering Nanodegree

Capstone Project

Hsiao-Tien Fan

June/25/2017

1. Project Overview

1.1. Domain background

Breast cancer is the second most common type of cancer behind skin cancer in the United States, with an estimated number of a quarter of a million cases expected to be diagnosed in 2017. [1] The diagnosis for breast cancer is usually conducted with a variety of tests such as a physical examination of the breast, mammograms, ultrasounds or biopsy. The selection of which of these tests is to be conducted is at the discretion of the doctor depending on the situation of a particular case. [2] However, it has been shown in previous studies that by analyzing the cytological information obtained from images of biopsied cells using Fine Needle Aspiration (FNA), it is possible to determine whether the sample is cancerous [3][4]. This project will attempt to apply machine learning algorithms datasets obtained from the FNA method and attempt to classify the tumor.

1.2. Problem Statement

The goal of this project is to create a python based model that is able to assess whether the biopsy from a subject's tumor is benign or malignant. The python script will:

- Load the dataset and preprocess the data for training.
- Train a classifier to determine if a subject is benign or malignant.

1.3. Dataset

The dataset used for this project is available from the UCI Machine Learning Repository or from Kaggle's database section [6][7]. There are 569 samples in total, with 357 benign and 212 malignant.

1.4. Solution Statement

For this project, as the number of features is relatively low, it may be a good idea to start with a decision tree and evaluate the performance from there. The decision tree will be trained with a subset of the dataset to predict whether the sample is benign or malignant, with the remaining subset used for testing. The quality of the solution will be measured by the performance of the model's prediction on the test set. The evaluation metrics for determining the performance will be described below.

1.5. Evaluation Metrics

Accuracy - This metric tells us the amount of correct predictions made by the classifier by looking at the true positives and true negatives. For an application such as this it is important for there to be a straight forward and direct way to access how well the classifier has done in solving the problem.

$$accuracy = \frac{No.of\ true\ positives + No.of\ true\ negatives}{Dataset\ size}$$

F1 score - This metric is a measure of accuracy of the model. It is based on the recall and the precision of the model. This test is useful for the data set used in this project as the benign and malignant classes as not equal in size, and may falsely represent the effect of each class on the accuracy metric.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

False negative rate – This metric is important for this particular application as for cancer detection, the worst case scenario is to not recognize the presence of cancer and declare the patient healthy. It is important to tune the model so that the false negative rate is as low as possible.

$$False\ negative\ \% = \frac{No.of\ \text{faslse}\ negatices}{No.\ of\ \text{true}\ negatives + No.of\ false\ negatives}$$

2. Analysis

2.1. Data Exploration

Each sample in the dataset has the following features where each is in relation to the cell nucleus:

a) radius (mean of distances from 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)

For each of these features 3 separate numerical values are given: ‘mean’, ‘stand error’ and ‘worst’. Treating each of these as separate features would give 30 features, all of which are numerical. The data also includes the column ‘diagnosis’ for each sample which indicates whether the sample is benign or malignant and is stored as text.

An example of a malignant and benign sample is shown in Tables 1 and 2

diagnosis	M				
radius_mean	17.99	radius_se	1.095	radius_worst	25.38
texture_mean	10.38	texture_se	0.9053	texture_worst	17.33
perimeter_mean	122.8	perimeter_se	8.589	perimeter_worst	184.6
area_mean	1001	area_se	153.4	area_worst	2019
smoothness_mean	0.1184	smoothness_se	0.006399	smoothness_worst	0.1622
compactness_mean	0.2776	compactness_se	0.04904	compactness_worst	0.6656
concavity_mean	0.3001	concavity_se	0.05373	concavity_worst	0.7119
concave points_mean	0.1471	concave points_se	0.01587	concave points_worst	0.2654
symmetry_mean	0.2419	symmetry_se	0.03003	symmetry_worst	0.4601
fractal_dimension_mean	0.07871	fractal_dimension_se	0.006193	fractal_dimension_worst	0.1189

Table.1

diagnosis	B				
radius_mean	13.54	radius_se	0.2699	radius_worst	15.11
texture_mean	14.36	texture_se	0.7886	texture_worst	19.26
perimeter_mean	87.46	perimeter_se	2.058	perimeter_worst	99.7
area_mean	566.3	area_se	23.56	area_worst	711.2
smoothness_mean	0.09779	smoothness_se	0.008462	smoothness_worst	0.144
compactness_mean	0.08129	compactness_se	0.0146	compactness_worst	0.1773
concavity_mean	0.06664	concavity_se	0.02387	concavity_worst	0.239
concave points_mean	0.04781	concave points_se	0.01315	concave points_worst	0.1288
symmetry_mean	0.1885	symmetry_se	0.0198	symmetry_worst	0.2977
fractal_dimension_mean	0.05766	fractal_dimension_se	0.0023	fractal_dimension_worst	0.07259

Table.2

There is no missing data in this dataset, however when the data is read there is a column 32 that is 'NaN' in every row. This needs to be removed. It can be seen that the typical value between features varies, ranging from in the hundreds to decimal values. These features will need to be normalized before being applied to a learning algorithm.

2.2. Exploratory Visualization

In order to understand the features being used in this project, it is a good idea to look at the distributions of the feature data. For an number of the features there seemed to be a certain amount of skew in the data, such as it not being normally distributed, or the values being clustered around a certain value. Examples of this can be seen Figures 1 and 2.

For these cases, the data needs to be normalized to prevent unintended effects during the training phase. The way this is done is discussed in the data preprocessing section.

For the sake of saving space in the main body of the report, not all the plots for each feature is displayed in this section. Please refer to Appendix A.1 for the comprehensive list.

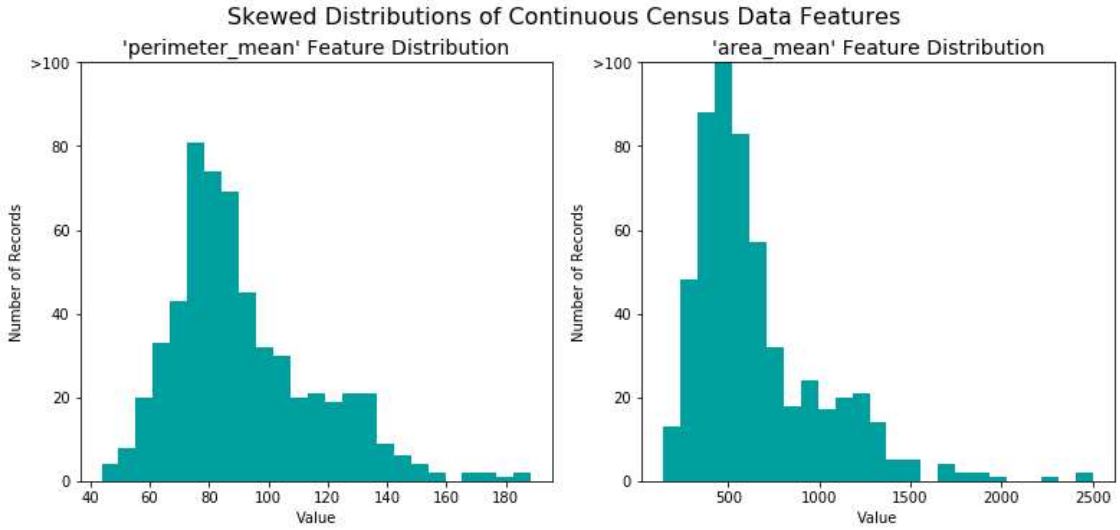


Figure 1

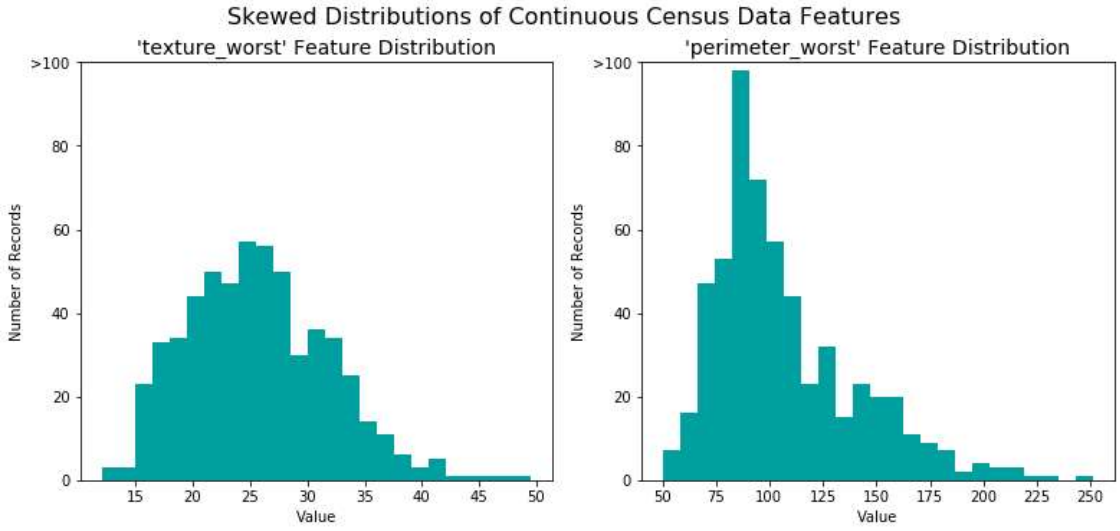
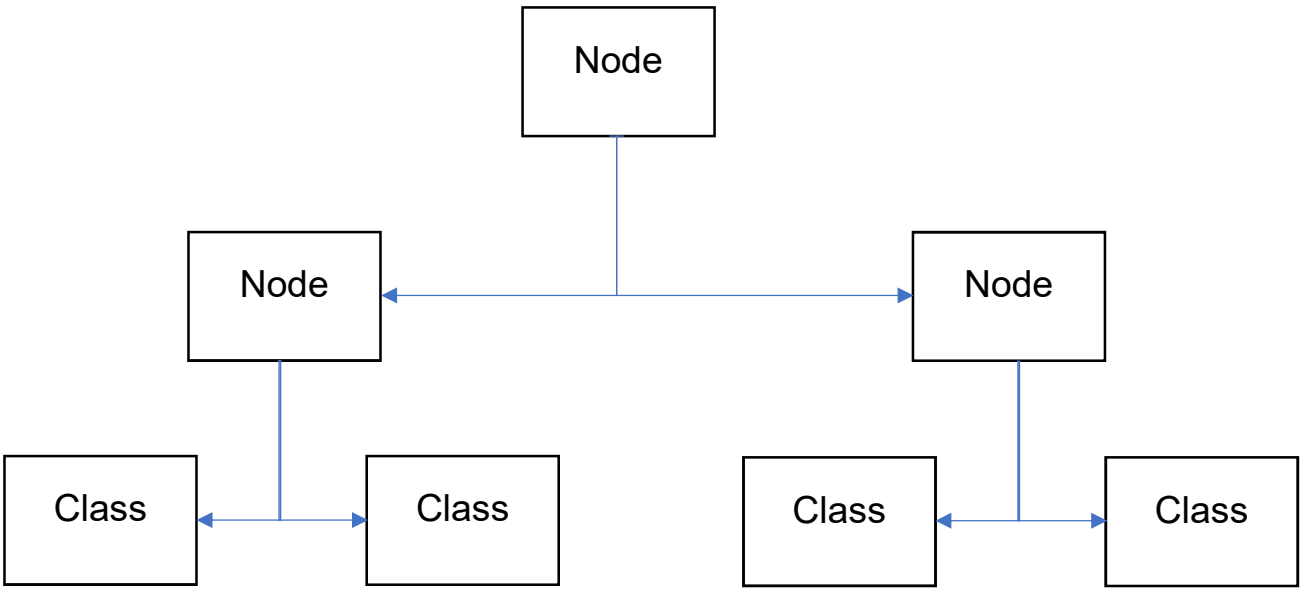


Figure 2

2.3. Algorithms and Techniques

The decision tree classifier is used for this project. It is a powerful classifier and can be easily trained with the dataset provided. The decision tree classifier is based on the concept of making an assessment of the data set at its current 'position' in the model. At each of the nodes, a decision is made on the data set, such as whether it is above a threshold or not. The answer of the assessment will place allow the model to decide which new position the model will move to. For the classification case the position at the end of

the model will determine which class the data point is placed in. This can be seen in the example diagram below.



The way that the nodes are chosen so that the maximum information gain will be achieved at each split. This means that the 'question' that is asked at each node will be able to split the data into distinct groups that will ultimately assist in the classification task. The training process will determine the structure of the nodes and the leaves (The bottom layer of the tree) and how many branches each node breaks into.

A number of parameters may be adjusted for the decision tree classifier:

- Depth – How many levels the decision tree has
- Sample/leaf – The number of samples required in each leaf for the leaf to exist.
- Required sample for split: The number of samples required to further split a branch

There are other possible parameters, and such parameters can be used to adjust the classifier to avoid overfitting and/or improve performance.

The decision tree classifier is easy and fast to train, which makes it very useful in many applications. However, it can very easily over-train, which is something that needs to be taken into consideration when designing the training process and should be checked with techniques like cross-validation.

2.4. Benchmark Model

A simple benchmark model to use would be the naïve predictor. For this project to be meaningful, it needs to perform better the probability of getting the correct result from purely guessing. The proposed benchmark is the probability of being correct when always assigning the prediction to be malignant. This was chosen, as with cancer prediction, it is better to have false positives rather than false negatives.

3. Methodology

3.1. Data Preprocessing

The first thing that need to be addressed is to separate the ‘diagnosis’ from the data to be used as the label. The labels ‘B’ and ‘M’ are encoded into 0 and 1 respectively to so they can be used for training. Following this, the numerical aspects of the dataset needs to be normalized. As discussed previously in data exploration some of the data is skewed, and to normalized these features, the data point was added to the value of 1, and the natural log was taken The 1 was added so that the new value would be greater than 0. Figures 3 and 4 show the logged version of the same features shown previously.

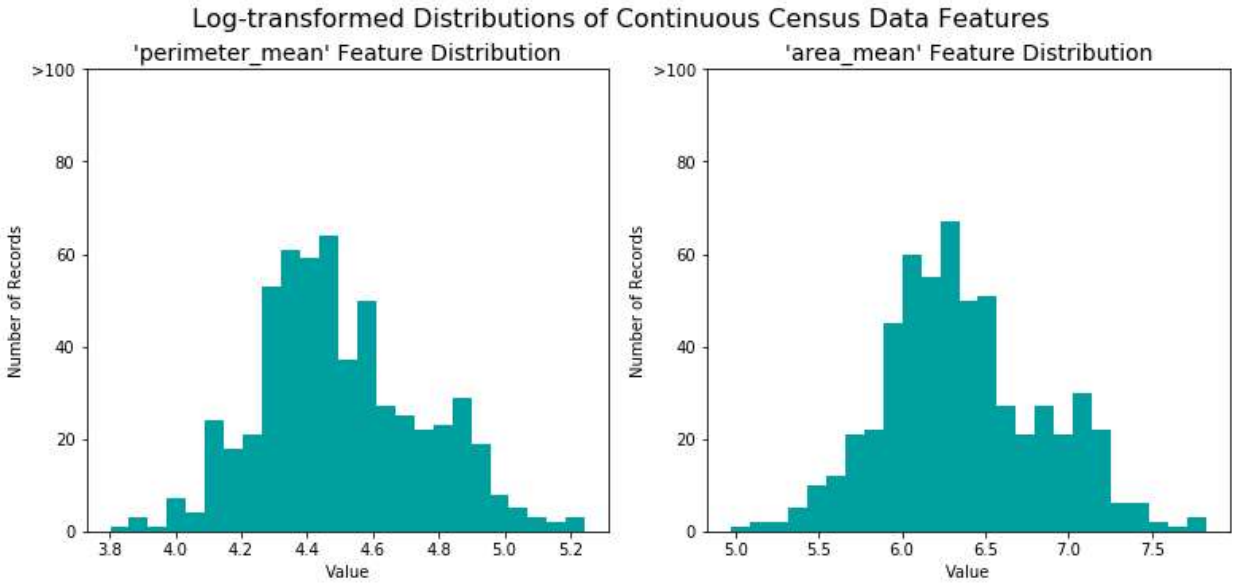


Figure 3

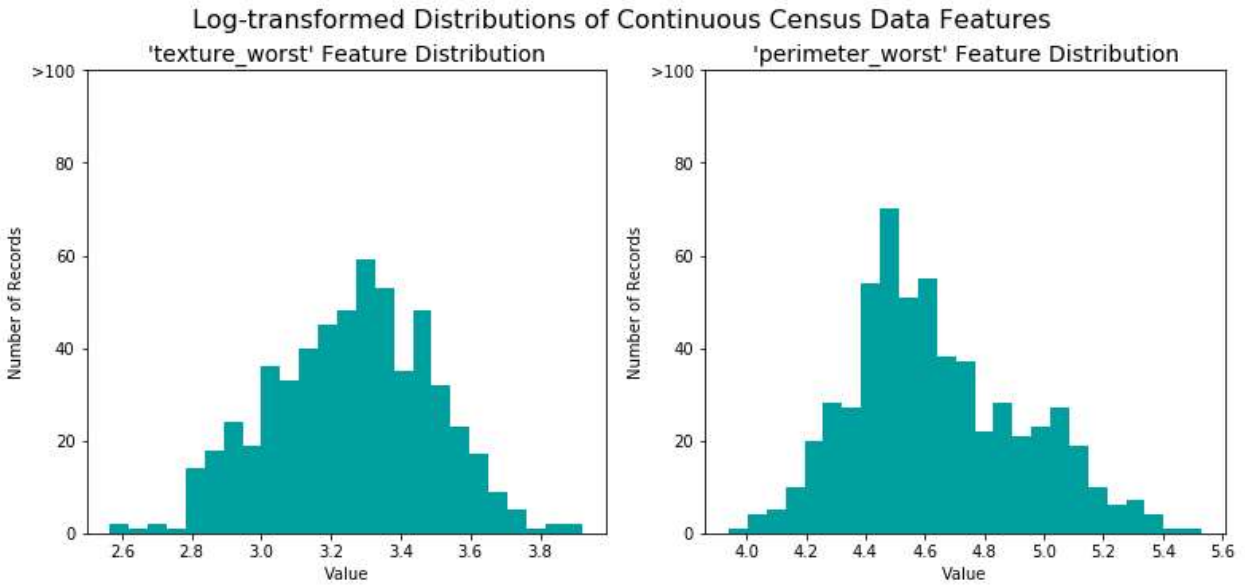


Figure 4

Following this, each of the numerical feature is normalized to between the range of 0 and 1, so that they can be applied to the learning algorithm without unintentionally giving some feature more dominance over others.

3.2. Implementation

The dataset was split into a training set and a testing set at a ratio of 8:2. Even though there seemed to be no apparent pattern to the data, the split was conducted randomly to minimize effects from any pre-existing arrangement of the data.

For this project, the sklearn.tree.DecisionTreeClassifier was used from scikit-learn version 0.18.2. The initial model was trained with all the parameters of the decision tree set on default.

The decision tree classifier was trained using the training set and evaluated on the testing set. Cross-validation was also performed to ensure no over fitting occurred.

3.3. Refinement

In order to optimize the classifier, the grid search method was conducted on ‘depth’, ‘sample/leaf’ and ‘required sample for split’ with each ranging from 1 to 5, 2 to 5 and 2 to 5 respectively, with the F1 score used as the target scorer. A corresponding optimized classifier is found.

Following this, the top five important features are identified for the trained classifier. Using the 'feature_importances_' attritribute of the 'sklearn.tree.DecisionTreeClassifier' it is possible to identify the weighting of the contribution of each feature to the trained model. The top five features are used to retain the classifier. The performance of the newly trained model is assessed.

4. Results

4.1. Model Evaluation and Validation

The accuracy of the default setting is 0.9123 with a F1 score of 0.898

```
Accuracy score on testing data: 0.9123
F1 score on testing data: 0.8980
```

Looking at the score of the cross validation, an accuracy of 0.89 and F1 of 0.89 was achieved, indicating that no overfitting was present.

```
Accuracy: 0.89 (+/- 0.07)
F1: 0.89 (+/- 0.07)
```

Upon applying the grid search method, it was found that the accuracy of the model can be improved from 0.9123 to 0.0.9561 while the F1 increased from 0.898 to 0.9462. The configuration that gave this performance was with a 'depth' of 3, with a 'sample/leaf' of 3 and a 'required sample for split' of 2.

```
Unoptimized model
Accuracy on testing data: 0.9123
F1 score on testing data: 0.8980

Optimized Model
Accuracy on testing data: 0.9561
F1 score on testing data: 0.9462

The optimized configuration of the decision tree:
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=5,
            max_features=None, max_leaf_nodes=None,
            min_impurity_split=1e-07, min_samples_leaf=4,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            presort=False, random_state=0, splitter='best')
```

Following this, it was seen that when the model is retrained with only the top five important weights the performance improved further, increasing to 0.9649 from the previous 0.9561. It can be seen from Figure 5 that the first 5 features constitute almost 0.9 of the total weight for prediction the output.

```
Model trained on full data
Accuracy on testing data: 0.9561
F1 score on testing data: 0.9462

Model trained on reduced data
Accuracy on testing data: 0.9649
F1 score on testing data: 0.9574
```

The classifier received a false negative rate of 1.75% and a false positive rate of 1.75%

4.2. Justification

The final accuracy rate of 96.49% for the optimized classifier performs much better than the original benchmark naïve predictor, which is at 37.26% accuracy. Looking at the F1 score, the improvement was also significant, reaching a value of 0.9574 from a previous values of 0.5429. What is worrisome is perhaps the rate of false negatives, which is just shy of 2%. As this algorithm is for cancer detection, it may be important to attempt to further reduce this number as falsely determining cancer cells as benign would be harmful to patients. This being said, the algorithm performs a lot better than the proposed benchmark and does a reasonable job of classifying cancer.

5. Conclusion

5.1. Free-Form Visualization

What was interesting for this project was number of features that contributed to the classification of cancer. It was seen from the feature weights in the plot below that just the first two features contributed to over 80 percent of the weight for the overall features. This was very surprising for me, as I had expected the mechanism to be complex. It can also be seen from the plot that the concave point feature appeared twice (worse and standard error) in

the top five features. This shows that concave is perhaps the defining feature to look for when trying to classify cancer.

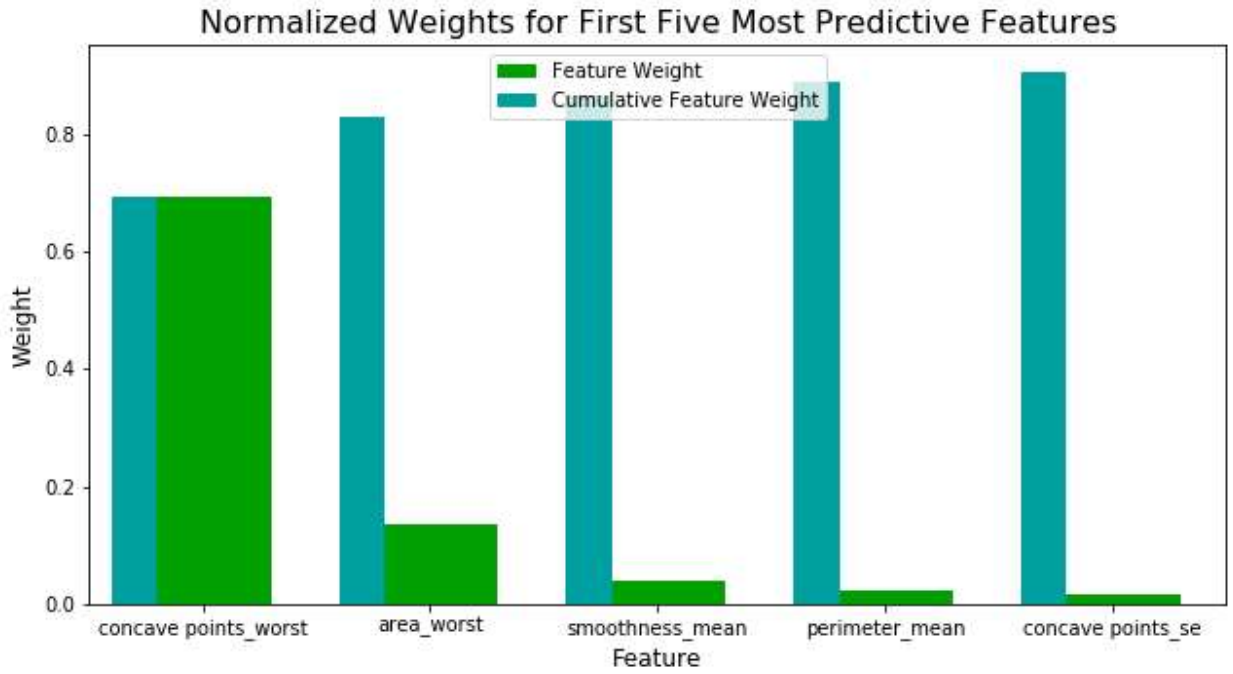


Figure 5

5.2. Reflection

Flow of the project:

- Problem identification: Breast cancer classification from cell cytology
- Obtain public dataset
- Preprocess data: Separate label from data and normalize features
- Determine benchmark and metrics
- Train and test with default decision tree
- Employ cross validation
- Evaluate results
- Apply optimization with grid search
- Compare initial and final results to benchmark

The project was very enlightening in terms of the process of using a completely unfamiliar dataset. Not being of medical background, the significance of each feature was unclear throughout the majority of the project. I wasn't even sure that this task would be achievable when I started, let alone achieving an accuracy of over 95%. It really showed me how

stepping through each step can reveal more information about the nature of the data, and affect the following process of determining what to implement, and subsequently affecting the performance of the model.

5.3. Improvement

As mentioned previously, the false negative rate of the current classifier could be considered high for this application. For improvement, it may be a good idea to investigate methods of further decreasing the false negative rate by adjusting the parameters in the decision tree classifier. Another possibility is to investigate other types of classifiers such as support vector machines or naïve Bayes methods so see if the performance can be improved.

6. References

[1]http://www.breastcancer.org/symptoms/understand_bc/statistics

[2]http://www.mayoclinic.org/diseases-conditions/breast-cancer/diagnosis-treatment/diagnosis/dxc-20207942

[3]https://en.wikipedia.org/wiki/Fine-needle_aspiration

[4]Mangasarian, O. L., Street, W. N., & Wolberg, W. H. (1995). Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, *43*(4), 570-577.

[5]W. H. Wolberg and O. L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proceedings of the National Academy of Sciences, U.S.A., 87:9193-9196, 1990.

[6]https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

[7] https://www.kaggle.com/uciml/breast-cancer-wisconsin-data