Analysis of the Wisconsin Breast Cancer Dataset

Hari Hara Priya Kannan

Contents

1. Abstract

In this project we will compare three machine learning classification techniques using the Wisconsin Breast Cancer Diagnosis dataset. We will create three classifier models that discriminates benign and malignant breast lumps and test the accuracy of the model and report the performance.

The first part of this project will present the dataset, when and how it was created, if it is noisy, does it have any missing values and so on. This section will focus on understanding the issues that will need to be processed while preparing the data to create the classifier.

In the second part we will create three different classification models on the dataset namely k-Nearest Neighbors, Decision Tree and Support Vector Machine. After building these models we will compare the performance in terms of Confusion Matrix, Classification Error Rate, Precision, Recall and F1 Score.

2. Introduction

Breast cancer is the most common cancer among women and one of the major causes of death among women worldwide. Every year approximately 124 out of 100,000 women are diagnosed with breast cancer, and the estimation is that 23out of the 124 women will die of this disease. When detected in its early stages, there is a 30% chance that the cancer can be treated effectively, but the late detection of advanced-stage tumors makes the treatment more difficult. Currently, the most used techniques to detect breast cancer in early stages are: mammography (63%to 97% correctness), FNA (Fine Needle Aspiration) with visual interpretation (65% to 98% correctness) and surgical biopsy (approximately 100% correctness). Therefore, mammography and FNA with visual interpretation correct-ness varies widely, and the surgical biopsy, although reliable, is invasive and costly.

This paper discusses a diagnosis technique that uses the FNA (Fine Needle Aspiration) with computational interpretation via machine learning and aims to create a classifier that provides a high level of accuracy. Building a classifier using machine learning can be a difficult task if the dataset used is not on its best format or if it is not being correctly interpreted. Therefore, a considerable portion of this work will be spent preparing and comprehending the dataset.

3. Methodology

3.1. Dataset

The dataset used in this paper is publicly available and was created by Dr. William H. Wolberg, physician at the University of Wisconsin Hospital at Madison, Wisconsin, USA. It was donated by Olvi Mangasarian on July 15th,1992.

3.2. Attribute Information

Descriptive Features

1) ID number
2) Diagnosis (M = malignant, B = benign)
   3-32)

Ten real-valued features are computed for each 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)

The dataset contains three sets of above values namely the mean, SE (Standard Deviation) and Worst. We will explore all of the values and build the classifier based on the mean values.

Target Feature

The target feature of the dataset is diagnosis (Benign or Malignant)

3.2 Data Preprocessing

1. The data set was derived as a csv file from the website before loading it into python. The names of the header were labelled appropriately and the data values are values are equivalent to the data in the source (CSV) as the first five rows proved it.

2. The data type of diagnosis (target feature) was converted into the correct data type.

3. No Null values were found and the data was free of noise.

4. As the ID was not useful for this work, this has been removed from the dataset.

The resulting data looks as follows:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave_points_mean	symmet
0	1	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	
1	1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	
2	1	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	
3	1	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	
4	1	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	

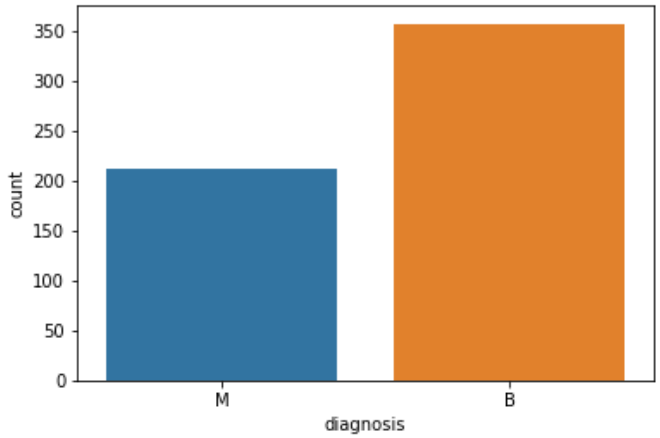
5 rows × 31 columns

3.3 Data Visualization

All of the features were explored and the visualisation part was done using statistical summary as the feature values were numerical data. The plots and graphs were used to visualise the co-relation of the values.

The target feature has two levels and the count of each level is illustrated using a bar chart.

Figure 1: Bar chart of count of target variables by values



The above table shows us that the count of ‘benign’ cases were more than that of ‘malignant’ cases.

All of the feature values are summarised and the below table shows the summary:

None of them are negative values and you can easily see the differences in the center and spread of the data for each variables. The data is pretty balanced. Hence, not many outliers found.

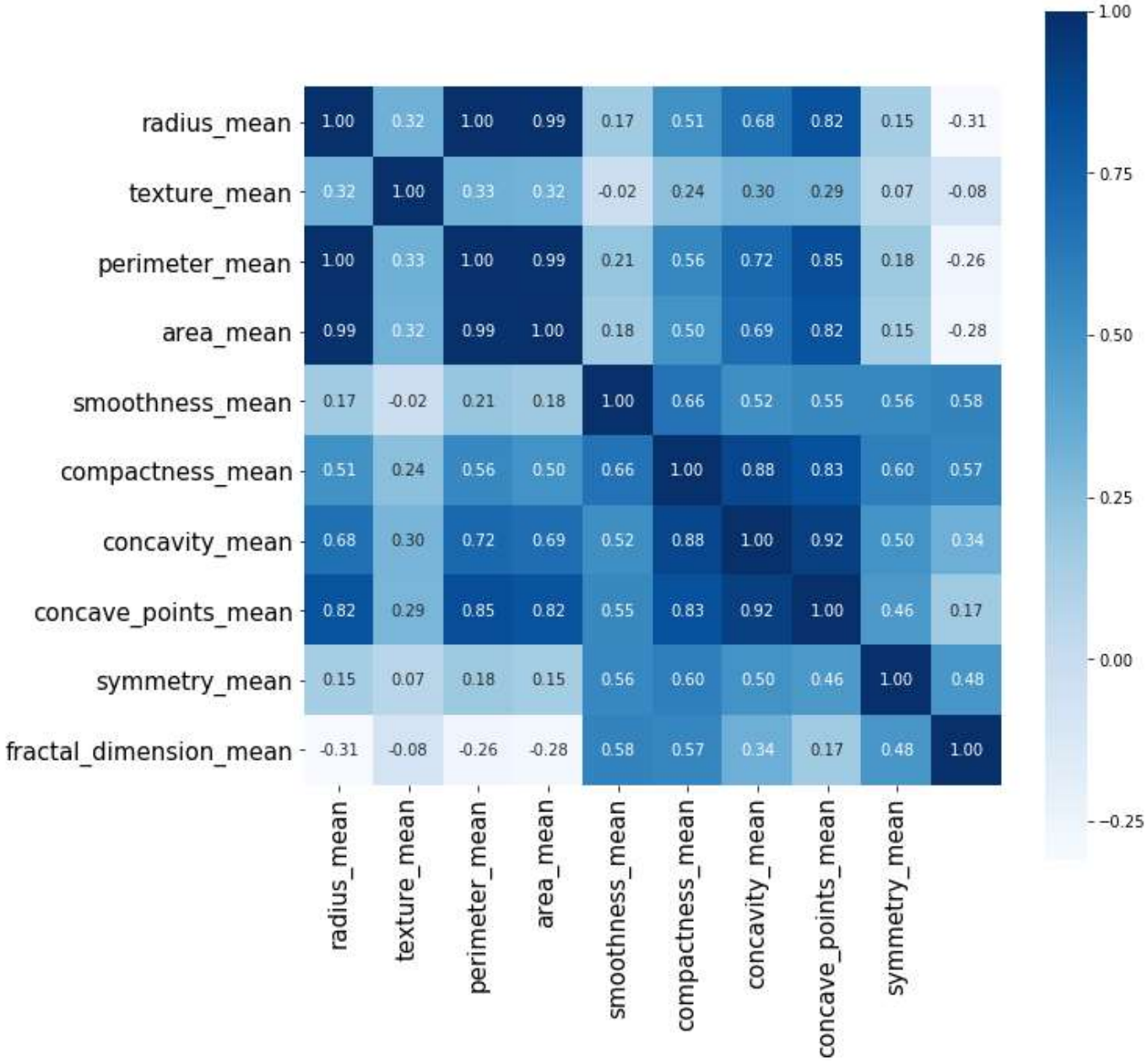
	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave_points_mean
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	0.372583	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919
std	0.483918	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803
min	0.000000	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000
25%	0.000000	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310
50%	0.000000	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500
75%	1.000000	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000
max	1.000000	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200

8 rows × 31 columns

All pairs of mean values are correlated. This suggests a high correlation and a predictable relationship. The illustration is shown below:

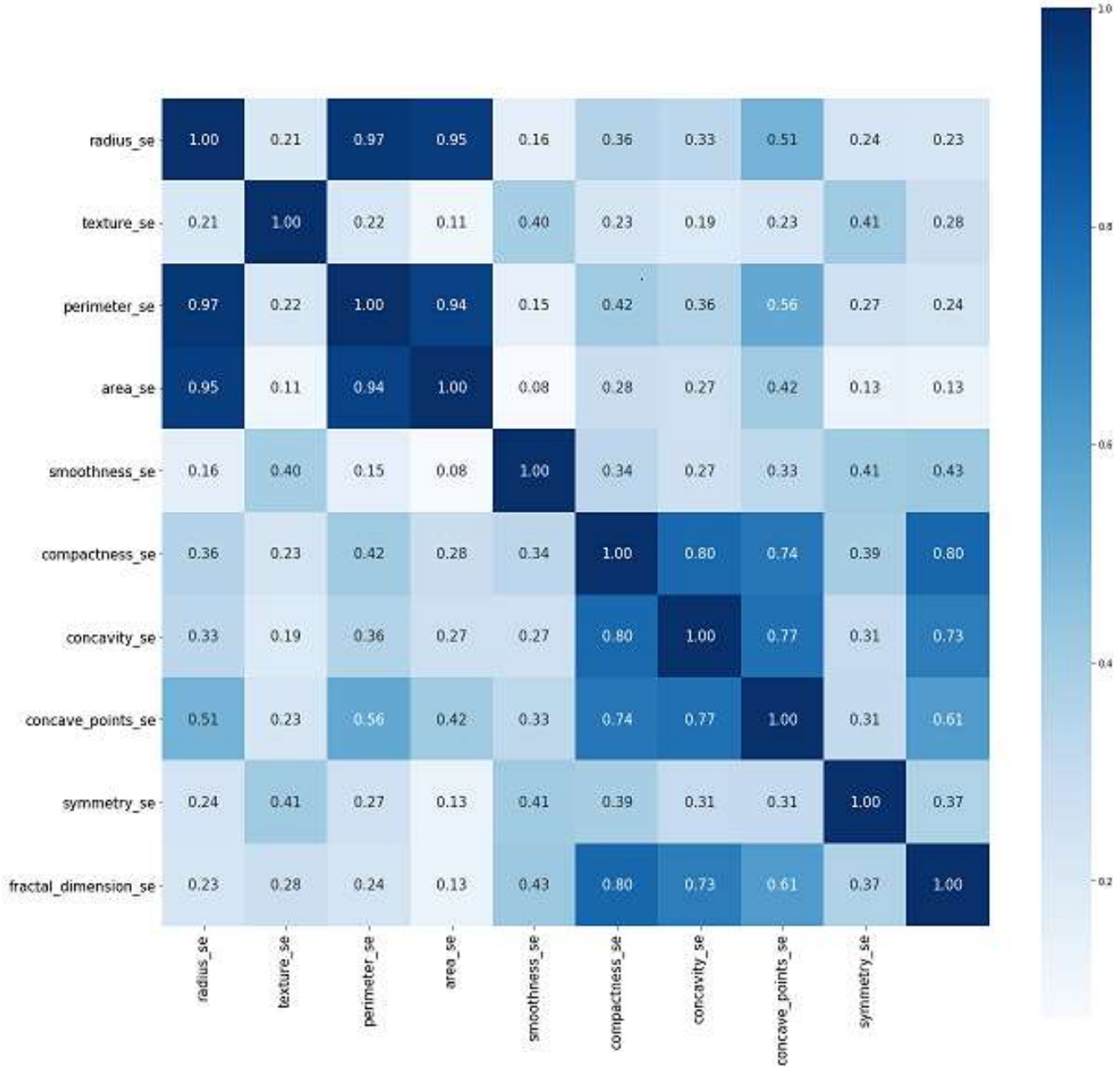
Values of radius, perimeter, area, compactness, concavity and concave points are highly correlated and shows a high correlation with the target feature.

Figure 2: Correaltion heat map of all the 'mean' values in the descriptive features



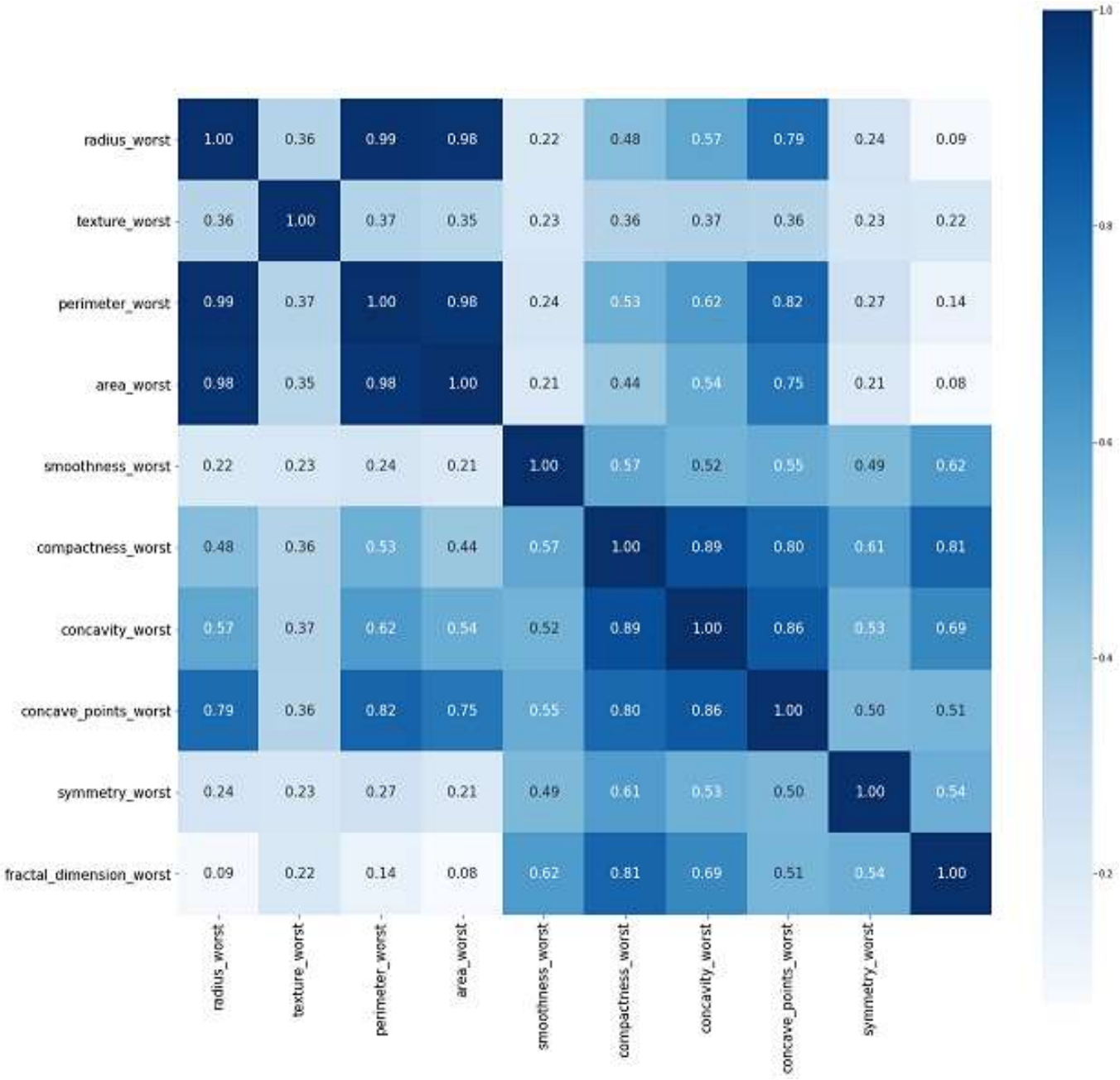
All pairs of standard error values are correlated. This suggests a high correlation and a predictable relationship. The illustration is shown below:

Figure 3 Correaltion heat map of all the 'se' (standard error) values in the descriptive features



All pairs of worst values are correlated. This suggests a high correlation and a predictable relationship. The illustration is shown below:

Figure 4: Correaltion heat map of all the 'worst' values in the descriptive features



We tried to plot the histogram for all the feature variables by the target feature. As a result, the data are lying between the theoretical distribution and not many visible outliers found. This will lead us to achieve the expected result.

The following plots show the above-mentioned features.

Figure 5: Histogram of all mean values by target feature

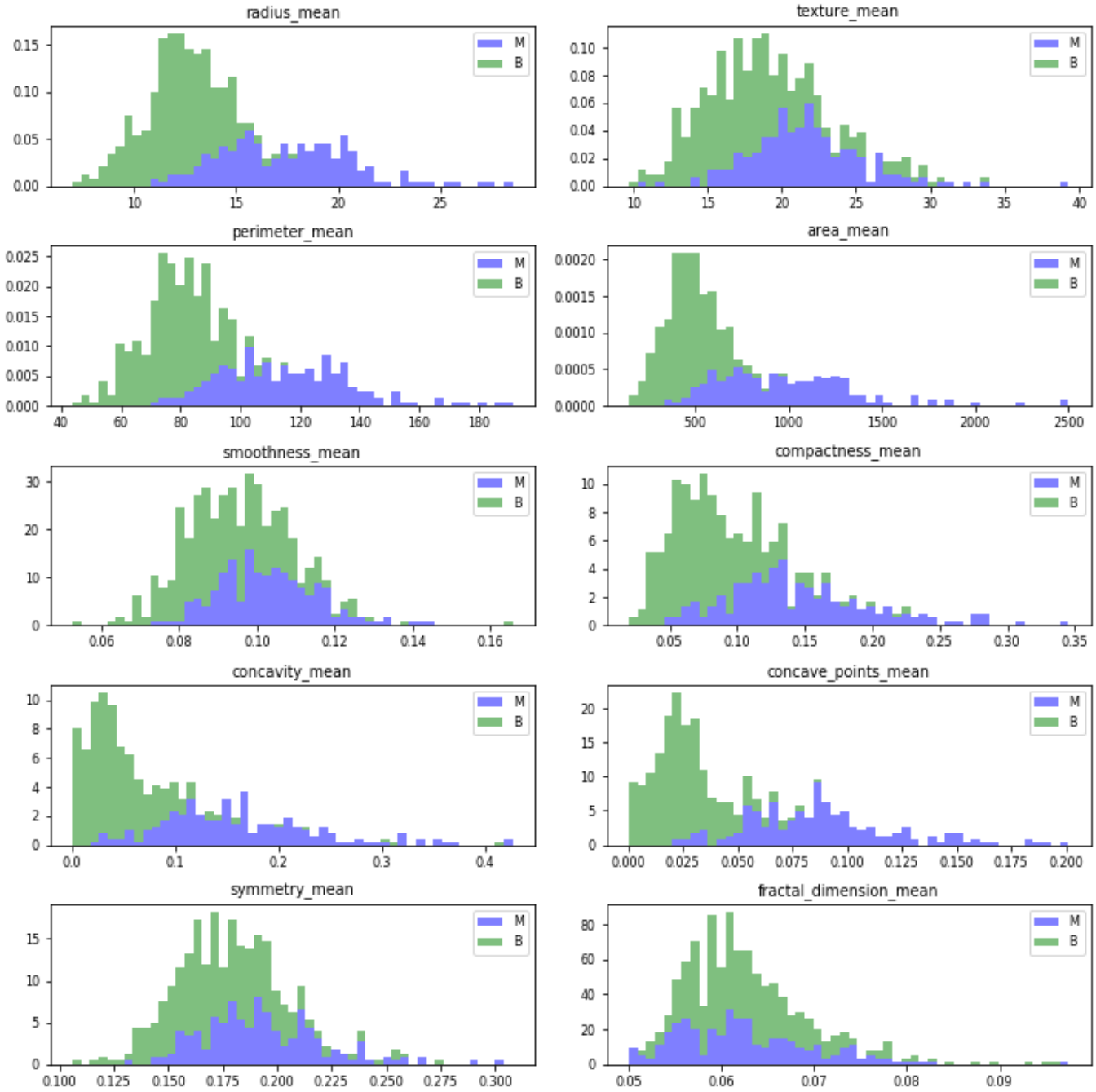


Figure 6: Histogram of all se (standard error) values by target feature

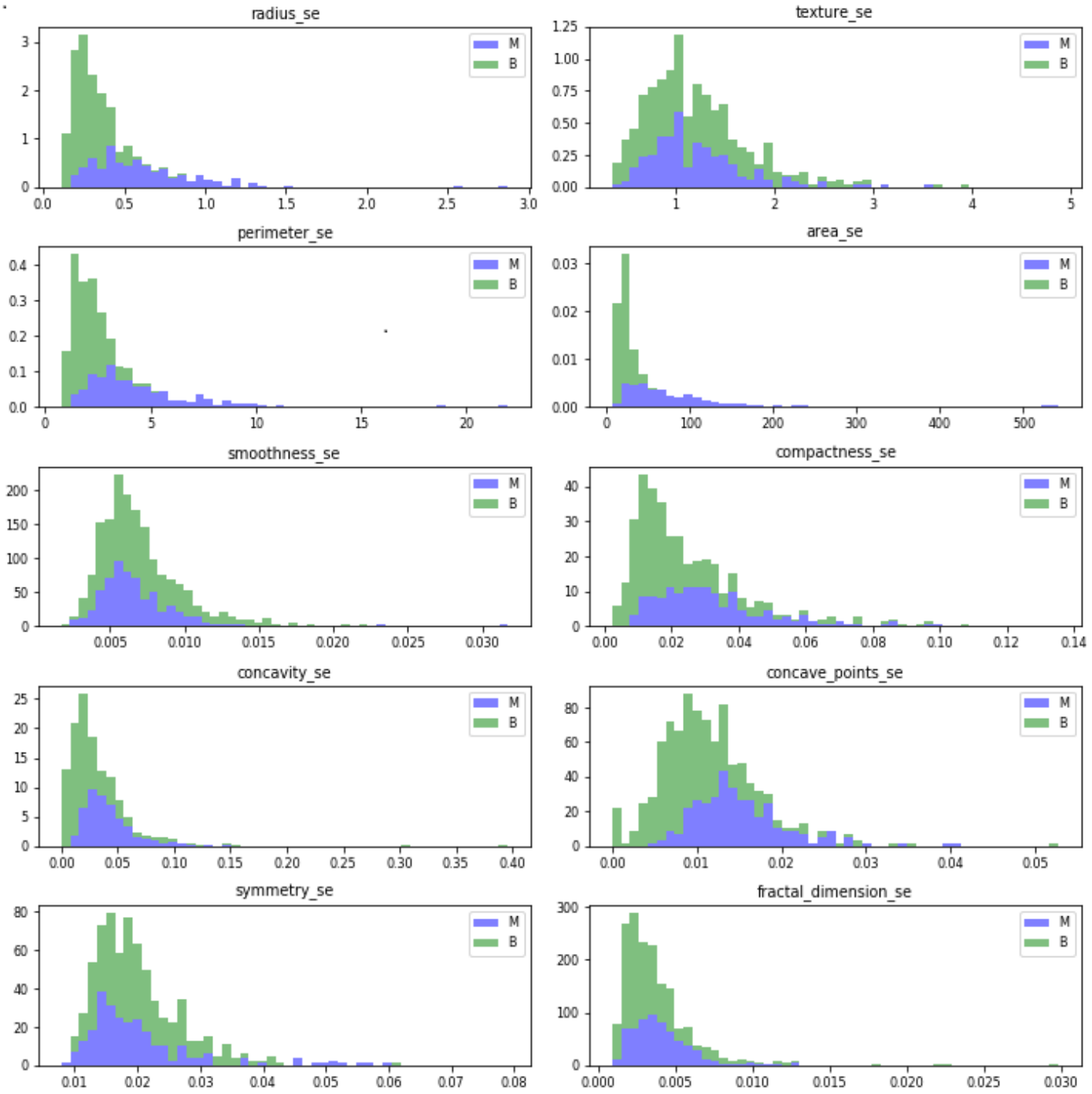
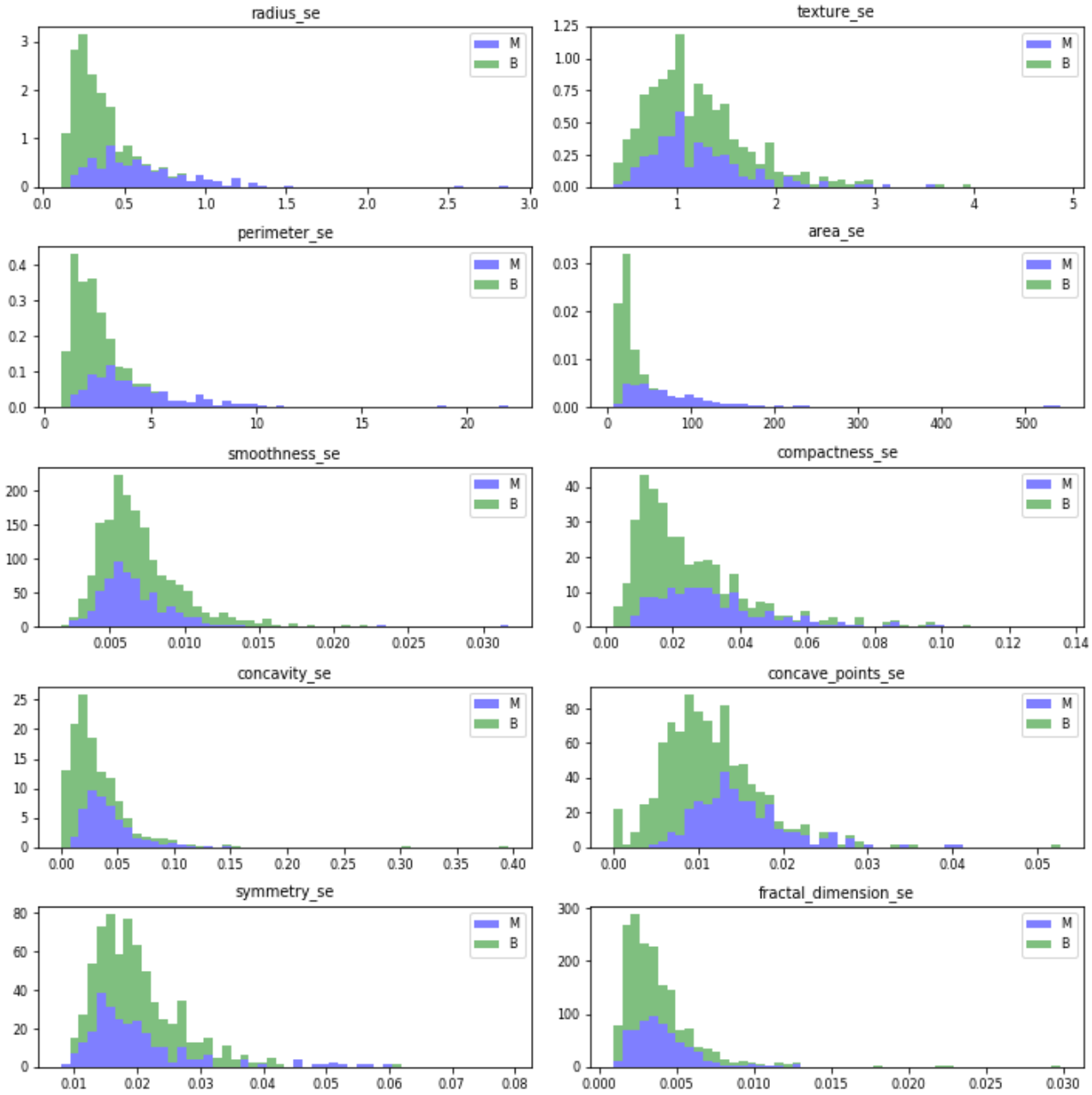


Figure 7: Histogram of all worst values by target feature



None of the histograms show any noticeable large outliers so there is no need for further cleaning of data.

We will use mean values in the descriptive features for our classification

Now the data set is ready for modelling purposes.

3.3 Data Modelling

The modelling was done using three types of data modelling techniques: Decision Tree, KNearest Neighbour and Naïve Baiyes Algortihms.

For all the 3 tools, we followed below the three steps:

1. Splitting data into training set and test set
2. Training the data
3. Testing the accuracy of model on the test set

Model 1: Decision Tree

We applied decision tree on the training data using default parameters. The criterion we used was 'gini' and min_samples_split = 2.

```
DecisionTreeClassifier()
```

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```

Accuracy of Decision Tree classifier on training set: 1.00
Accuracy of Decision Tree classifier on test set: 0.93

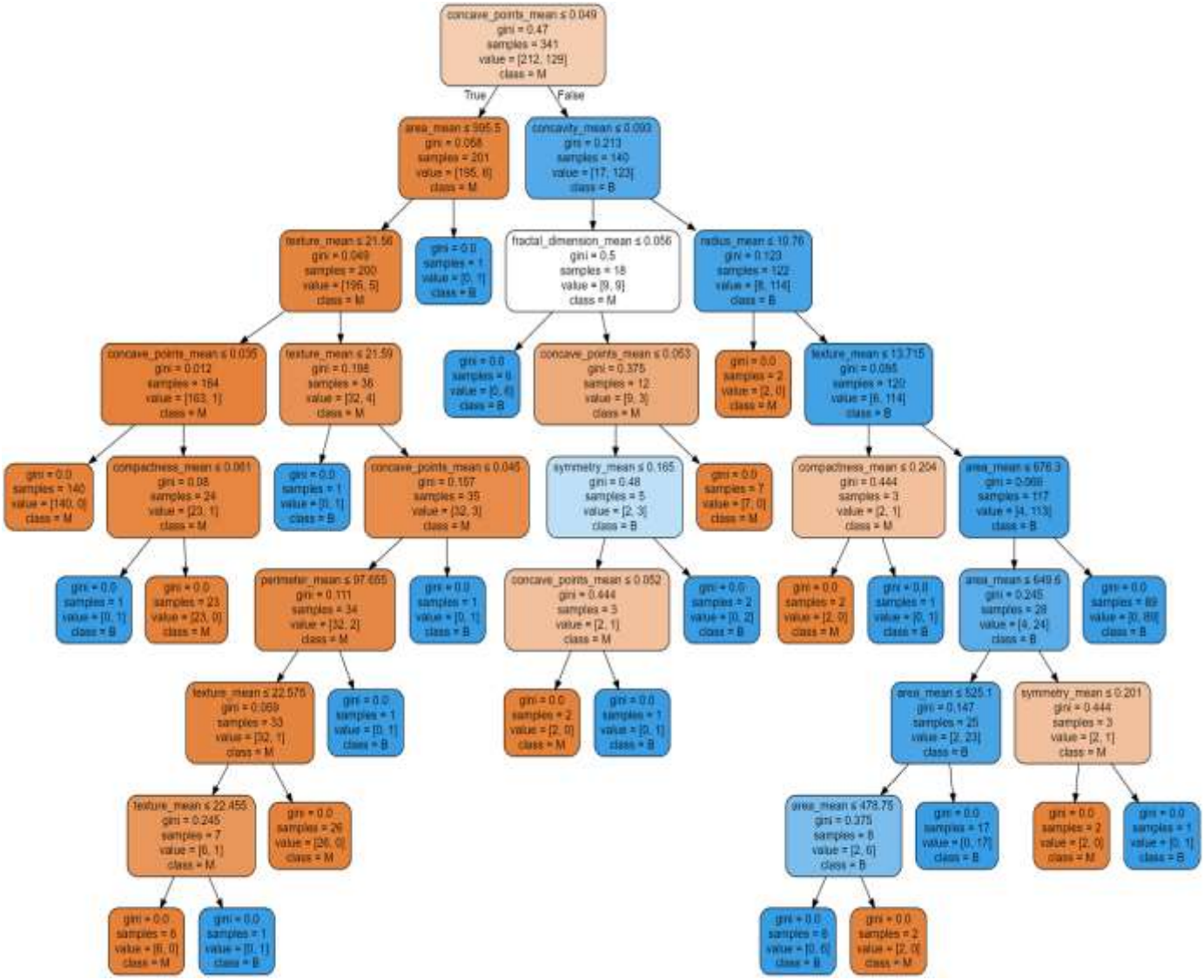
Confusion matrix and results

	Predicted '0"	Predicted '1'	
Actual '0' [	137	8	]
Actual '1' [	17	66	]

The prediction on test data shows that the recall (fraction of relevant instances that are successfully predicted) for the 'Malignant' label was 0.84 and precision (fraction of correctly predicted instances) of 'Malignant' was 0.89. f1 score (harmonic mean of precision and recall) of Malignant was 0.87.

	precision	recall	f1-score	support
0	0.92	0.94	0.93	145
1	0.89	0.86	0.87	83
avg / total	0.91	0.91	0.91	228

Figure 8: Decision Tree Visualization



Parameter Tuning for decision tree

We also applied decision tree on the training data using entropy parameters. The criterion we used was ‘entropy and min_samples_split = 2.

```
clf_1
```

```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```

We obtained better results as per the table below:

Decision tree report – **Table 3**

	precision	recall	f1-score	support
0	0.94	0.95	0.95	145
1	0.91	0.89	0.90	83
avg / total	0.93	0.93	0.93	228

Model 2: K Nearest Neighbors:

The parameters that were used for K Nearest Neighbours as follows:

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=5, p=2,
           weights='uniform')
```

Confusion matrix and results

```
k = 4
Confusion matrix:
 [[142   3]
 [ 19  64]]
             precision    recall  f1-score   support

          0       0.88      0.98      0.93       145
          1       0.96      0.77      0.85        83

avg / total       0.91      0.90      0.90       228


Accuracy for kNN : 90.351%
```

The prediction on test data shows that the recall (fraction of relevant instances that are successfully predicted) for the 'Malignant' label was 0.81 and precision (fraction of correctly predicted instances) of 'Malignant' was 0.91. f1 score (harmonic mean of precision and recall) of Malignant was 0.85.

Furthermore, we perform k-fold classification for the above test and train dataset and the following results were obtained.

```
for k, (train_index, test_index) in enumerate(kf.split(f_data)):
    X_train, X_test = f_data.iloc[train_index], f_data.iloc[test_index]
    y_train, y_test = t_data.iloc[train_index], t_data.iloc[test_index]
    clf.fit(X_train, y_train)
    print "[fold {0}] score: {1:.5f}".format(k, clf.score(X_test, y_test))
```

```
[fold 0] score: 0.76316
[fold 1] score: 0.85088
[fold 2] score: 0.94737
[fold 3] score: 0.90351
[fold 4] score: 0.89381
```

The result for the fold 0 shows a score of only 0.76316. This indicates a possible case of overfitting but we can notice that the values of the folds 1,2,4 and 4 are in the acceptable range.

Model 3: Naïve Bayes

The parameters that were used for K Nearest Neighbours as follows:

```
from sklearn.naive_bayes import GaussianNB
X_train, X_test, y_train, y_test = train_test_split(f_data, t_data, test_size=0.4, random_state=0)
clf = GaussianNB()
clf.fit(X_train, y_train)
prediction = clf.predict(X_test)
```

Confusion matrix and results

	Predicted '0"	Predicted '1'
Actual '0'	[136	9]
Actual '1'	[13	70]

The prediction on test data shows that the recall (fraction of relevant instances that are successfully predicted) for the 'Malignant' label was 0.84 and precision (fraction of correctly predicted instances) of 'Malignant' was 0.89. f1 score (harmonic mean of precision and recall) of Malignant was 0.86.

	precision	recall	f1-score	support
0	0.91	0.94	0.93	145
1	0.89	0.84	0.86	83
avg / total	0.90	0.90	0.90	228

The cross-validation results are better than the kNN model but still shows slight inconsistencies.

```
[fold 0] score: 0.82456
[fold 1] score: 0.90351
[fold 2] score: 0.95614
[fold 3] score: 0.92982
[fold 4] score: 0.92920
```

4. Results

Based on the three models of classification, the accuracy of each model has been provided below

Accuracy for Naive Bayes:	90.315%
Accuracy for K Nearest Neighbours:	89.912%
Accuracy for Decision Tree:	92.105%

The highest accuracy was obtained in the Decision Tree model with an accuracy of 92.105%

5. Discussion

We found that the Decision tree model had the highest accuracy with 92.105%. Since the data is a medical related, precision would be a better measure in order to validate the model. Hence, when we compare models with respect to the precision values, we observe the following pattern:

Decision Tree(with entropy) = 92 **> knn**(k=4) = 91 **> naïve Bayes** = 90

During the k-fold cross validation of the K Nearest Neighbours model we observed that the values were inconsistent which could have been possible overfitting. We can notice that the Decision Tree model is overall more consistent than the Naïve Bayes model and hence can be selected as the best model out of the three.

In order to improve the model, we can remove all the outliers completely to increase the accuracy. Also, we observed that all the three models have almost the same accuracy with very minor difference. Hence, we can try to use an ensemble model for better results.

6. Conclusion

In this project, we investigated the usage of three distinct machine learning classification models for breast cancer diagnosis. It is important to understand that before running any model it is important for to preprocess the data and make sure that the data is balanced and does not have any missing values in order to obtain a better performance. The first model we used was decision tree classifier which gave us a good accuracy of 92.105%. However, we can notice that the decision tree is very big and can be time consuming. One solution to this can be pruning the decision tree in order to get quicker and accurate results. The second model was the K Nearest Neighbors which gave us an accuracy of 89.912%. However, during the k-fold cross validation we saw inconsistent values which may be due to overfitting. Finally, we used the Naïve Bayes model and got the best accuracy of 90.351% and observed Decision Tree to be the best model with a slightly higher accuracy.

7. References

1. Dataset Available at: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
2. Dataset description available at: https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.names
3. Models and validations were refereed from lab tute exercises.
4. Report formatted from: https://www.researchgate.net/publication/311950799_Analysis_of_the_Wisconsin_Breast_Cancer_Dataset_and_Machine_Learning_for_Breast_Cancer_Detection