Machine Learning Engineer Nanodegree

Capstone Project: Breast Cancer Prediction

CL. Chen
March 21st, 2018

I. Definition

(approx. 1-2 pages)

Project Overview

Breast cancer is the most prevalent diseases and common cancer of the women in the worldwide. Recently, the death rate of breast cancer has gradually increased. Approximately 60% of the patients with breast cancers are able to survive for ten more years with early diagnosis coupled with appropriate treatment. It is important to accurately diagnose early to increase the survival rate. With technology continuously progress, data mining and classification method can be used to process a huge amount information of hospital patients. Now, we can use machine learning method to extract the features of cancer cell nuclei image and classify them. It would be helpful to determine whether a given sample appears to be benign ("B") or malignant ("M").

The related academic paper: Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis by Hiba Asri et al. / Procedia Computer Science 83 (2016) 1064 – 1069

link https://doi.org/10.1016/j.procs.2016.04.224

This database is also available through the UW CS ftp server:

ftp ftp.cs.wisc.edu cd math-prog/cpo-dataset/machine-learn/WDBC/

Also can be found on UCI Machine Learning Repository:

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

Problem Statement

In this problem we have to predict the Stage of Breast Cancer M (Malignant) and B (Bengin). The project focus on finding classifier that can accurately determine malignancy of sample. The Breast Cancer (Wisconsin) Diagnosis dataset is used for training. Features 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. In the database, each data is labeled and contains real-valued morphological characteristics of nucleus. I will use several classifier to train and test on these data based on the following described metrics. The benchmark sample of similar case from kaggle will be compared later. The major models used and tested will be supervised learnning which are popular for these kind data analysis.

Metrics

Accuracy and F1 score would be used to compare the result with the benchmark model. Accuracy is the general score of the classifier model performance so I choose it. But this dataset is skewed so I add F1 score to check the performance to avoid "Accuracy Paradox". F1 score provides a metrics that combines precision and sensitivity.

Accuracy: the ratio of how many samples are correctly classified to all samples.

F1 score: the harmonic mean of precision and sensitivity.

Precision: the number of correct positive results divided by the number of all positive results returned by the classifier.

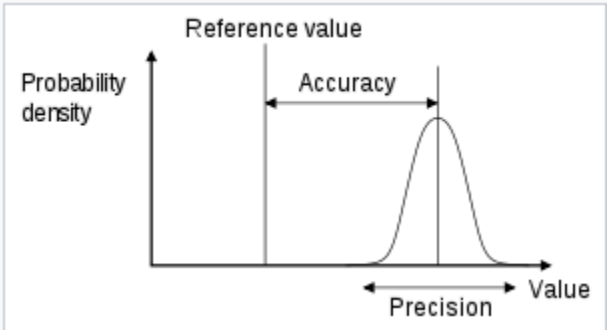
Sensitivity(recall): the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive)

$$Accuracy = \frac{True\ Positive + True\ Negative}{All\ samples}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F = 2 \cdot \frac{Precision \cdot Sensitivity}{Precision + Sensitivity}$$



II. Analysis

(approx. 2-4 pages)

Data Exploration

This database is also available through the UW CS ftp server: ftp ftp.cs.wisc.edu cd math-prog/cpo-dataset/machine-learn/WDBC/. Also can be found on UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29.

The Breast Cancer (Wisconsin) Diagnosis dataset contains the diagnosis and a set of 30 features describing the characteristics of the cell nuclei that present in the digitized image of a

of a fine needle aspirate (FNA) of a breast mass. There are ten features with real value which are computed for each cell nucleus:

Feature name	Description
1. radius	mean of distances from center to points on the perimeter
2. texture	standard deviation of gray-scale values
3. perimeter	
4. area	Number of pixels inside contour + ½ for pixels on perimeter
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

Table1: Features of cell nucleus

Before visualization, we look at the data features. There are 33 columns in the dataframe from Table2 and we can see the all data type of features through Table3. The data type includes int64, object and float64. There are 2 features seemed unnecessary. “Id” and “Unnamed” cannot be used for classification. We should drop them before analysis. “Diagnosis” is a object data, we should transform it to integer value. Dataset contains 357 samples which are labeled as benign and 212 samples as malignant[Figure1]. For the next step, visualization, we can describe the statistic value of data first [Table4] to realize if we have to standardized the data. No unusual properties, missing value and outlier was found in the dataset according to the statistic data of the features.

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	te
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	

5 rows × 33 columns

Table2: First 5 rows of dataframe

```
Data columns (total 33 columns):
id                         569 non-null int64
diagnosis                  569 non-null object
radius_mean                569 non-null float64
texture_mean               569 non-null float64
perimeter_mean             569 non-null float64
area_mean                  569 non-null float64
smoothness_mean            569 non-null float64
compactness_mean           569 non-null float64
concavity_mean             569 non-null float64
concave points_mean        569 non-null float64
symmetry_mean              569 non-null float64
fractal_dimension_mean     569 non-null float64
radius_se                  569 non-null float64
texture_se                 569 non-null float64
perimeter_se               569 non-null float64
area_se                    569 non-null float64
smoothness_se              569 non-null float64
compactness_se             569 non-null float64
concavity_se               569 non-null float64
concave points_se          569 non-null float64
symmetry_se                569 non-null float64
fractal_dimension_se       569 non-null float64
radius_worst               569 non-null float64
texture_worst              569 non-null float64
perimeter_worst            569 non-null float64
area_worst                 569 non-null float64
smoothness_worst           569 non-null float64
compactness_worst          569 non-null float64
concavity_worst            569 non-null float64
concave points_worst       569 non-null float64
symmetry_worst             569 non-null float64
fractal_dimension_worst    569 non-null float64
Unnamed: 32                0 non-null float64
dtypes: float64(31), int64(1), object(1)
```

Table3: Data type of features

```
Number of Benign: 357
Number of Malignant : 212
```

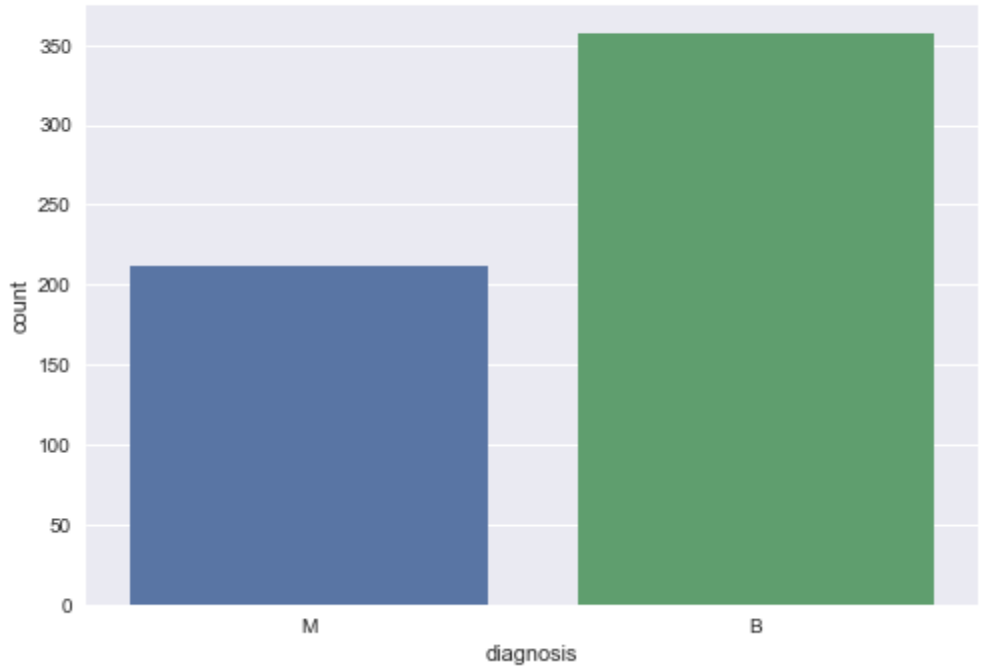


Figure1: Diagnosis distrbution

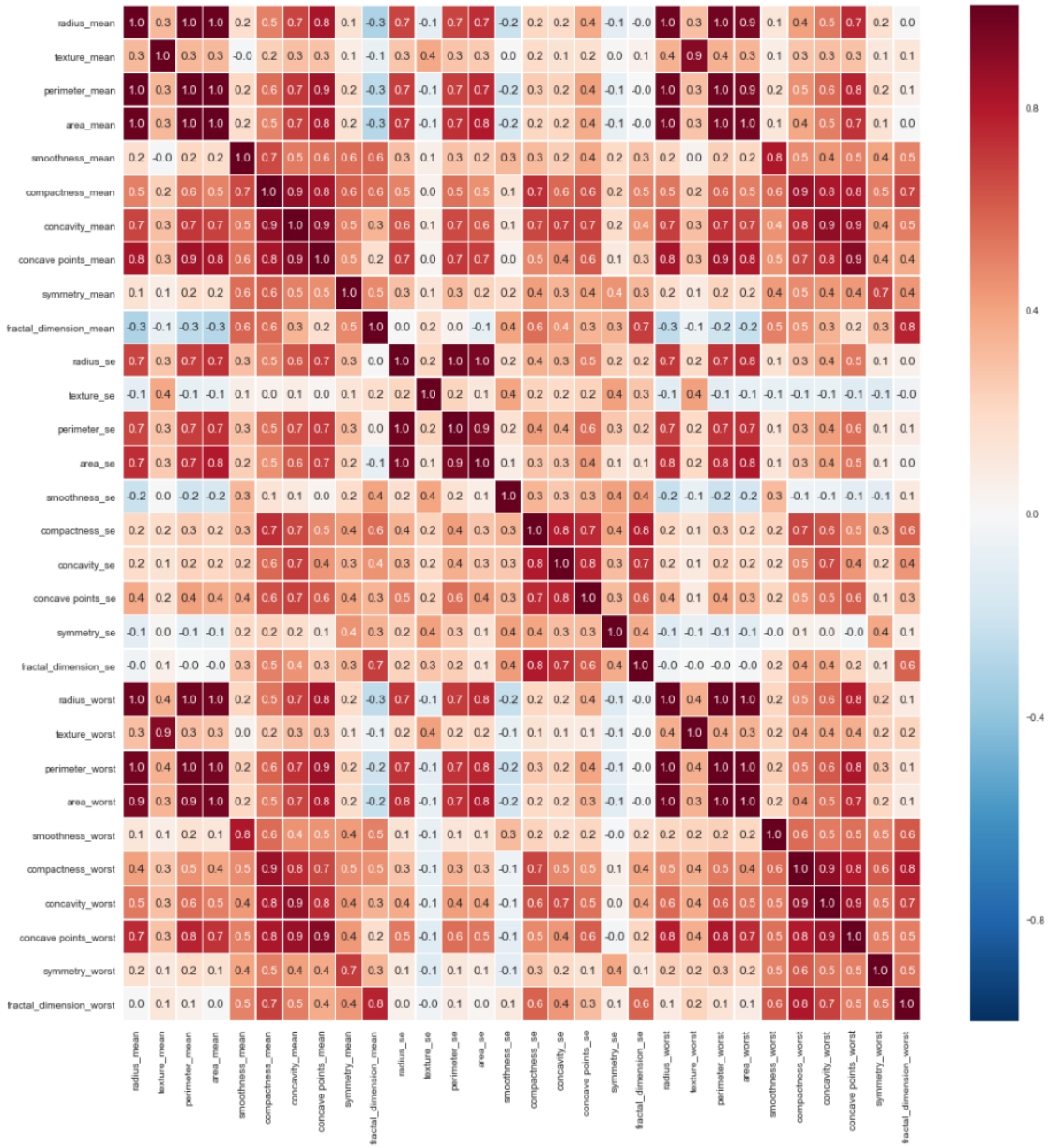
	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fract
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162	
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414	
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000	
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900	
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200	
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700	
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000	

8 rows × 30 columns

Table4: Statistic value of features

Exploratory Visualization

For visualization, we use heatmap to check the correlation of each feature[Figure2]. The correlation map is too complex to read so we divid them into 3 parts, value means, standard deivations, and worst. We can check to see if there is any correlation between these subsets of features[Figure3][Figure4]. The correlated features will be dropped and we choose one of them for analysis as feature selection. The heatmap plot tell us that radius_mean, perimeter_mean and area_mean are correlated with each other so we choose area_mean. Compactness_mean, concavity_mean and concave points_mean are correlated with each other so we choose concavity_mean. For standard error features, radius_se, perimeter_se and area_se are correlated so we choose area_se. Compactness_se, concavity_se and concave points_se are correlated so we choose concavity_se. For worst features, radius_worst, perimeter_worst and area_worst are correlated so we choose area_worst. Compactness_worst, concavity_worst and concave points_worst are correlated so we choose concavity_worst. Finally, overall check the heatmap of all features to see if any correlation between mean,se and



worst group. Texture_mean and texture_worst are correlated so we choose texture_mean. Area_worst and area_mean are correlated so we choose area_mean. If you ask why I choose texture_mean and area_mean, no answer exactly, just I feel mean is more suitable than worst.

Figure2: Correlation plot of all features

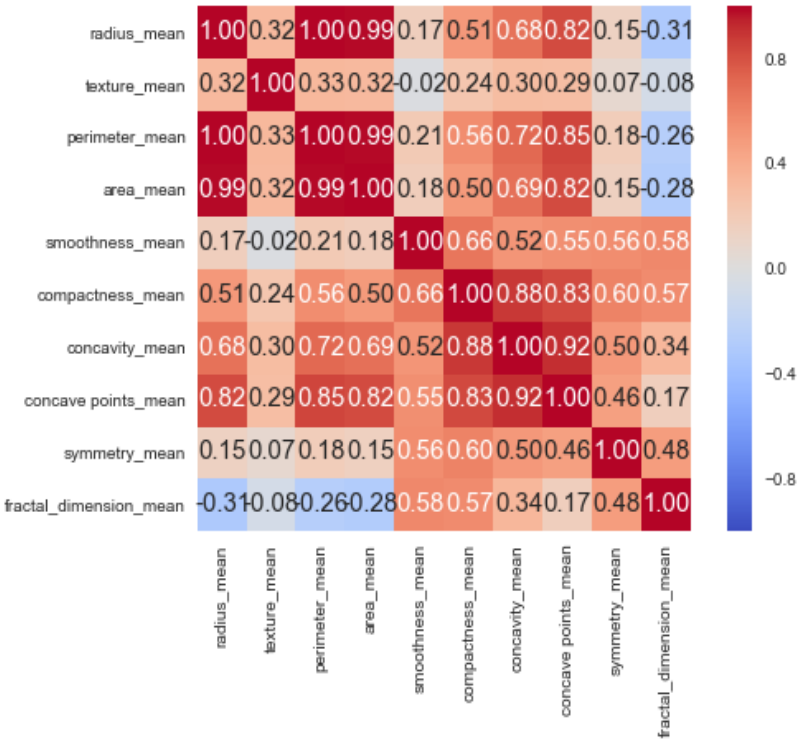


Figure3: Correlation plot of features_mean

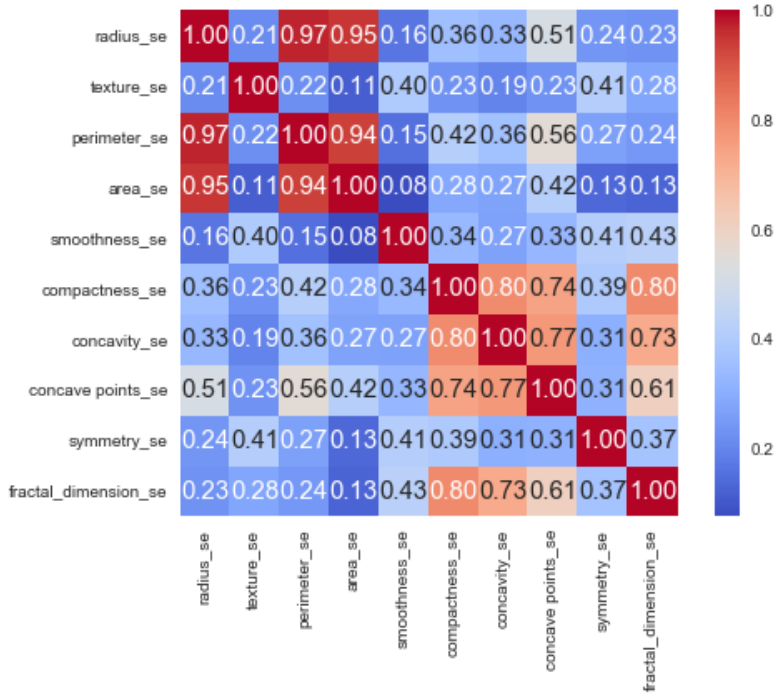


Figure4: Correlation plot of features_se

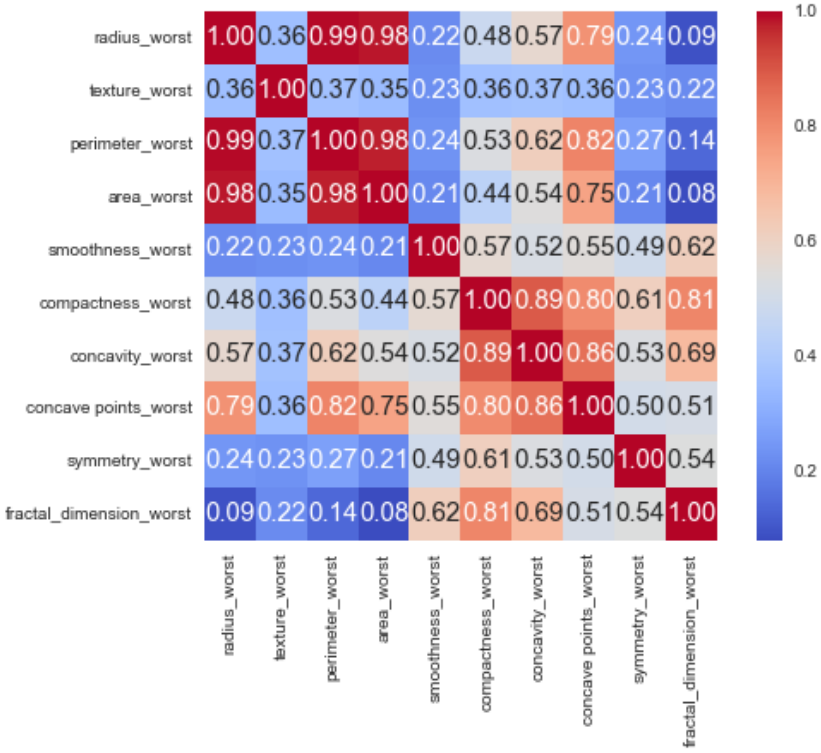


Figure5: Correlation plot of features_worst

Algorithms and Techniques

We will use supervised learning to solve the problem because all the dataset are labeled. In this case, all samples will be divided into two groups (Malignant and Bengin) which have the binary result. Some binary calssifier will be able to used like LR (logistic regression), SVM, Random Forest.... All dataset will be separated into 70% training and 30% test set randomly then compute their accuracy and F1 score. Feature selection will be performed to reduce the numbers of dimensions and to increase its accuracy. I choose Logistic Regression, Adaboost,Random Forest and Support Vector Machine as my machine learning techniques because they have been well demonstrated good performance in many kaggle pratices. They also represent different kinds of classifier models of supervised learning like linear model, decision tree,ensemble and kernelized model.

Logistic Regression(LR): Logistic Regression is widly used for binary classification like (0,1). The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features).

AdaBoost: AdaBoost is an ensemble method which uses classifier repeatdly on same database but uses different weights on diffcult samples. It is less susceptible to the overfitting problem than other learning algorithms and sensitive to noisy data and outliers. Using Adaboost as classifier should habe the same advantage compared with Decision tree but with the better classification accuracy.

Random Forest: Random forest is another ensemble method based on decision trees. It split data into sub-samples, trains decision tree classifiers on each sub-sample and averages prediction of each classifier. Splitting dataset causes higher bias but it is compensated by large decrease in variance.

Support Vector Machine(SVM): SVM can be use for classification and regression analysis. It constructs a hyperplane or set of hyperplanes in a high dimensional space. It can use various kernels for different application including of text categorization, image classification and hand

written digit recognition. The disavantage of SVM is that it tend to perform poorly when number of features is much larger than samples but is not the case with our database.

Benchmark

This dataset was analyzed in the public Kaggle. I choose Buddhini W's 'Breast cancer prediction' as my benchmark model. In her analysis, the best model to be used for diagnosing breast cancer as found is the Random Forest model with the top 5 predictors, 'concave points_mean','area_mean','radius_mean','perimeter_mean','concavity_mean'. It gives a prediction accuracy of ~95% and a cross-validation score ~ 93% for the test data set.

Kaggle, Breast cancer prediction, Buddhini Waidyawansa (12-03-2016)

https://www.kaggle.com/buddhiniw/breast-cancer-prediction

III. Methodology

(approx. 3-5 pages)

Data Preprocessing

Data contains high correlated features that can be removed without information loss. The feature selection is through visualization to decide and 14 features are dropped. In order to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges, we will scale the data before using SVM classification. “id” and “Unnamed” features which are unnecessary for analysis are dropped at the beginning. Based on previous heatmap correlation visualization, I drop the correlated features and keep the important features for analysis as below:

Selected features:

[texture_mean, area_mean, smoothness_mean, concavity_mean, symmetry_mean, fractal_dimension_mean, texture_se, area_se, smoothness_se, concavity_se, symmetry_se, fractal_dimension_se, smoothness_worst, concavity_worst, symmetry_worst, fractal_dimension_worst]

Implementation

Project is implemented by sklearn framwork in Python2.7 jupyter notebook. The first step, through the visualization we drop the 14 correlated features and choose the 15 features for analysis. The second, dataset was split to train and test set with 70% and 30% ratio. I use the following classifier to train and predict and then check their accuracy score, F1 score, confusion matrix and kfold validation result.

Random Forest (random_state=43)

Logistic Regression

Ada Boost with base_estimator =DecisionTreeClassifier(random_state=42, max_depth=1)

Support Vector Machine (kernel='rbf', C=10, gamma=0.1)

k-fold validation with n_folds = 5,shuffle = False

I didn't scale the dataset before SVM used at the beginning and found the accuracy score and F1 score very low. I will try to scale the x_train and x_test to improve the model at the refinement.

Model	Accuracy score	F1 score	Kfold validation
Random Forest	0.9532163743	0.935483871	0.9404016507
Logistic Regression	0.9649122807	0.953125	0.9316912972
Ada Boost	0.9532163743	0.9365079365	0.96302718
Support Vector Machine	0.6257309942	0.0857142857	0.6309664247

Table5. Classification score report

Refinement

In order to improve the SVM score, I scaled x_train and x_test data before running model. Scaling transforms the data to center it by removing the mean value of each feature, then scale it by dividing non-constant features by their standard deviation. After data scaled, the accuracy score was improved from 0.625 to 0.959 and F1 score was improved from 0.085 to 0.941 but 5-fold validation score was still low. The 5-fold validation is to partition data into 5 complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set). 5 iterations of cross-validation are used. In order to improve the 5-fold CV score, I tried to change C and gamma but not workable obviously. I think I may use the unsuitable kernel so I changed the kernel from 'rbf' to 'linear'. After kernel changed, the accuracy score and F1 score were very high and just slight overfitting found in the confusion matrix. The kfold validation score was improved very much from 0.63 to 0.94.

Model	Accuracy score	F1 score	Kfold validation
Support Vector Machine(kernel = rbf)	0.6257309942	0.0857142857	0.6309664247
Support Vector Machine (scaled data)	0.9590643275	0.9411764706	0.6309664247
Support Vector Machine(kernel = linear)	0.9824561404	0.9756097561	0.9419713076

Table6. SVM model performance refinement

IV. Results

(approx. 2-3 pages)

Model Evaluation and Validation

Among all the test model (Random Forest, Logistic Regression,Ada Boost, SupportVector Machine), they all get the good performance of the accuracy score >95%. Furthermore, comparing the F1 score of them, Logistic and SVM also get the score >95%. Finally, I choose Support Vector Machine(SVM) using linear kernel as the best model. It has the highest F1 score 97% and higher 5-fold CV on the test data. Its accuracy, F1 and 5-fold CV score are almost the best among these models.

The features were selected by visualization method by using heatmap. Features are:

[texture_mean, area_mean, smoothness_mean, concavity_mean, symmetry_mean, fractal_dimension_mean, texture_se, area_se, smoothness_se, concavity_se, symmetry_se, fractal_dimension_se, smoothness_worst, concavity_worst, symmetry_worst, fractal_dimension_worst]

Final Model: SVM

Parameters: kernel='linear', C=1, gamma=0.1

5-fold cross-validation was used to avoid overfitting and be less sensitive to data variance.

The dataset was split into training ans testing dataset with 70% and 30%. Testing dataset was used to verify robustness of classifiers. Higher score (Accuracy,F1) on testing dataset proves the model was the robust solution to the problem. The dataset was also divided into 5-fold with cross-validation to elimate the dataset variance impact.

Model	Accuracy score	F1 score	Kfold validation
Random Forest	0.9532163743	0.935483871	0.9404016507
Logistic Regression	0.9649122807	0.953125	0.9316912972
Ada Boost	0.9532163743	0.9365079365	0.96302718
Support Vector Machine(kernel = rbf)	0.6257309942	0.0857142857	0.6309664247
Support Vector Machine (scaled data)	0.9590643275	0.9411764706	0.6309664247
Support Vector Machine(kernel = linear)	0.9824561404	0.9756097561	0.9419713076

Table7. Score summary of different model

Justification

Compared with benchmark model using Random Forest with the top5 predictors, I used Random Forest, Logistic Regression, AdaBoost and SVM with the selected features. Accuracy and Kfold validation score of my Random Forest, Logistic Regression, AdaBoost and SVM models are a little better than benchmark model. Benchmark model just used 5 features may induced more overfitting than my model. I think through visualization to find out the best predictors is a good approach to improve model's accuracy.

Model	Accuracy score	F1 score	Kfold validation
Benchmark	0.9472		0.9187
Random Forest	0.9532163743	0.935483871	0.9404016507
Logistic Regression	0.9649122807	0.953125	0.9316912972
Ada Boost	0.9532163743	0.9365079365	0.96302718
Support Vector Machine(kernel = linear)	0.9824561404	0.9756097561	0.9419713076

Table8. Compared with benchmark model

V. Conclusion

(approx. 1-2 pages)

Free-Form Visualization

Visualization shows SVM(kernel=linear) has 0 false positive and 3 false negative which has the lowest FP. Logistic Regression has 4 false positive and 2 false negative which has the lowest FN. It indicates that although SVM has the highest precision but Logistic Regression has the highest sensitivity. SVM has almost comparable sensitivity with LR and more better precision than LR so I still choose SVM as the final model for this project.

Model	TP	FP	FN	TN	Precision	Sensitivity
Random Forest	105	3	5	58	0.9722222222	0.9545454545
Logistic Regression	104	4	2	61	0.962962963	0.9811320755
Ada Boost	104	4	4	59	0.962962963	0.962962963
Support Vector Machine(kernel = rbf)	104	4	60	3	0.962962963	0.6341463415
Support Vector Machine (scaled data)	108	0	7	56	1	0.9391304348
Support Vector Machine(kernel = linear)	108	0	3	60	1	0.972972973

Table9. Confusion matrix summary

Reflection

In this problem we have to predict the Stage of Breast Cancer M (Malignant) and B (Bengin). The project focus on finding classifier that can accurately determine malignancy of sample. We use visualization method to select the important features and then trensfer the output to binary mode(0,1) before analysis. All dataset were divided into training and test data. Random Forest, Logistic Regression, Ada Boost and SVM model were used as classifier and we checked the accuracy, F1 and 5-fold cross-validation score to decide the final model. Finally, we got the better result than benchmark model.

The interesting thing is that unscaled dataset and unsuitable kernel caused SVM model failed. After data preprocessing and kernel changed, SVM got the great performance than before.

The difficult aspect of this project is the feature delection. Many correlated features should be dropped before analysis to improve the performance. There are many mothod of feature selection but I choose visualization. This maybe not a good method of high efficiency. It took me much time than other parts.

Improvement

I think my design can be improved by these aspect:

Feature selection: Other visualization method can be used like violin plot and swarm plot which are popular used in kaggle instead of heatmap for binary group of result. PCA could be also another method for grouping highly correlated features instead of visualization.

Parameter tuning: I didn't tune SVM parameters in this project. The default setting (C=10, gamma=0.1) may not be the best one. Parameter tuning should improve model performance.