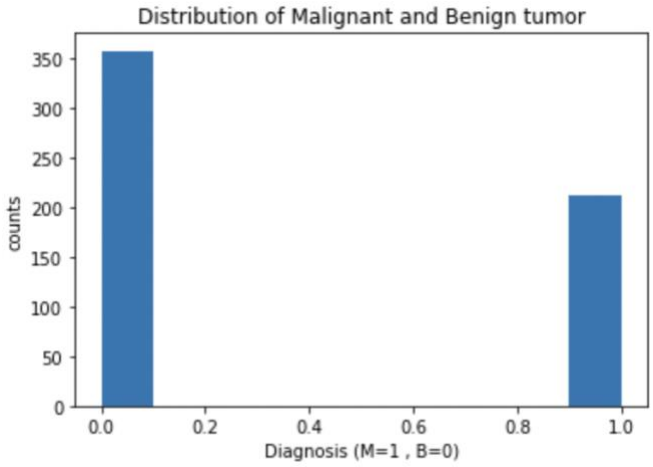
Topic: Assignment 3 Feature Engineering with Logistic Regression

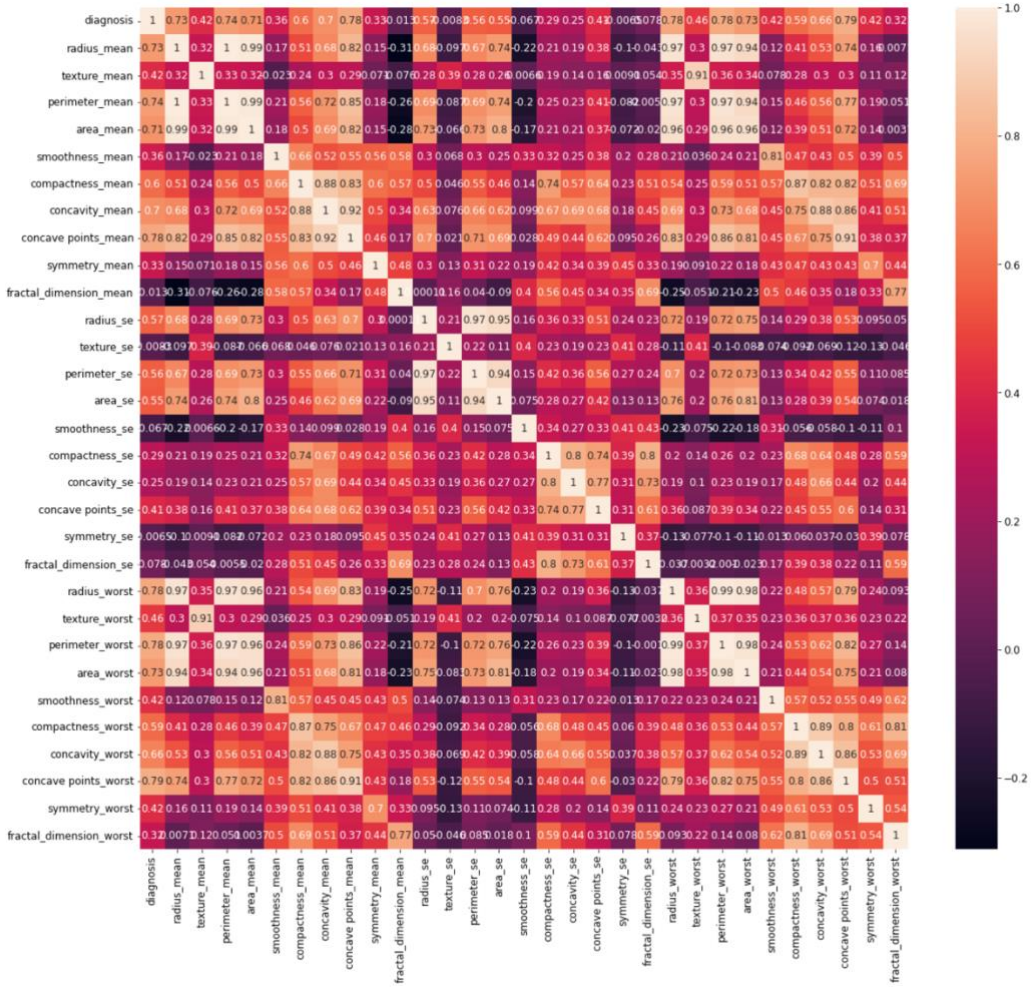
This analysis aims to observe which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in model selection.

Step 1: Data Exploration

At first, I performed an exploratory analysis of the data. I checked if there was any missing data and removed unnecessary columns (e.g. ‘Unamed’ data). After cleaning the data, I dug into the data and found any correlations between the target data and the features. From the plot below, we can tell that breast cancer tumor types that are diagnosed as benign are more than those diagnosed as malignant.



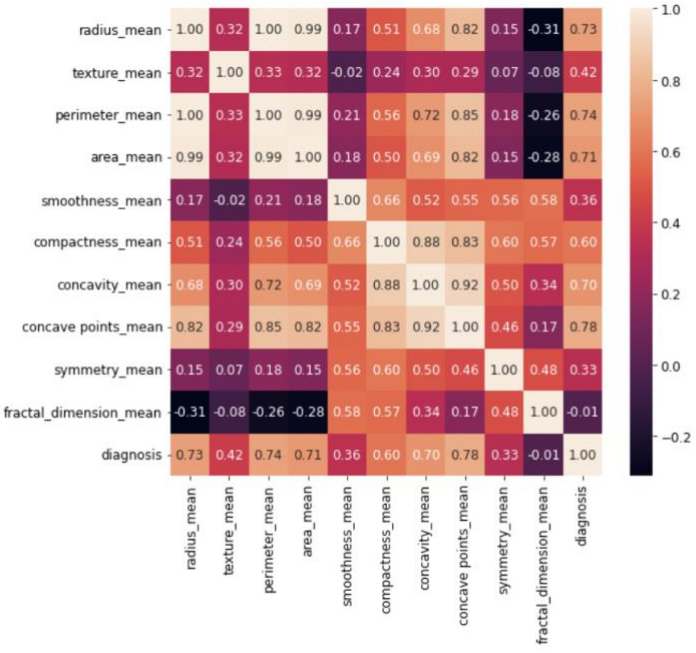
The plot below shows the correlation between each feature. Regarding feature engineering, finding the relation between each feature is essential. As seen in the confusion matrix, some variables are not very closely related, so we deleted them.



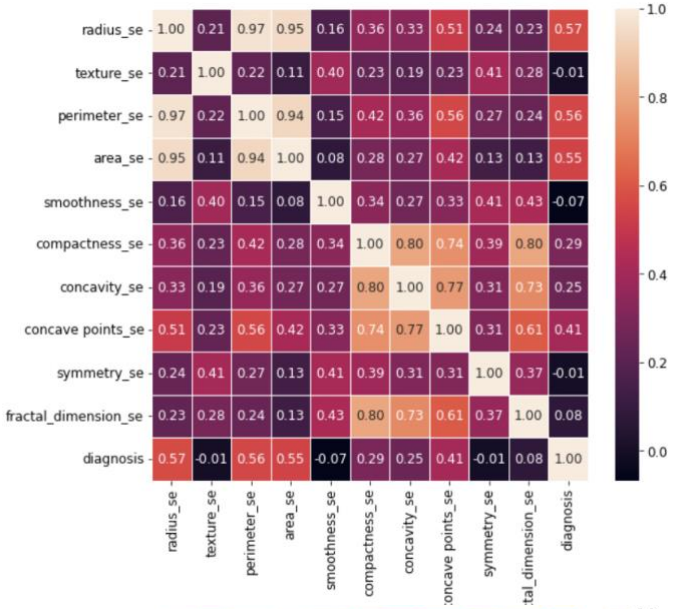
Step 2: Feature selection

I started by plotting the three main categories (means, SE, worst).

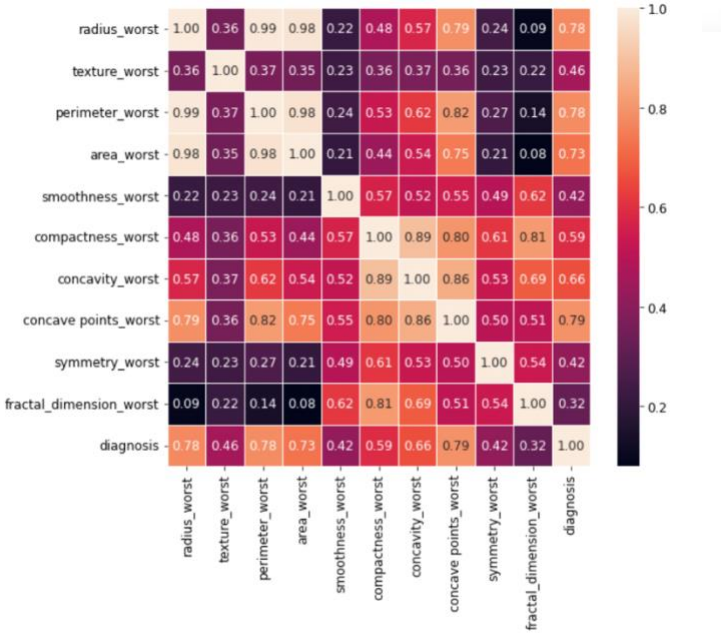
1. Mean:



2. SE



3. Worst



Observations:

Features that could be used in classification of the cancer
(I used the ones which have correlation coefficient>0.7):

['radius_mean', 'perimeter_mean', 'area_mean','concave points_mean', 'radius_worst', 'perimeter_worst', 'area_worst','concave points_worst']

1.feature_means: radius_mean,perimeter_mean,area_mean,concave points_mean
2.feature_worst: radius_worst,perimeter_worst,area_worst,concave points_worst

Step3: Standardize our data, perform our train/test split, then develop a Logistic Regression Model

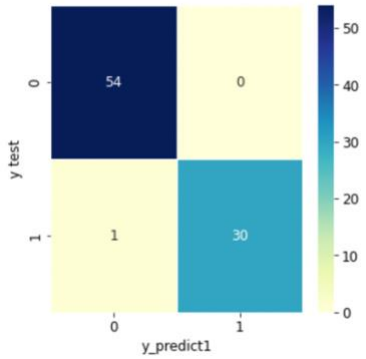
I tried five different feature sets to build logistic regression models:

- NO Feature Engineering
  - 1. use all features
- Feature Engineering
  - 2. use only 'mean' related features
  - 3. use only 'worst' related features
  - 4. use all the features which have correlation coefficient > 0.7 with the 'diagnosis'
  - 5. apply SelectKBest class to extract the top 10 best features

Result for model1

This is the confusion matrix for the model1. I used all the features in the dataset, and we can see that 54 Benign tumors and 30 Malignant tumors were accurately predicted, 1 tumor was presenting a faulty predictions..

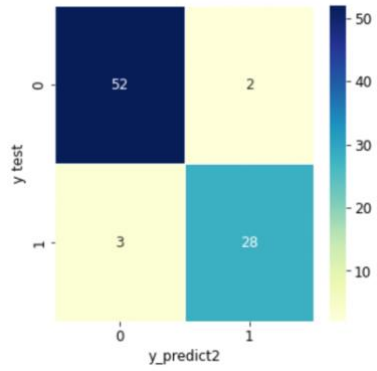
The F1-score is very high (>0.9) which means that this is a model that perfectly classifies each observation into the correct class.



	precision	recall	f1-score	support
0	0.98	1.00	0.99	54
1	1.00	0.97	0.98	31
accuracy			0.99	85
macro avg	0.99	0.98	0.99	85
weighted avg	0.99	0.99	0.99	85

Accuracy of the Logistic Regression Model is: 0.9882352941176471

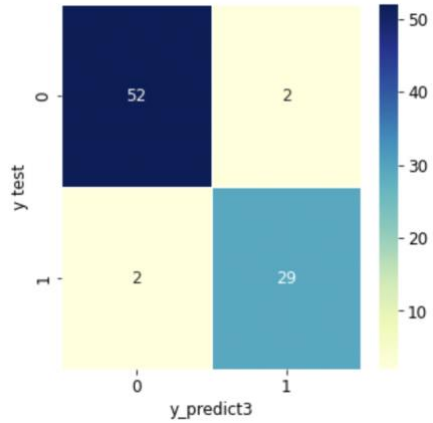
Result for model2



	precision	recall	f1-score	support
0	0.95	0.96	0.95	54
1	0.93	0.90	0.92	31
accuracy			0.94	85
macro avg	0.94	0.93	0.94	85
weighted avg	0.94	0.94	0.94	85

Accuracy of the Logistic Regression Model is: 0.9411764705882353

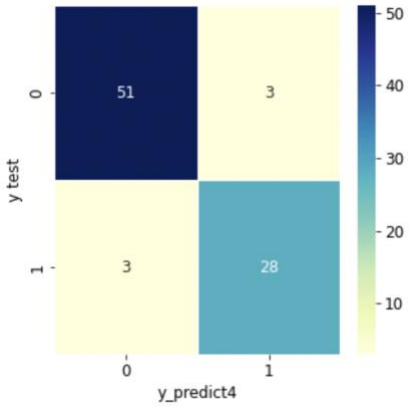
Result for model 3



	precision	recall	f1-score	support
0	0.96	0.96	0.96	54
1	0.94	0.94	0.94	31
accuracy			0.95	85
macro avg	0.95	0.95	0.95	85
weighted avg	0.95	0.95	0.95	85

Accuracy of the Logistic Regression Model is: 0.9529411764705882

Result for model 4



	precision	recall	f1-score	support
0	0.94	0.94	0.94	54
1	0.90	0.90	0.90	31
accuracy			0.93	85
macro avg	0.92	0.92	0.92	85
weighted avg	0.93	0.93	0.93	85

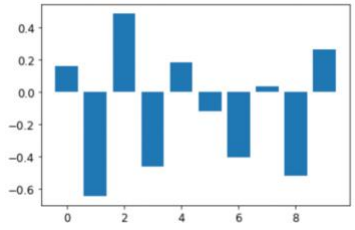
Accuracy of the Logistic Regression Model is: 0.9294117647058824

Result for model 5

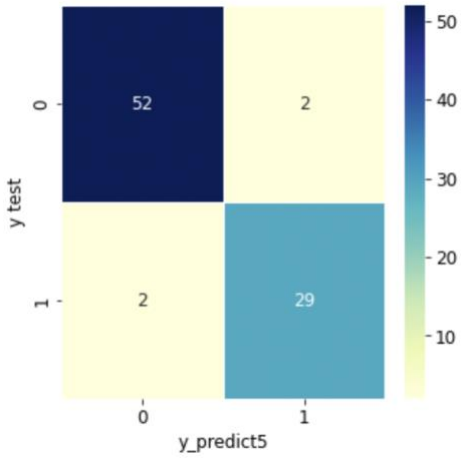
A bar chart is created for the feature importance scores. Base on the accuracy of the Logistic Regression Model, model one has the highest accuracy rate. The larger the coefficient is (in both positive and negative direction), the more influence it has on a prediction.

```
# summarize feature importance
for i,v in enumerate(importance):
    print('Feature: %0d, Score: %.5f' % (i,v))
# plot feature importance
pyplot.bar([x for x in range(len(importance))], impo
pyplot.show()
```

```
Feature: 0, Score: 0.16320
Feature: 1, Score: -0.64301
Feature: 2, Score: 0.48497
Feature: 3, Score: -0.46190
Feature: 4, Score: 0.18432
Feature: 5, Score: -0.11978
Feature: 6, Score: -0.40602
Feature: 7, Score: 0.03772
Feature: 8, Score: -0.51785
Feature: 9, Score: 0.26540
```



	Specs	Score
23	area_worst	112598.431564
3	area_mean	53991.655924
13	area_se	8758.504705
22	perimeter_worst	3665.035416
2	perimeter_mean	2011.102864
20	radius_worst	491.689157
0	radius_mean	266.104917
12	perimeter_se	250.571896
21	texture_worst	174.449400
1	texture_mean	93.897508

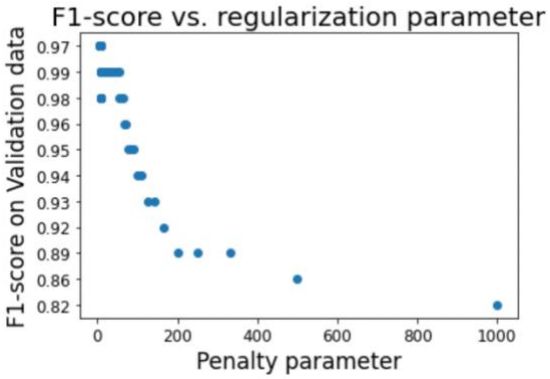


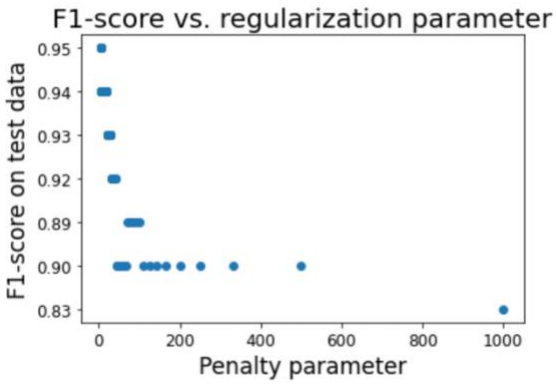
	precision	recall	f1-score	support
0	0.96	0.96	0.96	54
1	0.94	0.94	0.94	31
accuracy			0.95	85
macro avg	0.95	0.95	0.95	85
weighted avg	0.95	0.95	0.95	85

Accuracy of the Logistic Regression Model is: 0.9529411764705882

Evaluating on Validation Set

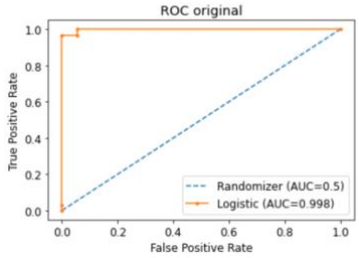
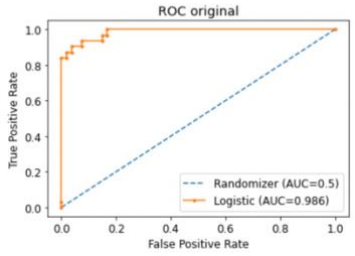
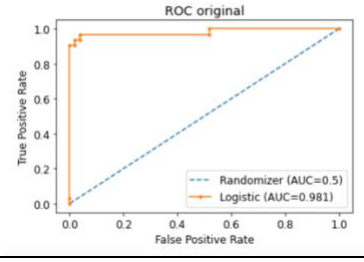
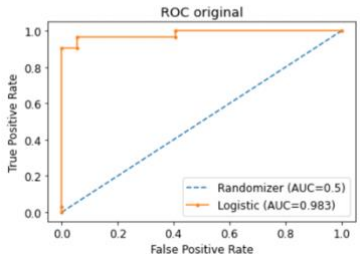
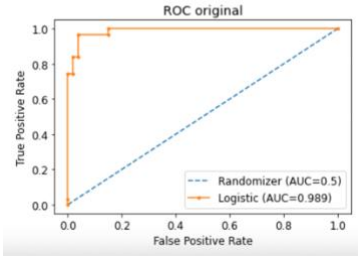
As we can see from the plot(Take model 5 as an example), while the Penalty parameter iterates, F1 score moves to the upper left of the plot, showing that the F1 scores get higher and the performance gets better.





Error Analysis Using ROC and AUC

ROC tells us how good the model is for distinguishing the given classes, in terms of the predicted probability. This classifier gave a curve close to the top-left corner which indicates a good performance. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes. Scores and the ROC curve as shown below.

Model1	Randomized Predictions: ROC AUC=0.500 Logistic Regression Classifier: ROC AUC=0.998 
Model2	Randomized Predictions: ROC AUC=0.500 Logistic Regression Classifier: ROC AUC=0.986 
Model3	Randomized Predictions: ROC AUC=0.500 Logistic Regression Classifier: ROC AUC=0.981 
Model4	Randomized Predictions: ROC AUC=0.500 Logistic Regression Classifier: ROC AUC=0.983 
Model5	Randomized Predictions: ROC AUC=0.500 Logistic Regression Classifier: ROC AUC=0.989 

Summary

Model one has the best accuracy among all the models, the area under the ROC curve is the largest. Model five has the best accuracy among all the feature selected models.