TERM

CCA 2021

PROJECT

컴퓨터공학부 2017-18538 황선영

컴퓨터공학부 2017-12734 윤종선

CONTENTS

01 Problem Definition

02 Data

03 Method we used

04 Results

05 Discussion of our work

01 & 02

Problem Definition & Data

Problem Definition: Curse of dimensionality

Sparse raw data

Computationally intractable

→ How to deal with high-dimensional data? Dimension Reduction

Breast cancer dataset

569 data (2 class: 212 malignant, 357 benign)

30 numerical features

provided with sklearn.datasets package

03

Method we used

Procedure

1. Apply the PCA function into breast cancer data for dimension reduction

2. Fit Ridge / Lasso / Logistic Regression to the data

3. Predict the result (malignant or benign) for classification

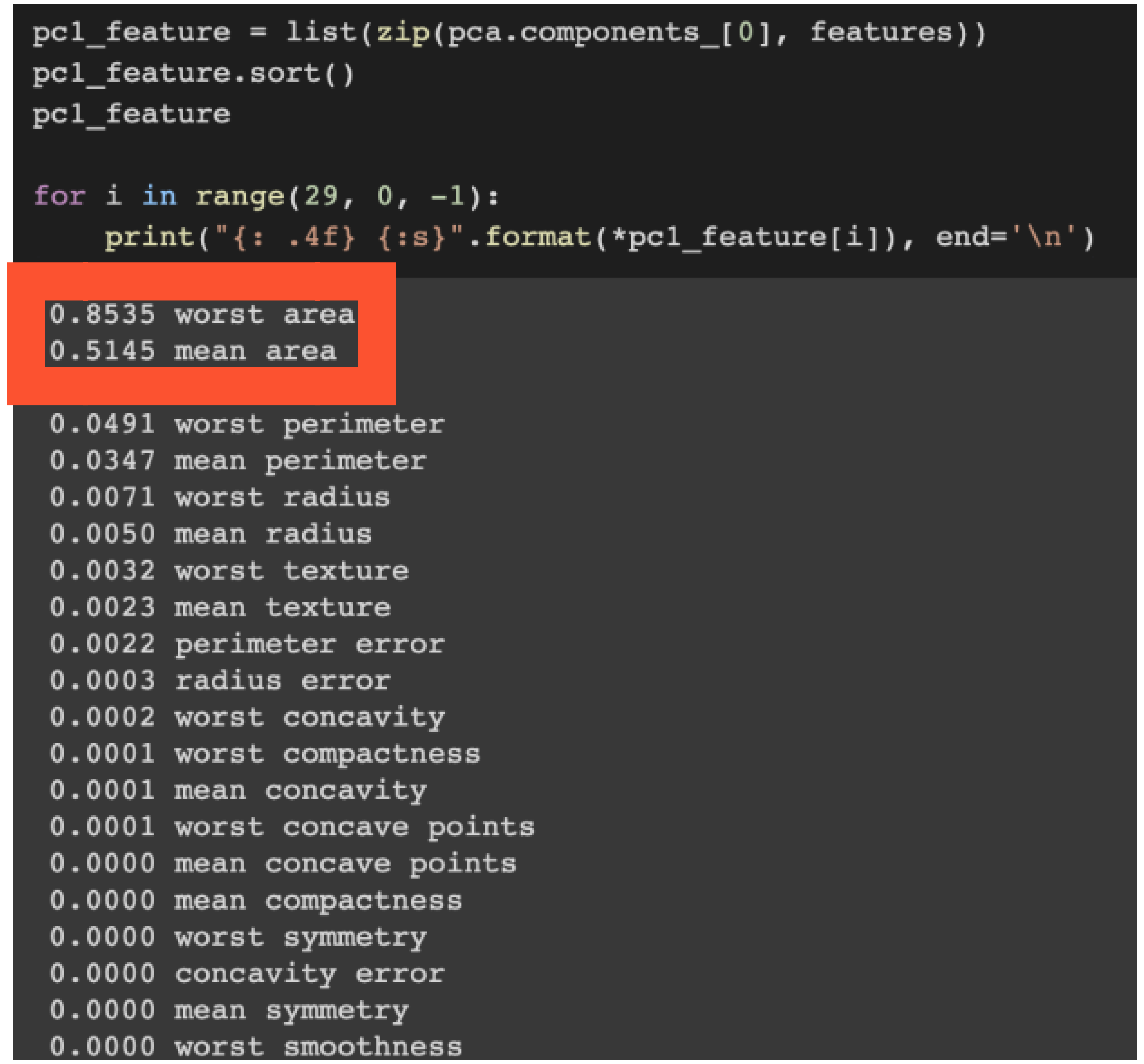
4. plot the data and linear regression

5. Compare real data and prediction result

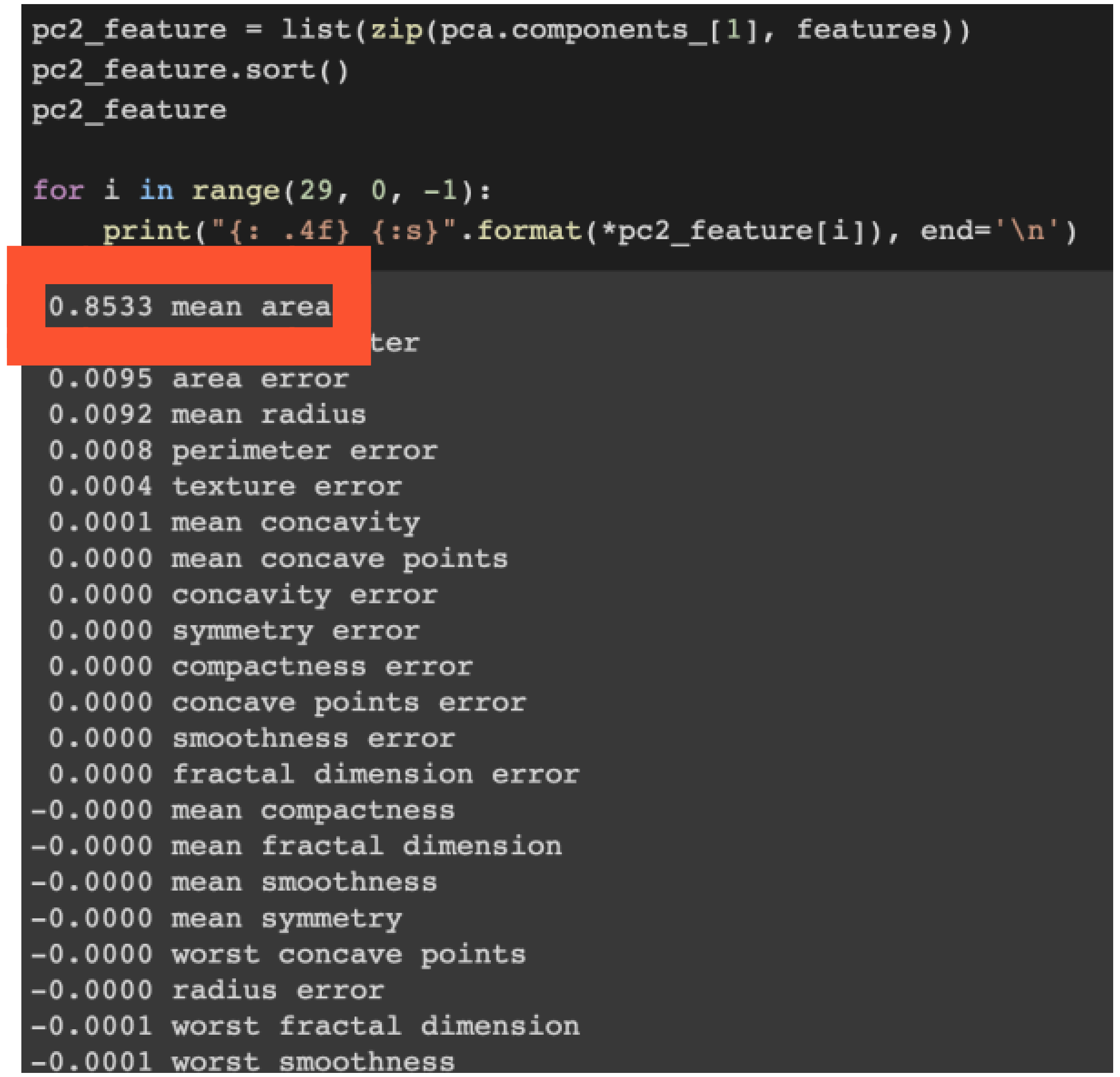
6. Analize result (prediction accuracy, reason of misprediction)

04

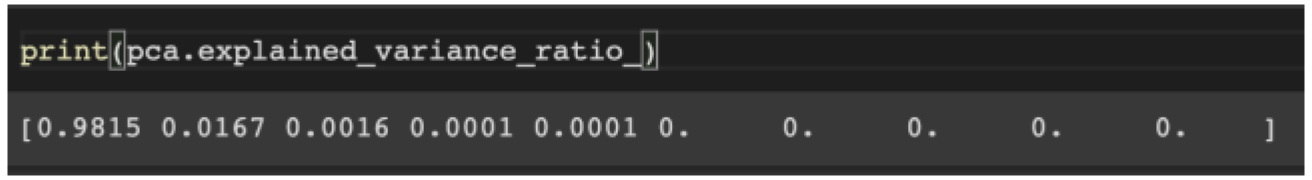
Results: PCA



```
pc1_feature = list(zip(pca.components_[0], features))
pc1_feature.sort()
pc1_feature

for i in range(29, 0, -1):
    print("{: .4f} {:s}".format(*pc1_feature[i]), end='\n')

 0.8535 worst area
 0.5145 mean area

 0.0491 worst perimeter
 0.0347 mean perimeter
 0.0071 worst radius
 0.0050 mean radius
 0.0032 worst texture
 0.0023 mean texture
 0.0022 perimeter error
 0.0003 radius error
 0.0002 worst concavity
 0.0001 worst compactness
 0.0001 mean concavity
 0.0001 worst concave points
 0.0000 mean concave points
 0.0000 mean compactness
 0.0000 worst symmetry
 0.0000 concavity error
 0.0000 mean symmetry
 0.0000 worst smoothness
```



```
pc2_feature = list(zip(pca.components_[1], features))
pc2_feature.sort()
pc2_feature

for i in range(29, 0, -1):
    print("{: .4f} {:s}".format(*pc2_feature[i]), end='\n')

 0.8533 mean area
                  ter
 0.0095 area error
 0.0092 mean radius
 0.0008 perimeter error
 0.0004 texture error
 0.0001 mean concavity
 0.0000 mean concave points
 0.0000 concavity error
 0.0000 symmetry error
 0.0000 compactness error
 0.0000 concave points error
 0.0000 smoothness error
 0.0000 fractal dimension error
-0.0000 mean compactness
-0.0000 mean fractal dimension
-0.0000 mean smoothness
-0.0000 mean symmetry
-0.0000 worst concave points
-0.0000 radius error
-0.0001 worst fractal dimension
-0.0001 worst smoothness
```

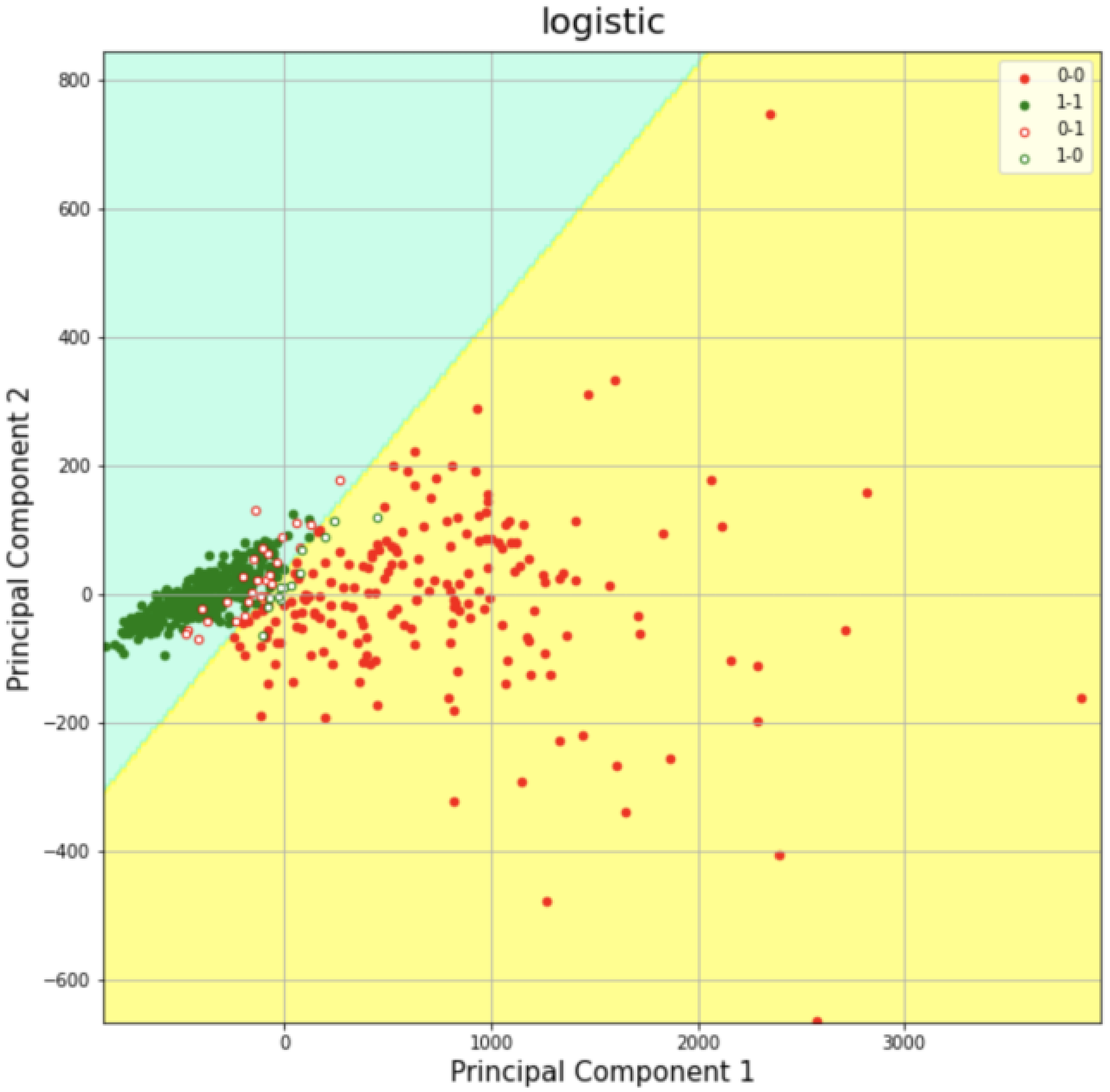


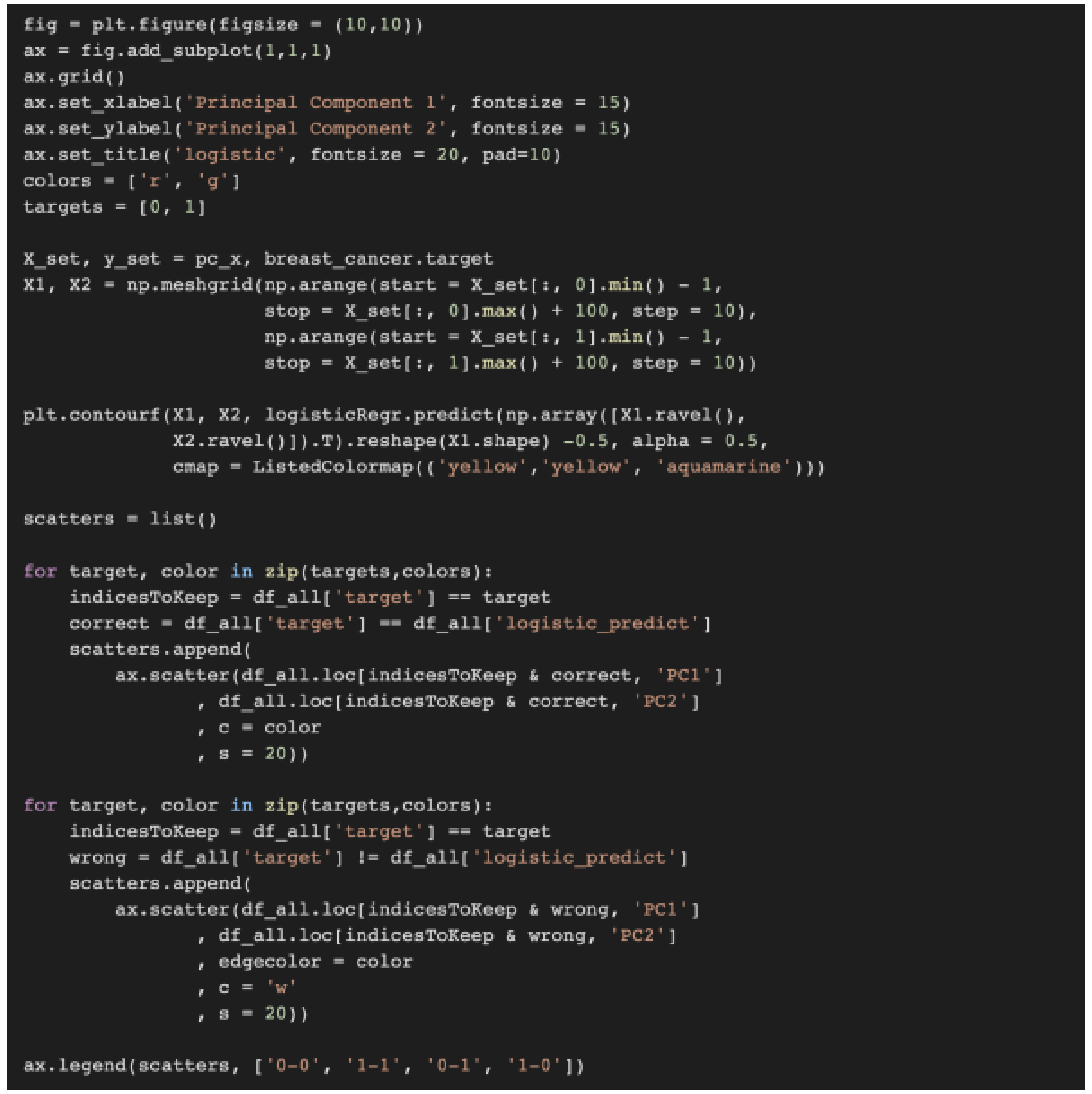
```
print(pca.explained_variance_ratio_)

[0.9815 0.0167 0.0016 0.0001 0.0001 0.      0.      0.      0.      0.     ]
```

n_components = 2 → explaination_ratio < 0.99, easy to visualization (2D)

04

Results: Logistic Regression





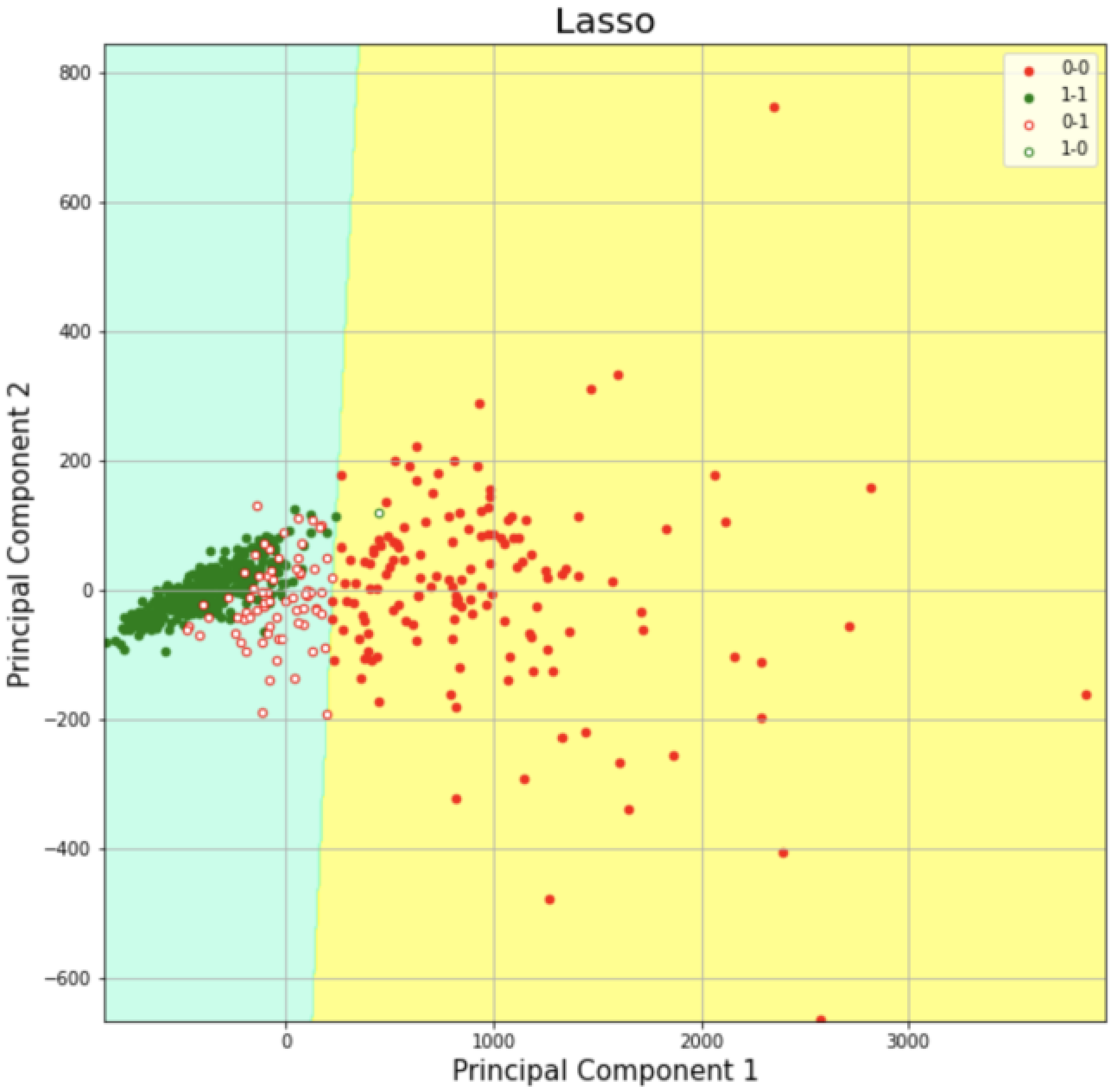
```
fig = plt.figure(figsize = (10,10))
ax = fig.add_subplot(1,1,1)
ax.grid()
ax.set_xlabel('Principal Component 1', fontsize = 15)
ax.set_ylabel('Principal Component 2', fontsize = 15)
ax.set_title('logistic', fontsize = 20, pad=10)
colors = ['r', 'g']
targets = [0, 1]

X_set, y_set = pc_x, breast_cancer.target
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1,
                     stop = X_set[:, 0].max() + 100, step = 10),
                     np.arange(start = X_set[:, 1].min() - 1,
                     stop = X_set[:, 1].max() + 100, step = 10))

plt.contourf(X1, X2, logisticRegr.predict(np.array([X1.ravel(),
             X2.ravel()]).T).reshape(X1.shape) -0.5, alpha = 0.5,
             cmap = ListedColormap(('yellow','yellow', 'aquamarine')))

scatters = list()

for target, color in zip(targets,colors):
    indicesToKeep = df_all['target'] == target
    correct = df_all['target'] == df_all['logistic_predict']
    scatters.append(
        ax.scatter(df_all.loc[indicesToKeep & correct, 'PC1']
               , df_all.loc[indicesToKeep & correct, 'PC2']
               , c = color
               , s = 20))

for target, color in zip(targets,colors):
    indicesToKeep = df_all['target'] == target
    wrong = df_all['target'] != df_all['logistic_predict']
    scatters.append(
        ax.scatter(df_all.loc[indicesToKeep & wrong, 'PC1']
               , df_all.loc[indicesToKeep & wrong, 'PC2']
               , edgecolor = color
               , c = 'w'
               , s = 20))

ax.legend(scatters, ['0-0', '1-1', '0-1', '1-0'])
```

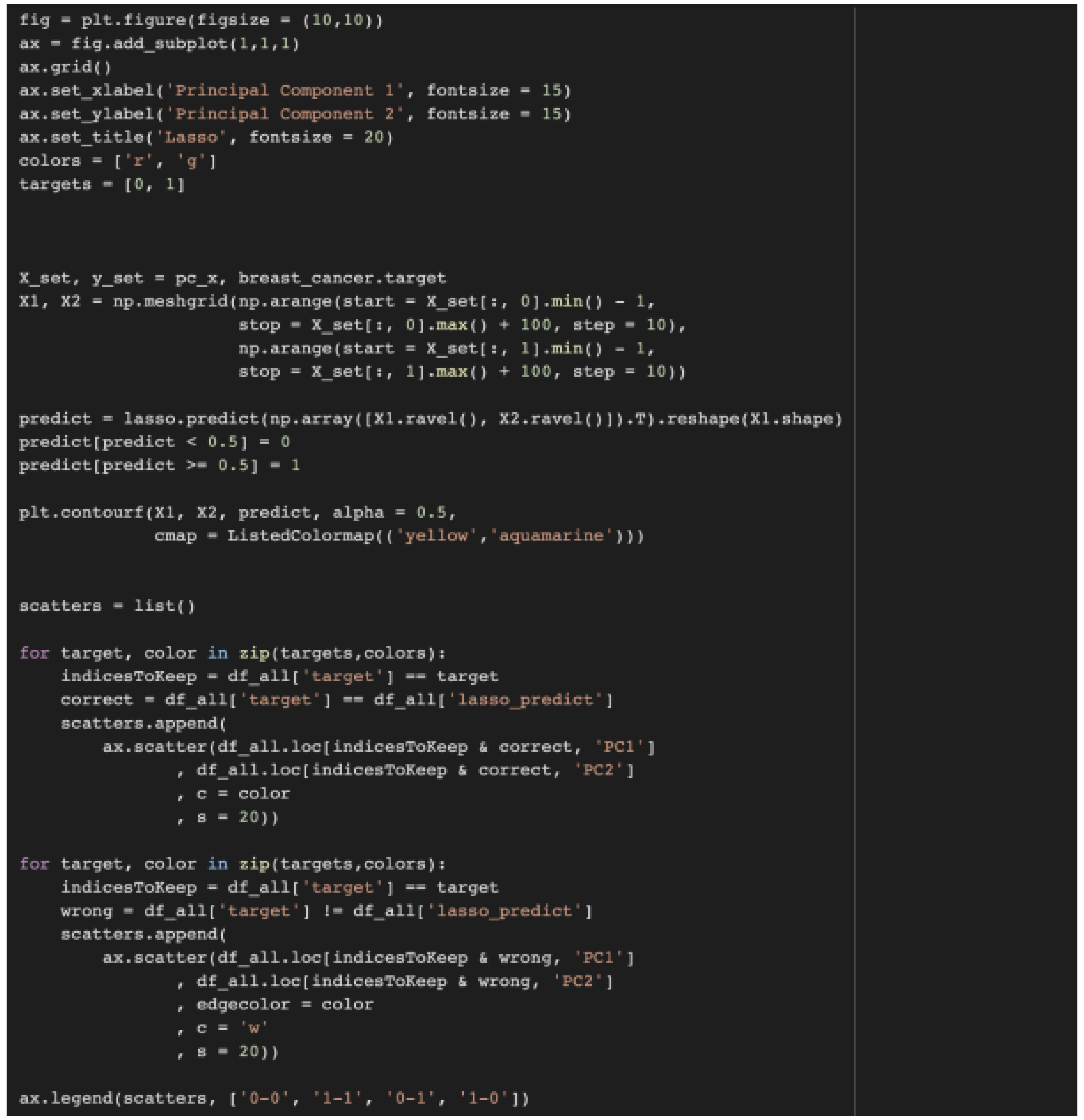
→Affected by PC1 and PC2 both

→Highest prediction accuracy

04

Results: Lasso Regression





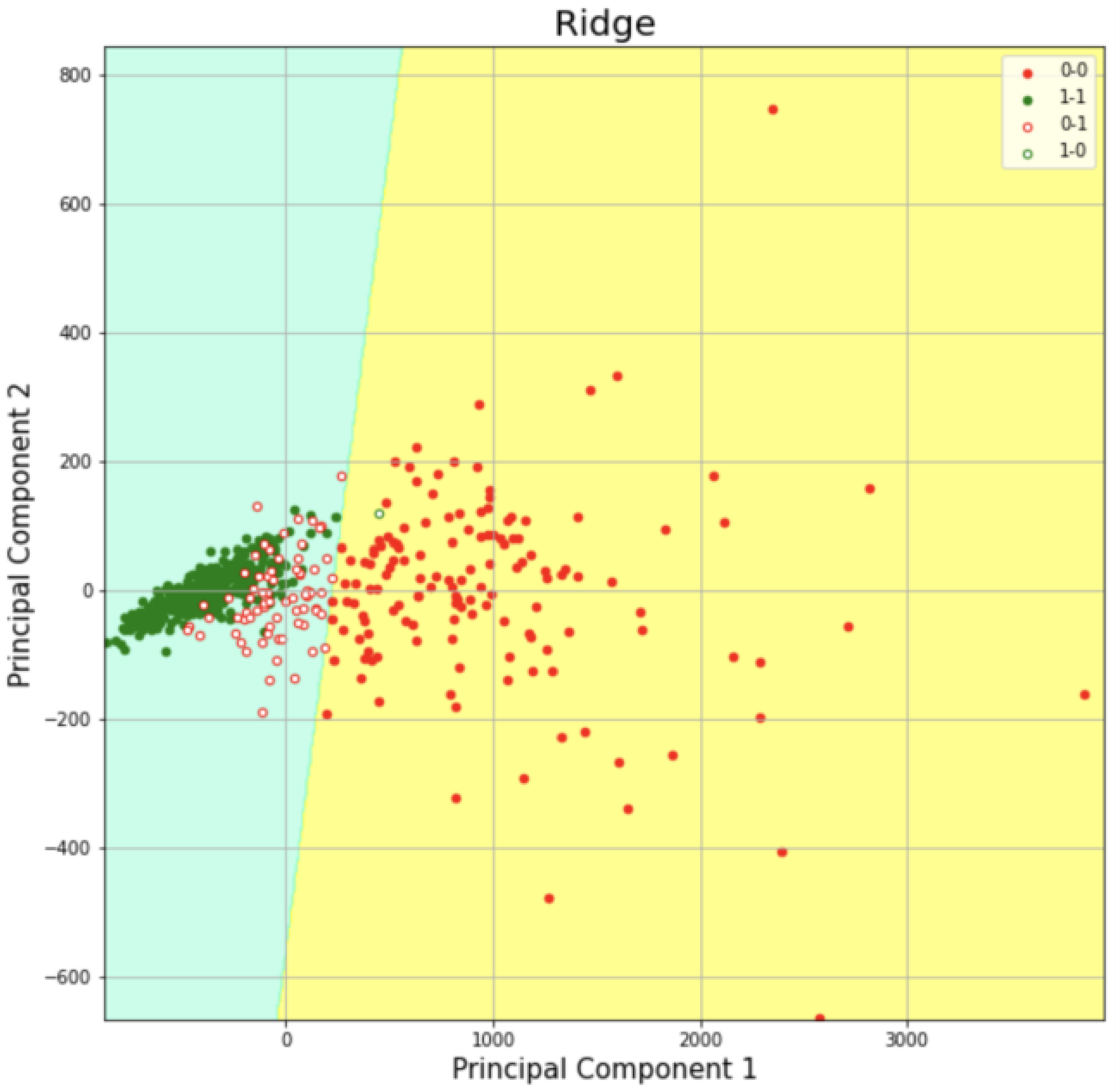
```
fig = plt.figure(figsize = (10,10))
ax = fig.add_subplot(1,1,1)
ax.grid()
ax.set_xlabel('Principal Component 1', fontsize = 15)
ax.set_ylabel('Principal Component 2', fontsize = 15)
ax.set_title('Lasso', fontsize = 20)
colors = ['r', 'g']
targets = [0, 1]



X_set, y_set = pc_x, breast_cancer.target
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1,
                     stop = X_set[:, 0].max() + 100, step = 10),
                     np.arange(start = X_set[:, 1].min() - 1,
                     stop = X_set[:, 1].max() + 100, step = 10))

predict = lasso.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape)
predict[predict < 0.5] = 0
predict[predict >= 0.5] = 1

plt.contourf(X1, X2, predict, alpha = 0.5,
             cmap = ListedColormap(('yellow','aquamarine')))


scatters = list()

for target, color in zip(targets,colors):
    indicesToKeep = df_all['target'] == target
    correct = df_all['target'] == df_all['lasso_predict']
    scatters.append(
        ax.scatter(df_all.loc[indicesToKeep & correct, 'PC1']
               , df_all.loc[indicesToKeep & correct, 'PC2']
               , c = color
               , s = 20))

for target, color in zip(targets,colors):
    indicesToKeep = df_all['target'] == target
    wrong = df_all['target'] != df_all['lasso_predict']
    scatters.append(
        ax.scatter(df_all.loc[indicesToKeep & wrong, 'PC1']
               , df_all.loc[indicesToKeep & wrong, 'PC2']
               , edgecolor = color
               , c = 'w'
               , s = 20))

ax.legend(scatters, ['0-0', '1-1', '0-1', '1-0'])
```

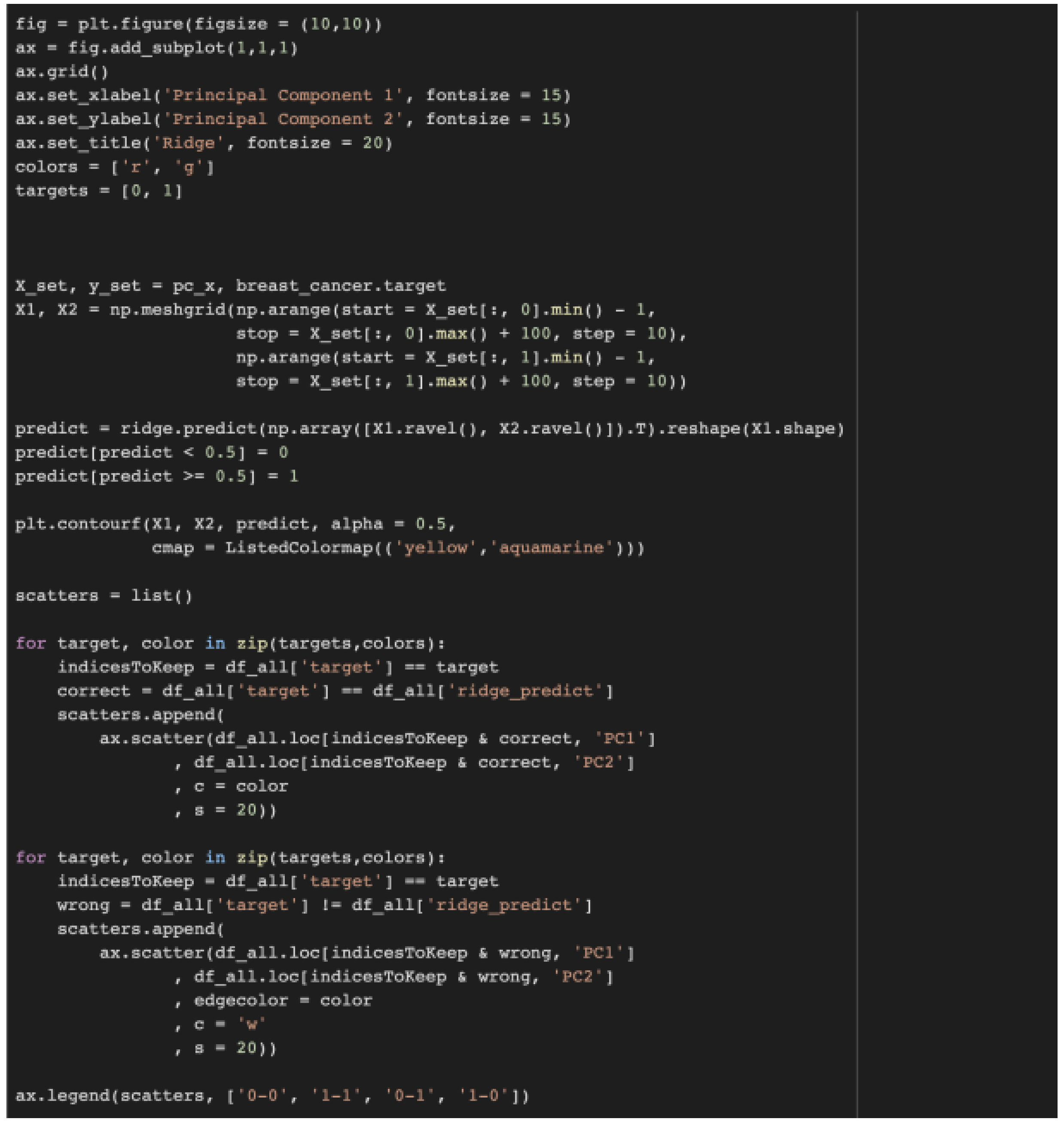
→Affected by PC1

→low prediction accuracy than logistic regression

04

Results: Ridge Regression





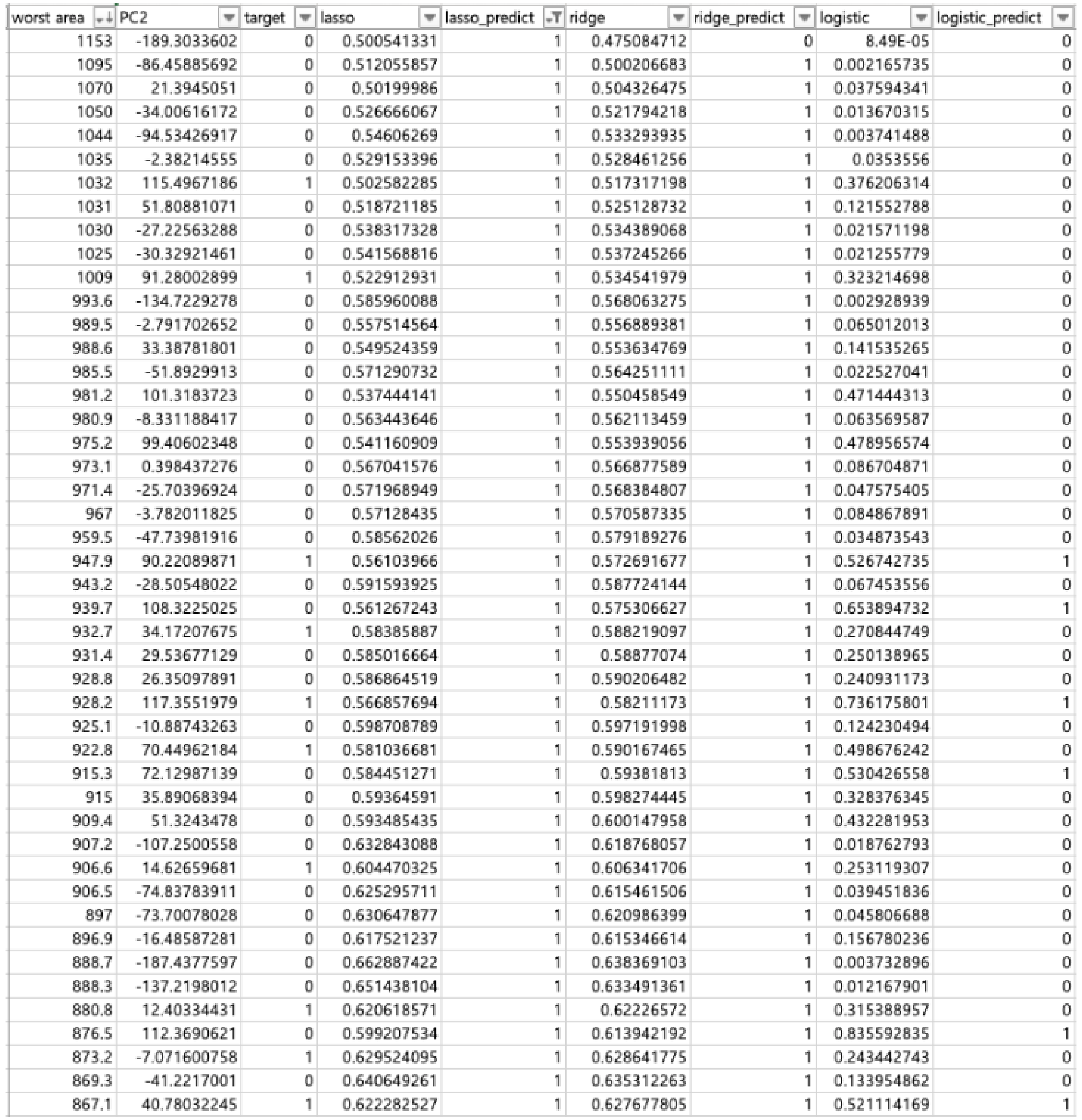
```
fig = plt.figure(figsize = (10,10))
ax = fig.add_subplot(1,1,1)
ax.grid()
ax.set_xlabel('Principal Component 1', fontsize = 15)
ax.set_ylabel('Principal Component 2', fontsize = 15)
ax.set_title('Ridge', fontsize = 20)
colors = ['r', 'g']
targets = [0, 1]



X_set, y_set = pc_x, breast_cancer.target
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1,
                     stop = X_set[:, 0].max() + 100, step = 10),
                     np.arange(start = X_set[:, 1].min() - 1,
                     stop = X_set[:, 1].max() + 100, step = 10))

predict = ridge.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape)
predict[predict < 0.5] = 0
predict[predict >= 0.5] = 1

plt.contourf(X1, X2, predict, alpha = 0.5,
             cmap = ListedColormap(('yellow','aquamarine')))

scatters = list()

for target, color in zip(targets,colors):
    indicesToKeep = df_all['target'] == target
    correct = df_all['target'] == df_all['ridge_predict']
    scatters.append(
        ax.scatter(df_all.loc[indicesToKeep & correct, 'PC1']
               , df_all.loc[indicesToKeep & correct, 'PC2']
               , c = color
               , s = 20))

for target, color in zip(targets,colors):
    indicesToKeep = df_all['target'] == target
    wrong = df_all['target'] != df_all['ridge_predict']
    scatters.append(
        ax.scatter(df_all.loc[indicesToKeep & wrong, 'PC1']
               , df_all.loc[indicesToKeep & wrong, 'PC2']
               , edgecolor = color
               , c = 'w'
               , s = 20))

ax.legend(scatters, ['0-0', '1-1', '0-1', '1-0'])
```

→Affected by PC1

→low prediction accuracy than logistic regression

05

Discussion of our work



worst area	PC2	target	lasso	lasso_predict	ridge	ridge_predict	logistic	logistic_predict
1153	-189.3033602	0	0.500541331	1	0.475084712	0	8.49E-05	0
1095	-86.45885692	0	0.512055857	1	0.500206683	1	0.002165735	0
1070	21.3945051	0	0.50199986	1	0.504326475	1	0.037594341	0
1050	-34.00616172	0	0.526666067	1	0.521794218	1	0.013670315	0
1044	-94.53426917	0	0.54606269	1	0.533293935	1	0.003741488	0
1035	-2.38214555	0	0.529153396	1	0.528461256	1	0.0353556	0
1032	115.4967186	1	0.502582285	1	0.517317198	1	0.376206314	0
1031	51.80881071	0	0.518721185	1	0.525128732	1	0.121552788	0
1030	-27.22563288	0	0.538317328	1	0.534389068	1	0.021571198	0
1025	-30.32921461	0	0.541568816	1	0.537245266	1	0.021255779	0
1009	91.28002899	1	0.522912931	1	0.534541979	1	0.323214698	0
993.6	-134.7229278	0	0.585960088	1	0.568063275	1	0.002928939	0
989.5	-2.791702652	0	0.557514564	1	0.556889381	1	0.065012013	0
988.6	33.38781801	0	0.549524359	1	0.553634769	1	0.141535265	0
985.5	-51.8929913	0	0.571290732	1	0.564251111	1	0.022527041	0
981.2	101.3183723	0	0.537444141	1	0.550458549	1	0.471444313	0
980.9	-8.331188417	0	0.563443646	1	0.562113459	1	0.063569587	0
975.2	99.40602348	0	0.541160909	1	0.553939056	1	0.478956574	0
973.1	0.398437276	0	0.567041576	1	0.566877589	1	0.086704871	0
971.4	-25.70396924	0	0.571968949	1	0.568384807	1	0.047575405	0
967	-3.782011825	0	0.57128435	1	0.570587335	1	0.084867891	0
959.5	-47.73981916	0	0.58562026	1	0.579189276	1	0.034873543	0
947.9	90.22089871	1	0.56103966	1	0.572691677	1	0.526742735	1
943.2	-28.50548022	0	0.591593925	1	0.587724144	1	0.067453556	0
939.7	108.3225025	0	0.561267243	1	0.575306627	1	0.653894732	1
932.7	34.17207675	1	0.58385887	1	0.588219097	1	0.270844749	0
931.4	29.53677129	0	0.585016664	1	0.58877074	1	0.250138965	0
928.8	26.35097891	0	0.586864519	1	0.590206482	1	0.240931173	0
928.2	117.3551979	1	0.566857694	1	0.58211173	1	0.736175801	1
925.1	-10.88743263	0	0.598708789	1	0.597191998	1	0.124230494	0
922.8	70.44962184	1	0.581036681	1	0.590167465	1	0.498676242	0
915.3	72.12987139	0	0.584451271	1	0.59381813	1	0.530426558	1
915	35.89068394	0	0.59364591	1	0.598274445	1	0.328376345	0
909.4	51.3243478	0	0.593485435	1	0.600147958	1	0.432281953	0
907.2	-107.2500558	0	0.632843088	1	0.618768057	1	0.018762793	0
906.6	14.62659681	1	0.604470325	1	0.606341706	1	0.253119307	0
906.5	-74.83783911	0	0.625295711	1	0.615461506	1	0.039451836	0
897	-73.70078028	0	0.630647877	1	0.620986399	1	0.045806688	0
896.9	-16.48587281	0	0.617521237	1	0.615346614	1	0.156780236	0
888.7	-187.4377597	0	0.662887422	1	0.638369103	1	0.003732896	0
888.3	-137.2198012	0	0.651438104	1	0.633491361	1	0.012167901	0
880.8	12.40334431	1	0.620618571	1	0.62226572	1	0.315388957	0
876.5	112.3690621	0	0.599207534	1	0.613942192	1	0.835592835	1
873.2	-7.071600758	1	0.629524095	1	0.628641775	1	0.243442743	0
869.3	-41.2217001	0	0.640649261	1	0.635312263	1	0.133954862	0
867.1	40.78032245	1	0.622282527	1	0.627677805	1	0.521114169	1

Wrong prediction analysis

: value of worst area, mean area

Case of malignant

: value of worst/mean area is large

Case of benign

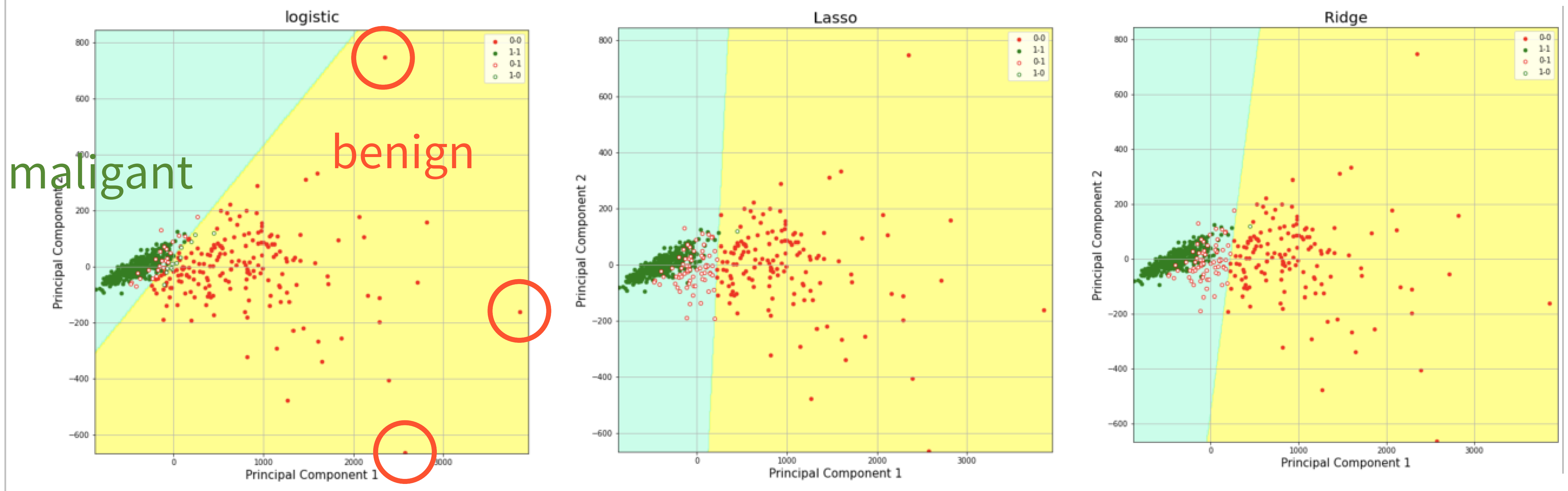
: value of worst/mean area is small

→ If the value is on the border,

the prediction is wrong.

05

Discussion of our work



Why is the predictive power of lasso/ridge regression less than that of logistic?

Logistic regression use sigmoid function: the value is 0 or 1

Lasso/Ridge regression: the value is real number

→ vulnerable to the outlier

THANK YOU