

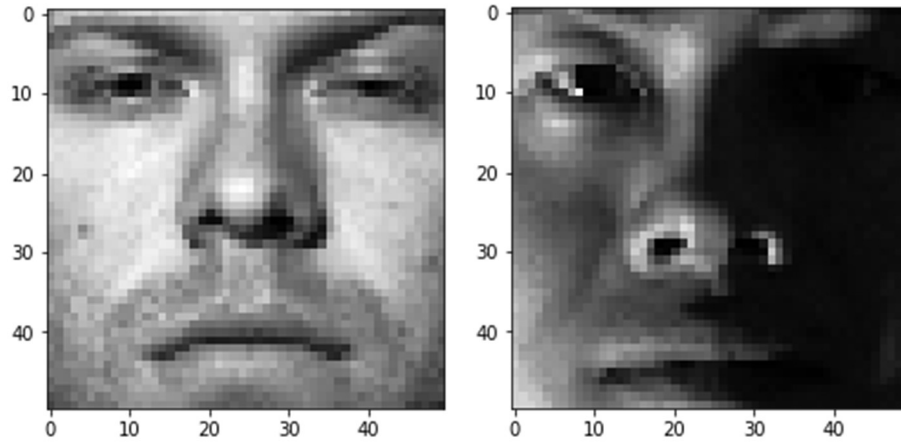
CS5785 Fall 2018: Homework 2

Matheus Clemente Bafutto (mc2722)

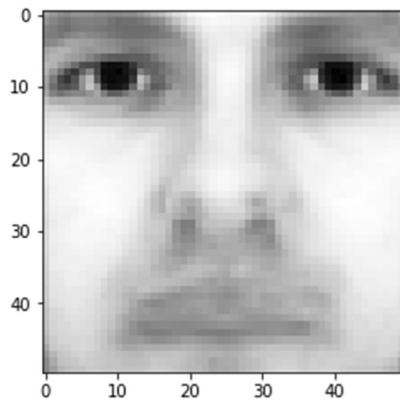
Sungseo Park (sp2528)

1. Eigenface for face recognition.

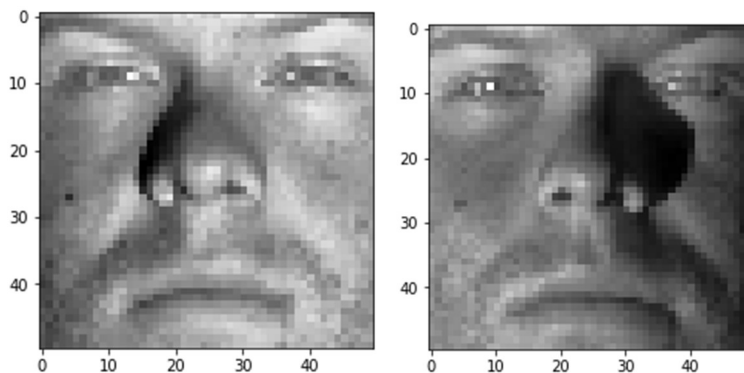
1b. Pick a face image from X and display that image in grayscale. Do the same thing for the test set.



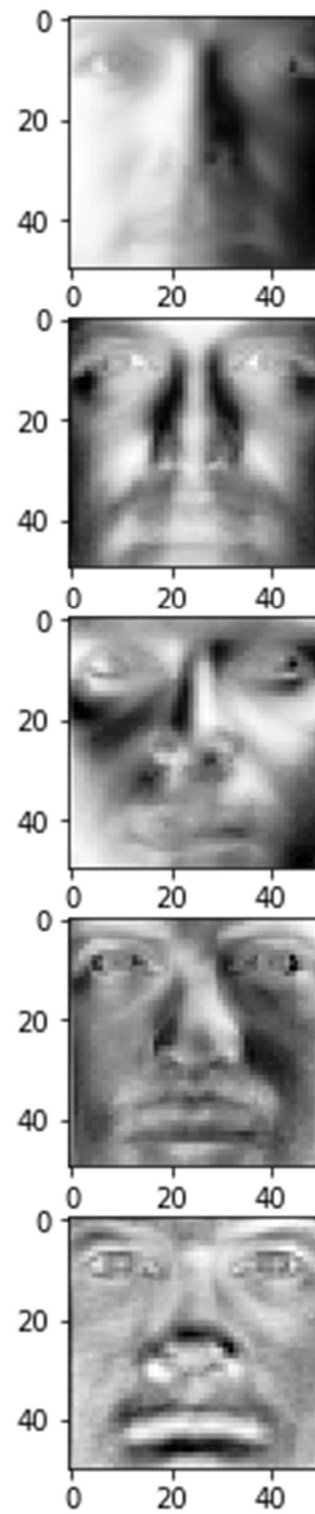
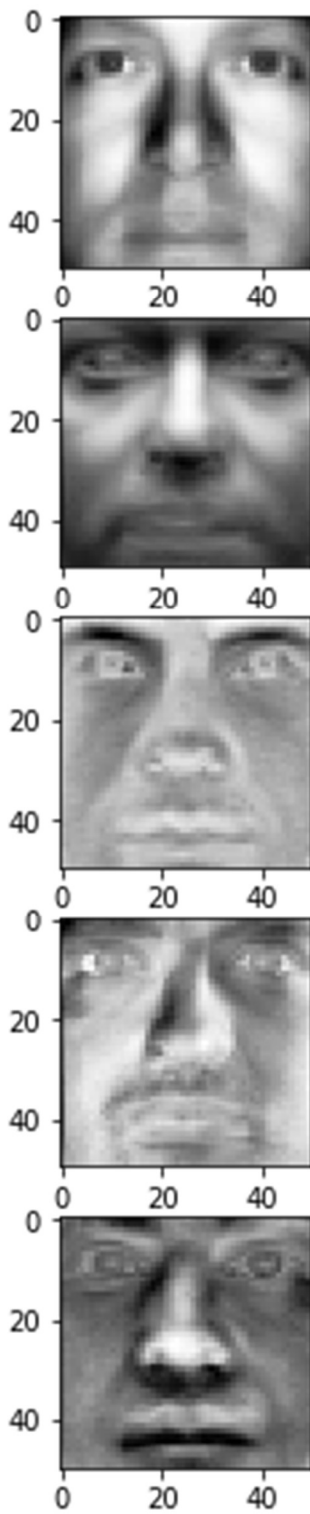
1c. Display the average face as a grayscale image.



1d. Pick a face image after mean subtraction from the new X and display that image in grayscale. Do the same thing for the test set X_{test} using the precomputed average face \bar{x} in (c).

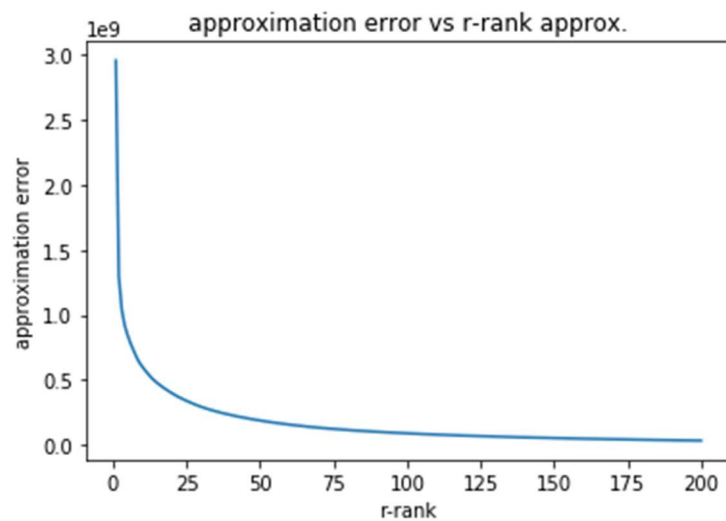


1e. Display the first 10 eigenfaces as 10 images in grayscale.



1f. Plot the rank-r approximation error.

```
1. errors = []
2. for r in range(1,201):
3.     train_data_hat = np.dot( np.dot(U[:, :r], S[:, :r]), Vh[:, :r])
4.     error = np.sum((train_data_normalized - train_data_hat)**2)
5.     errors.append(error)
6. plt.plot([i+1 for i in range(200)], errors)
7. plt.xlabel("r-rank")
8. plt.ylabel("approximation error")
9. plt.title("approximation error vs r-rank approx.")
10. plt.show()
```



1g. Write a function to generate r -dimensional feature matrix F and F_{test} .

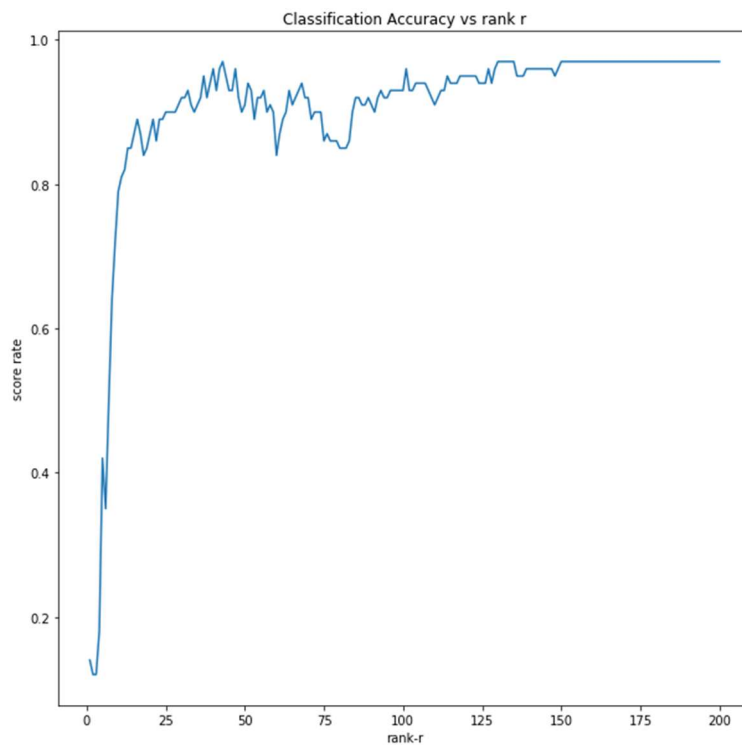
```
1. def generate_F(r, data):
2.     return np.dot(data, np.transpose(Vh[:, :r]))
3.
4. F = generate_F(10, train_data_normalized)
5. F_test = generate_F(10, test_data_normalized)
```

1h. Report the classification accuracy on the test set. Plot the classification accuracy on the test set as a function of r when $r = 1, 2, \dots, 200$.

```
1. from sklearn import linear_model
2.
3. model = linear_model.LogisticRegression()
4. model.fit(F, train_labels)
5. model.score(F_test, test_labels)
```

Classification accuracy = 0.79

```
1. performances = []
2. ranks = [i for i in range(1,201)]
3.
4. for rank in ranks:
5.     model = linear_model.LogisticRegression()
6.     model.fit(generate_F(rank,train_data_normalized), train_labels)
7.     performance = model.score(generate_F(rank, test_data_normalized), test_labels)
8.     performances.append(performance)
9.
10. plt.figure(figsize=(10,10))
11. plt.plot(ranks, performances)
12. plt.xlabel("rank-r")
13. plt.ylabel("score rate")
14. plt.title("Classification Accuracy vs rank r")
15. plt.show()
```



2. What's Cooking?

2b. Tell us about the data. How many samples (dishes) are there in the training set? How many categories (types of cuisine)? Use a list to keep all the unique ingredients appearing in the training set. How many unique ingredients are there?

of samples in training set: 39774

of categories in training set: 6714

of unique ingredients in training set: 20

2c. Represent each dish by a binary ingredient feature vector.

```
1. feature_codes = []
2.
3. for feature in unique_features:
4.     feature_code = ( unique_features == feature ).astype(int)
5.     feature_codes.append(feature_code)
6.
7. feature_codes = np.array(feature_codes)
8.
9. def MLify(data, targets=None):
10.     X = []
11.     if targets != None:
12.         labels = []
13.
14.     # for every item in data,
15.     # 1) encode ingredients into a row of X
16.     # 2) encode item label into one vs all format
17.     for i in range(len(data)):
18.         X.append(np.array([0 for j in range(len(unique_features))]))
19.         if targets != None:
20.             labels.append( (unique_labels == targets[i]).astype(int) )
21.             for ingredient in data[i]["ingredients"]:
22.                 X[i] = np.bitwise_or(X[i], (unique_features == ingredient).astype(int))
23.     print("features: ", len(X), " ", len(X[0]))
24.     if targets != None:
25.         print("labels: ", len(labels), " ", len(labels[0]))
26.
27.     if targets != None:
28.         return X, labels
29.     return X
30.
31. X_train, y_train = MLify(train_data, train_labels)
32. X_test = MLify(test_data)
```

2d. Using Naïve Bayes Classifier to perform 3 fold cross-validation on the training set and report your average classification accuracy. Try both Gaussian distribution prior assumption and Bernoulli distribution prior assumption.

```
1. from sklearn import model_selection
2. from sklearn.naive_bayes import GaussianNB
3. from sklearn.naive_bayes import BernoulliNB
4.
5. kfold = model_selection.KFold(n_splits=3)
6.
7. cv_results = model_selection.cross_val_score(GaussianNB(), X_train, train_labels, cv=kfold)
8. print('GaussianNB accuracy: {}'.format(cv_results.mean()))
9.
10. cv_results = model_selection.cross_val_score(BernoulliNB(), X_train, train_labels, cv=kfold)
11. print('BernoulliNB accuracy: {}'.format(cv_results.mean()))
```

GaussianNB accuracy: 0.38039925579524314

BernoulliNB accuracy: 0.6829587167496354

2e. For Gaussian prior and Bernoulli prior, which performs better in terms of cross-validation accuracy? Why? Please give specific arguments.

Bernoulli describes whether an event occurred. So, it should be used for features with binary or Boolean values. On the other hand, Gaussian is generally used for continuous data such as a real number. Since we represent each dish by a binary ingredient feature vector, Bernoulli performs better in terms of cross-validation accuracy.





2f. Using Logistic Regression Model to perform 3-fold cross-validation on the training set and report your average classification accuracy.

```
1. from sklearn.linear_model import LogisticRegression
2.
3. cv_results = model_selection.cross_val_score(LogisticRegression(), X_train, train_labels, cv=kfold)
4. print('LogisticRegression accuracy: {}'.format(cv_results.mean()))
```


Logistic Regression accuracy: 0.7751294815708755

2g. Train your best-performed classifier with all of the training data, and generate test labels on test set. Submit your results to Kaggle and report the accuracy.

```
1. import pandas as pd
2. from collections import OrderedDict
3.
4. logistic = LogisticRegression()
5. logistic.fit(X_train, train_labels)
6. logistic_prediction = logistic.predict(X_test)
7.
8. d = pd.DataFrame(data=OrderedDict([('id', test_id), ('cuisine', logistic_prediction)]))
9. d.to_csv('submission_whats_cooking.csv', index=False)
```

 Search kaggle  Competitions Datasets Kernels Discussion Learn ...  

Signed in as Sung
My Profile
My Account
Sign Out



What's Cooking?

Use recipe ingredients to categorize the cuisine

1,388 teams · 3 years ago

Overview Data Kernels Discussion Leaderboard Rules Team [My Submissions](#) [Late Submission](#)

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
submission_whats_cooking.csv	6 minutes ago	0 seconds	0 seconds	0.78338

Complete

[Jump to your position on the leaderboard](#) ▼