

```

import os
import random
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn.metrics as sm
from sklearn import svm
from sklearn.model_selection import GroupKFold
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import permutation_test_score
from google.colab import drive

```

```

drive.mount("/content/drive", force_remount=True)
os.chdir('/content/drive/MyDrive/Colab Notebooks/PSY3100')

```

Mounted at /content/drive

Making it easy to iterate

```

def group_permutation(area, n_permutations = 1000, n_features = None,
seed = None):
    numSubj = 10
    scores = []
    permutations = []

    for subj in range(numSubj):
        # construct file name
        filename = 'fMRI_Scenes/S%02d_%s.csv' % (subj + 1, area.upper())
        print('Processing Subj %d out of %d: %s' % (subj+1, numSubj,
filename))

        allData = pd.read_csv(filename, sep=r',', skipinitialspace = True,
index_col='type')

        # only use lineDrawings data
        lineData = allData.loc['lineDrawings']
        labels = lineData['category'].to_numpy()
        runIdx = lineData['run'].to_numpy()

        if n_features is None:
            samples = lineData.iloc[:, 2:].to_numpy()
        else:
            samples = lineData.iloc[:, 2:n_features].to_numpy()

        # set up cross validation classification
        numSplits = len(np.unique(runIdx))
        CVfolds = GroupKFold(n_splits=numSplits)

```

```

# defining the model (using the same one Prof. Dirk used)
clf = make_pipeline(StandardScaler(), svm.SVC(kernel='linear'))

# run permutation analysis
random.seed(seed)
score, permutation_scores, _ = permutation_test_score(clf,
samples, labels, groups=runIdx, scoring='accuracy', cv = CVfolds,
n_permutations=n_permutations, n_jobs=-1)
scores.append(score)
permutations.append(permutation_scores)

return np.mean(score), np.mean(permutations, axis=0), scores,
permutations

```

```

def plot_results(mean_score, mean_perm):
    chance = 1/6
    pvalue = (mean_score < mean_perm).sum() / len(mean_perm)

    plt.hist(mean_perm, 20, label='Permutation scores')

    ylim = plt.ylim()
    plt.vlines(mean_score, ylim[0], ylim[1], linestyle='--',
                color='g', linewidth=3, label='Classification Score'
                ' (pvalue %s)' % pvalue)
    plt.vlines(chance, ylim[0], ylim[1], linestyle='--',
                color='k', linewidth=3, label='Luck')

    plt.ylim(ylim)
    plt.legend()
    plt.xlabel('Score')
    plt.show()

```

PPA (using the first 20 features)

```

%%time
mean_score, mean_perm, scores, perms = group_permutation('ppa', 1000,
n_features=20, seed=1)
print('\nMean score: %.3f | std: %.3f' % (mean_score, np.std(scores)))
plot_results(mean_score, mean_perm)

```

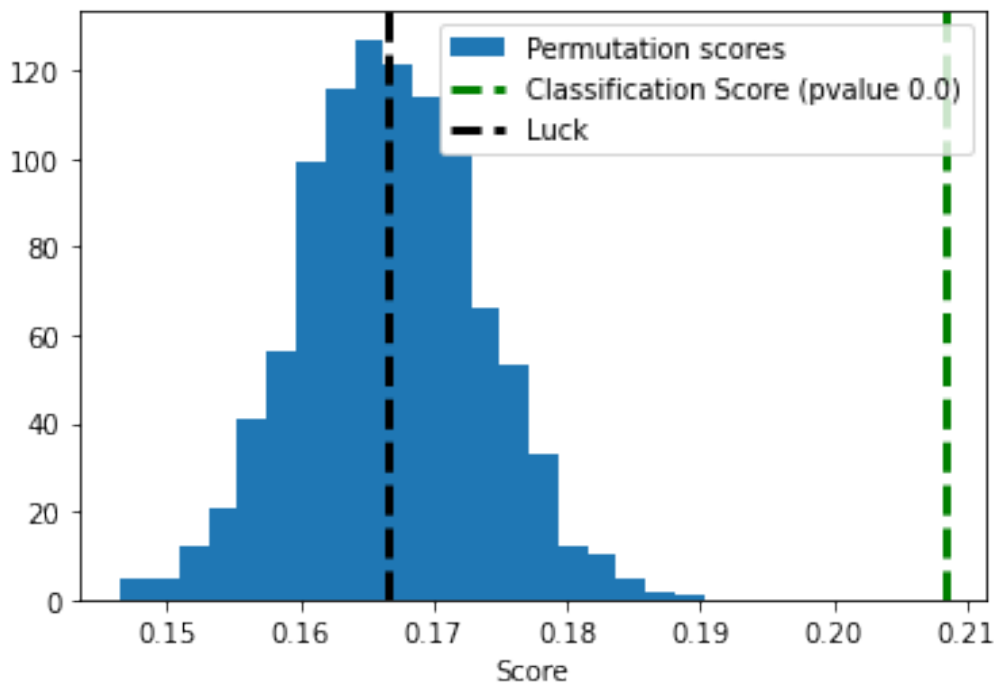
```

Processing Subj 1 out of 10: fMRI_Scenes/S01_PPA.csv
Processing Subj 2 out of 10: fMRI_Scenes/S02_PPA.csv
Processing Subj 3 out of 10: fMRI_Scenes/S03_PPA.csv
Processing Subj 4 out of 10: fMRI_Scenes/S04_PPA.csv
Processing Subj 5 out of 10: fMRI_Scenes/S05_PPA.csv
Processing Subj 6 out of 10: fMRI_Scenes/S06_PPA.csv
Processing Subj 7 out of 10: fMRI_Scenes/S07_PPA.csv
Processing Subj 8 out of 10: fMRI_Scenes/S08_PPA.csv
Processing Subj 9 out of 10: fMRI_Scenes/S09_PPA.csv

```

Processing Subj 10 out of 10: fMRI_Scenes/S10_PPA.csv

Mean score: 0.208 | std: 0.033



CPU times: user 1min 4s, sys: 3.26 s, total: 1min 7s

Wall time: 43min 25s

V1 (using the first 20 features)

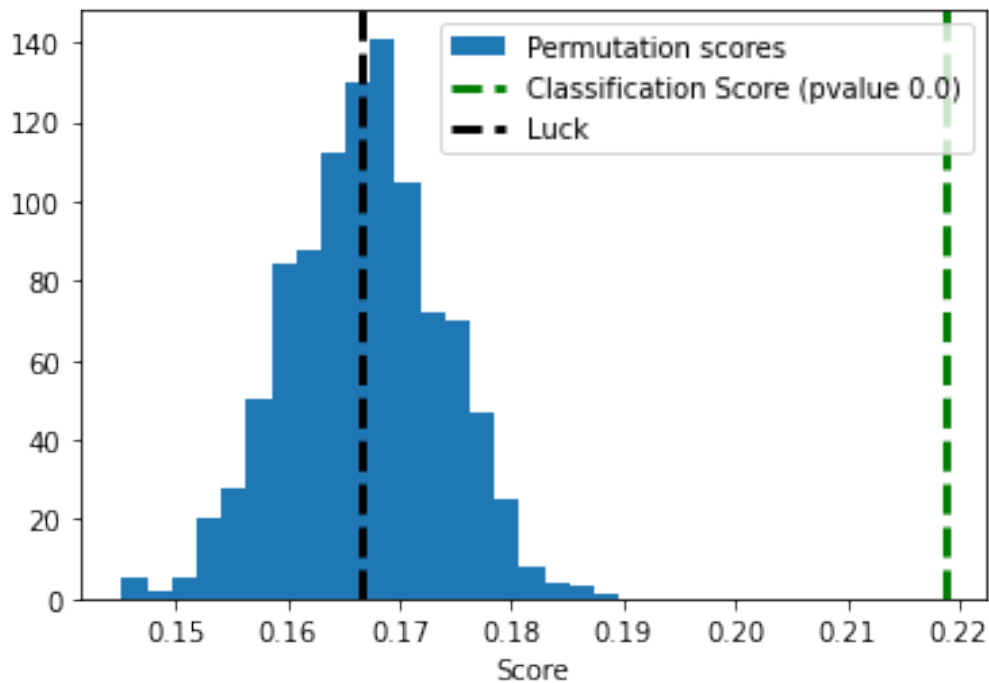
Runtime significantly higher because of the number of permutations (43 min 25s).

P-value was zero, meaning that none of the permutations exceeded the performance on the original labels.

```
%%time
mean_score, mean_perm, scores, perms = group_permutation('v1', 1000,
n_features=20, seed=1)
print('\nMean score: %.3f | std: %.3f' % (mean_score, np.std(scores)))
plot_results(mean_score, mean_perm)
```

Processing Subj 1 out of 10: fMRI_Scenes/S01_V1.csv
Processing Subj 2 out of 10: fMRI_Scenes/S02_V1.csv
Processing Subj 3 out of 10: fMRI_Scenes/S03_V1.csv
Processing Subj 4 out of 10: fMRI_Scenes/S04_V1.csv
Processing Subj 5 out of 10: fMRI_Scenes/S05_V1.csv
Processing Subj 6 out of 10: fMRI_Scenes/S06_V1.csv
Processing Subj 7 out of 10: fMRI_Scenes/S07_V1.csv
Processing Subj 8 out of 10: fMRI_Scenes/S08_V1.csv
Processing Subj 9 out of 10: fMRI_Scenes/S09_V1.csv
Processing Subj 10 out of 10: fMRI_Scenes/S10_V1.csv

Mean score: 0.219 | std: 0.027



CPU times: user 1min, sys: 2.73 s, total: 1min 3s

Wall time: 43min 17s

Runtime significantly higher because of the number of permutations (43min 17s).

P-value was zero, meaning that none of the permutations exceeded the performance on the original labels.