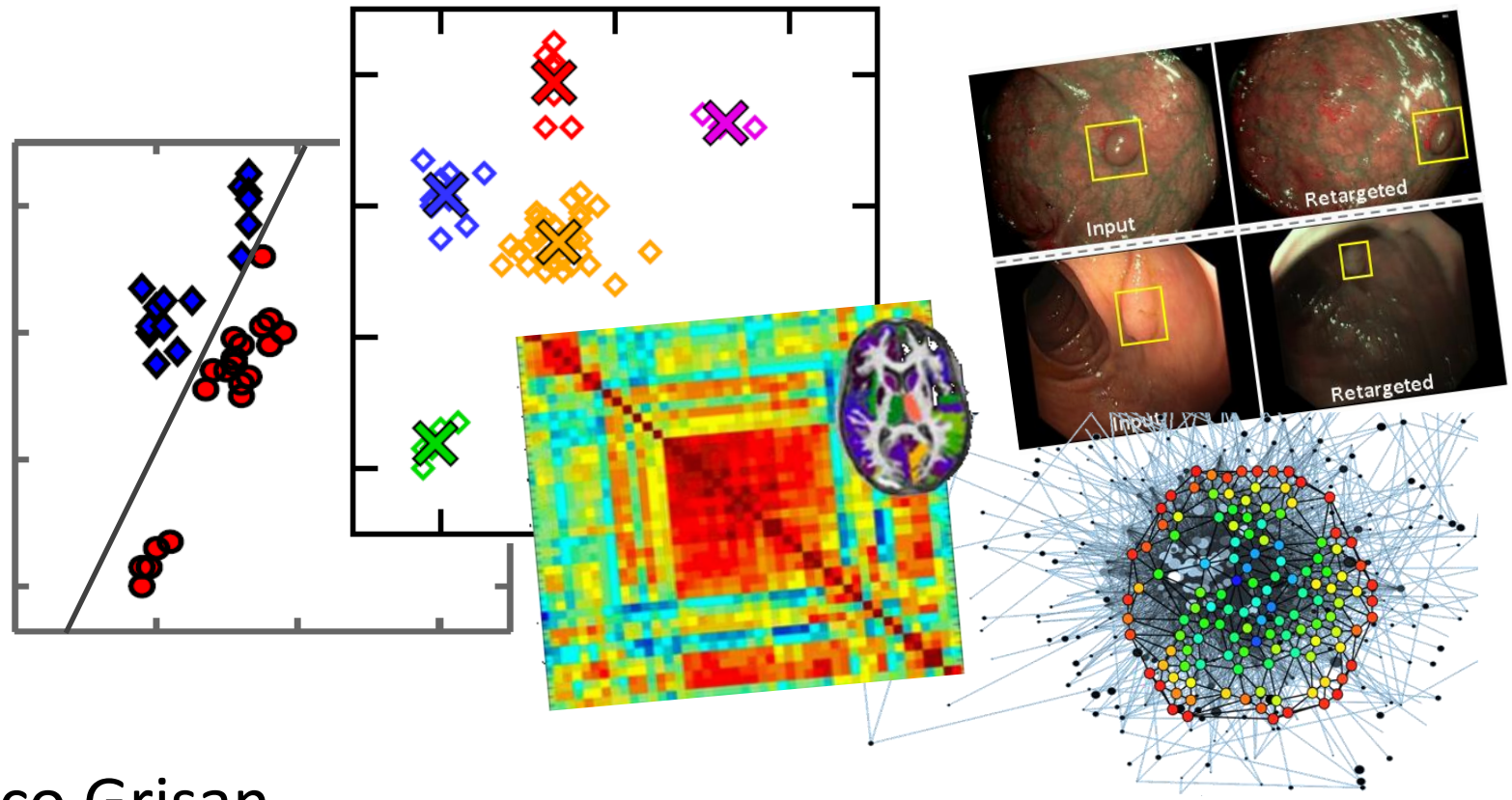
Machine Learning
CSI-7-MAL



Enrico Grisan

enrico.grisan@lsbu.ac.uk

Week 4 tutorial

- Same as last week but:

- Using ***Logistic Regression*** to predict amyloid positivity from demographic, cognitive and MRI data.

- Using ***validation***

Available data

Data are 702 (a subset of) patients extracted from the ADNI initiative database (https://adni.loni.usc.edu/) and used for the The Alzheimer's Disease Prediction Of Longitudinal Evolution (TADPOLE) grand challenge https://tadpole.grand-challenge.org/

Available data

Data contains:

- Patient code (PTID)
- Target variables: PET amyloid SUVR, and amyloid positive/negative
- Age, gender, education, diagnosis
- Apoe4 genetic alleles presence (0,1,2)
- Cognitive test scores (CD-RSB, Adas11, Adas13, MMSE, Ravlt, FAQ)
- Volumes of brain region computes from baseline brain MRI

Load the data in Python

```
import csv
import numpy as np
import pandas as pd

from sklearn.linear_model import LogisticRegression

from sklearn.model_selection import KFold
from sklearn.model_selection import train_test_split

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import accuracy_score

import matplotlib.pyplot as plt
```

```
filename='ADNI_Tadpole_data_tutorial.xlsx'
data_df=pd.read_excel(filename)

data_df.head()
```

Extract target variable and data

```
y_target = data_df['Amyloid status']
x_data=data_df.drop(['PTID','SUVR','Amyloid status'],axis=1)
```

Building the ApoE classifier

```
age=data_df['AGE'].to_numpy()
gender=data_df['GENDER '].to_numpy()
edu=data_df['EDUCATION'].to_numpy()
apoe=data_df['ApoE4'].to_numpy()


y=data_df['Amyloid status']


logit=-0.027*age-0.165*gender+0.080*edu+2.42*apoe+0.659
api=np.exp(logit)/(1+np.exp(logit))
```

Evaluating the ApoE classifier

```
from sklearn.metrics import roc_curve, auc, roc_auc_score
from sklearn.metrics import accuracy_score

print('AUC: {:.4f}'.format(roc_auc_score(y,api)))
print('Accuracy: {:.4f}'.format(accuracy_score(y,api>0.6)))

cm = confusion_matrix(y, api>0.6)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot()
plt.show()
```

Ex 1

WE WANT YOU!

- Go on the SciKit Learn web page: https://scikit-learn.org/stable/

- Look for documentation on ***Logistic Regression***

Ex 2

- Using the same variables as the ApoE classifier, retrain it and check the values of the estimated coefficient.

- What is the AUC and accuracy of the obtained classifier with a train-test split when evaluated on the training set?

- What is the AUC and accuracy of the obtained classifier with a train-test split when evaluated on the test set?

Import Logistic Regression

```
import csv
import numpy as np
import pandas as pd

from sklearn.linear_model import LogisticRegression

from sklearn.model_selection import KFold
from sklearn.model_selection import train_test_split

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import accuracy_score

import matplotlib.pyplot as plt
```

Prepare the validation

```
import csv
import numpy as np
import pandas as pd

from sklearn.linear_model import LogisticRegression

from sklearn.model_selection import KFold
from sklearn.model_selection import train_test_split

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import accuracy_score

import matplotlib.pyplot as plt
```

Prepare the validation

```
import csv
import numpy as np
import pandas as pd

from sklearn.linear_model import LogisticRegression

from sklearn.model_selection import KFold
from sklearn.model_selection import train_test_split

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import accuracy_score

import matplotlib.pyplot as plt
```

```
filename='ADNI_Tadpole_data_tutorial.xlsx'
data_df=pd.read_excel(filename)

data_df.head()
```

Split train and test

```
y_target = data_df['Amyloid status']
x_data=data_df[['AGE','GENDER ','EDUCATION','ApoE4']]

X_train, X_test, y_train, y_test = train_test_split(x_data, y_target, test_size=0.25)
```

Random split the input data and corresponding labels into two subsets.

Ratio of the split is 75% in the training set and 25% (0.25) in the test set

Using logistic regression

```
y_target = data_df['Amyloid status']
x_data=data_df[['AGE','GENDER ','EDUCATION','ApoE4']]

X_train, X_test, y_train, y_test = train_test_split(x_data, y_target, test_size=0.25)

clf = LogisticRegression(penalty='none', class_weight='none',max_iter=10000, solver='saga')
clf.fit(X_train, y_train)
y_hat = clf.predict(X_test)
p_hat = clf.predict_proba(X_test)
```

1) Prepare the classifier (look in the help for options!)
2) Fit the classifier on the ***training data***
3) Predict the classes y_hat on the ***test data***
4) Predict the logistic scores p_hat on the ***test data***

Cross validation

Go on the SciKit Learn web page:
https://scikit-learn.org/stable/

Look for documentation on
Cross validation: 3.1. Cross-validation: evaluating estimator performance — scikit-learn 1.3.1 documentation

Look for documentation on:
`Kfold` function
`cross_val_score` function

Homework

- Build a logistic classifier using all available variables
- What is the AUC and accuracy of the obtained classifier with a train-test split?
- What is the mean AUC and mean accuracy when running a 10-fold cross validation?