

CS 457 Homework 8: First Steps in ML
for Mon Sep 21 2026

Seaborn comes with some datasets. Let's play with them!

1. Load the measurements taken of some examples of three different kinds of iris flowers:

```
import matplotlib.pyplot as plt
import seaborn as sns
iris = sns.load_dataset("iris")
```

Get a sense of the dataset textually:

```
iris.describe()
```

and visually:

```
sns.pairplot(iris)
```

Let's try to capture the structure in this data. (This is called 'unsupervised machine learning'.) We'll use the scikit-learn module to try to find three clusters in the data:

```
from sklearn.mixture import GaussianMixture
justdata = iris.drop(columns=["species"])
model = GaussianMixture(n_components=3)
model.fit(justdata)
iris["cluster"] = model.predict(justdata).astype(str)
sns.pairplot(iris, hue="cluster")
plt.savefig("iris-clusters.pdf")
```

That last command should create a file in your colab project — you should see it if you click on the file folder icon (second from bottom) in the toolbar on the left edge of the window. Compare your plot to the true species from which each measurement was taken (which is recorded in the dataset):

```
sns.pairplot(iris, hue="species")
```

2. Load data on passengers from the titanic.

```
import matplotlib.pyplot as plt
import seaborn as sns
titanic = sns.load_dataset("titanic")
```

Get a sense of the dataset textually:

```
titanic.info()
```

and visually:

```
sns.pairplot(titanic)
```

Let's see if a passenger's survival can be predicted based on some of their other data. (This is called 'supervised machine learning', as our dataset has answers that we are learning to predict.) First, we prepare the data:

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# drop missing values
data = titanic[["survived", "pclass", "sex", "age", "fare"]].dropna()
# convert sex to numeric (female = 1)
data["sex"] = (data["sex"] == "female").astype(int)
```

To learn our predictor, we need to specify which data the predictor takes as its input (the ‘features’) and which data it is trying to predict (its ‘output’):

```
input = data[["pclass", "sex", "age", "fare"]]
output = data["survived"]
```

We will use some of the data to learn (ie ‘train’) our predictor and the rest to test it:

```
# split into training and test sets
in_train, in_test, out_train, out_test = train_test_split(input, output, test_size=0.2)
```

OK, here we go:

```
# fit a logistic regression predicition model
model = LogisticRegression(max_iter=1000)
model.fit(in_train, out_train)
# evaluate
predictions = model.predict(in_test)
print("Accuracy:", accuracy_score(out_test, predictions))
print("Coefficients:", dict(zip(input.columns, model.coef_[0])))
```

If you happen to be familiar with linear regression, these coefficients might give you thoughts about what factors were important for survival. If you aren’t, no worries — we’ll talk through it in class.

3. Please email the plot of the clusters you found in exercise 1 and the accuracy and coefficients of your prediction model in exercise 2 to Jack, CCing Wheeler.