Lecture 16
Classification and K-Nearest Neighbors

Dennis Sun
Stanford University
DATASCI / STATS 112

February 17, 2023

Review

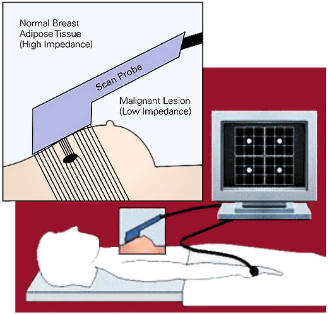
So far, we've been looking at **regression** problems, where the label y we are trying to predict is quantitative (e.g., price of wine).

Today, we switch to **classification** problems, where the label y is categorical.

We will focus on the differences between regression and classification.

Breast Tissue Classification

Electrical signals can be used to detect whether tissue is cancerous.



The goal is to determine whether a sample of breast tissue is:

1. connective tissue
2. adipose tissue
3. glandular tissue
4. carcinoma
5. fibro-adenoma
6. mastopathy

Reading in the Data

```python
import pandas as pd
df_breast = pd.read_csv(
    "http://dlsun.github.io/stats112/data/BreastTissue.csv")
df_breast
```

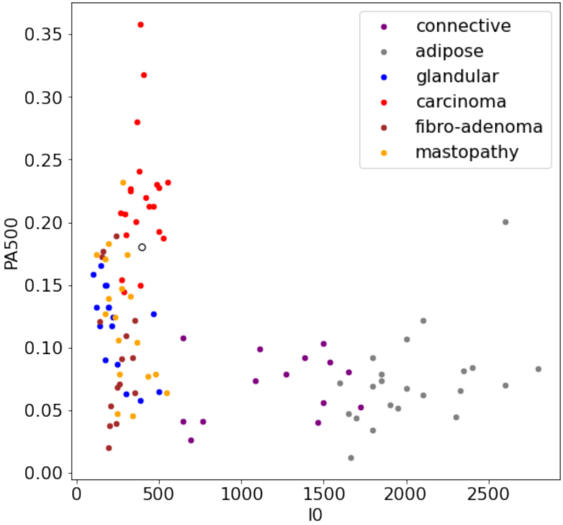
	Case #	Class	I0	PA500	HFS	DA	Area	A/DA	Max IP	DR	P
0	1	car	524.794072	0.187448	0.032114	228.800228	6843.598481	29.910803	60.204880	220.737212	556.828334
1	2	car	330.000000	0.226893	0.265290	121.154201	3163.239472	26.109202	69.717361	99.084964	400.225776
2	3	car	551.879287	0.232478	0.063530	264.804935	11888.391827	44.894903	77.793297	253.785300	656.769449
3	4	car	380.000000	0.240855	0.286234	137.640111	5402.171180	39.248524	88.758446	105.198568	493.701814
4	5	car	362.831266	0.200713	0.244346	124.912559	3290.462446	26.342127	69.389389	103.866552	424.796503
...	...	...	...	...	...	...	...	...	...	...	...
101	102	adi	2000.000000	0.106989	0.105418	520.222649	40087.920984	77.059161	204.090347	478.517223	2088.648870
102	103	adi	2600.000000	0.200538	0.208043	1063.441427	174480.476218	164.071543	418.687286	977.552367	2664.583623
103	104	adi	1600.000000	0.071908	-0.066323	436.943603	12655.342135	28.963331	103.732704	432.129749	1475.371534
104	105	adi	2300.000000	0.045029	0.136834	185.446044	5086.292497	27.427344	178.691742	49.593290	2480.592151
105	106	adi	2600.000000	0.069988	0.048869	745.474369	39845.773698	53.450226	154.122604	729.368395	2545.419744

106 rows × 11 columns

We will focus on two features:

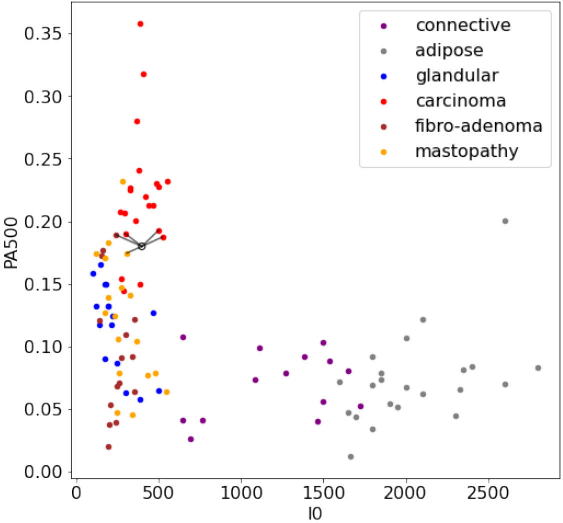
- I_0: impedivity at 0 kHz,
- PA_{500}: phase angle at 500 kHz.

Visualizing the Data



Consider a new sample, with an I_0 of 400 and a PA_{500} of 0.18.
What kind of tissue is it?

Visualizing the Data



Of its 5 nearest neighbors in the training data:
3 are carcinomas, 1 is fibro-adenoma, 1 is mastopathy,
so our best guess is that it is a carcinoma.

K-Nearest Neighbors

```
X_train = df_breast[["IO", "PA500"]]
y_train = df_breast["Class"]
x_test = pd.Series({"IO": 400, "PA500": .18})
```

Here is code we used for k-nearest neighbor regression.

```
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsRegressor
from sklearn.pipeline import make_pipeline

pipeline = make_pipeline(
    StandardScaler(),
    KNeighborsRegressor(n_neighbors=5, metric="euclidean"))

pipeline.fit(X_train, y_train)
pipeline.predict(pd.DataFrame([x_test]))
```

KNeighborsClassifier

What would need to change for classification?

K-Nearest Neighbors

```
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.pipeline import make_pipeline

pipeline = make_pipeline(
    StandardScaler(),
    KNeighborsClassifier(n_neighbors=5, metric="euclidean"))
pipeline.fit(X_train, y_train)
pipeline.predict(pd.DataFrame([x_test]))
```

```
array(['car'], dtype=object)
```

Instead of returning a single predicted class, we can ask it to return the predicted probabilities for each class.

```
pipeline.predict_proba(pd.DataFrame([x_test]))
```

```
array([[0. , 0.6, 0. , 0.2, 0. , 0.2]])
```

```
pipeline.classes_
```

```
array(['adi', 'car', 'con', 'fad', 'gla', 'mas'], dtype=object)
```

How did Scikit-Learn calculate these predicted probabilities?

Cross-Validation for Classification

Here is code we used to cross-validate a regression model.

```
from sklearn.model_selection import cross_val_score

cross_val_score(
    pipeline, X_train, y_train,
    scoring="neg_mean_squared_error",
    cv=10)
```

What would need to change for classification?

We need a different scoring method for classification. A simple one is **accuracy**:

$$\text{accuracy} = \frac{\#\text{ correct predictions}}{\#\text{ predictions}}.$$

Cross-Validation for Classification

```
scores = cross_val_score(
    pipeline, X_train, y_train,
    scoring="accuracy",
    cv=10)
scores
```

```
array([0.63636364, 0.81818182, 0.45454545, 0.54545455, 0.63636364,
       0.54545455, 0.5       , 0.6       , 0.4       , 0.7       ])
```

As before, we can get an overall estimate of test accuracy by averaging the cross-validation accuracies:

```
scores.mean()
```

```
0.5836363636363637
```

Accuracy is not always the best measure of a classification model. We'll talk about some better measures next time.

1. Review

2. Case Study: Breast Tissue Classification

3. K-Nearest Neighbors Classification

4. **Reminders**

Reminders

- Assignment 4 is due Monday. Start early, so you can maximize the number of Kaggle submissions.
- There is no class on Monday (Presidents' Day), but I will hold office hours at the usual time on Zoom. Check the website for the Zoom link.
- In section on Tuesday, you will pair up with a partner and get started on Assignment 5. You do not need to prepare anything!