Week 11: Classification, Final Review

DSUA111: Data Science for Everyone, NYU, Fall 2020

TA Jeff, `jpj251@nyu.edu`

- This slideshow: https://jjacobs.me/dsua111-sections/week-11 (https://jjacobs.me/dsua111-sections/week-11)
- All materials: https://github.com/jpowerj/dsua111-sections (https://github.com/jpowerj/dsua111-sections)

Outline

I. Classification

1. The K-Nearest Neighbors Algorithm
2. Evaluating KNN

II. Final Review

1. Big Picture Ideas
2. Math/Programming Details

Part I: Classification

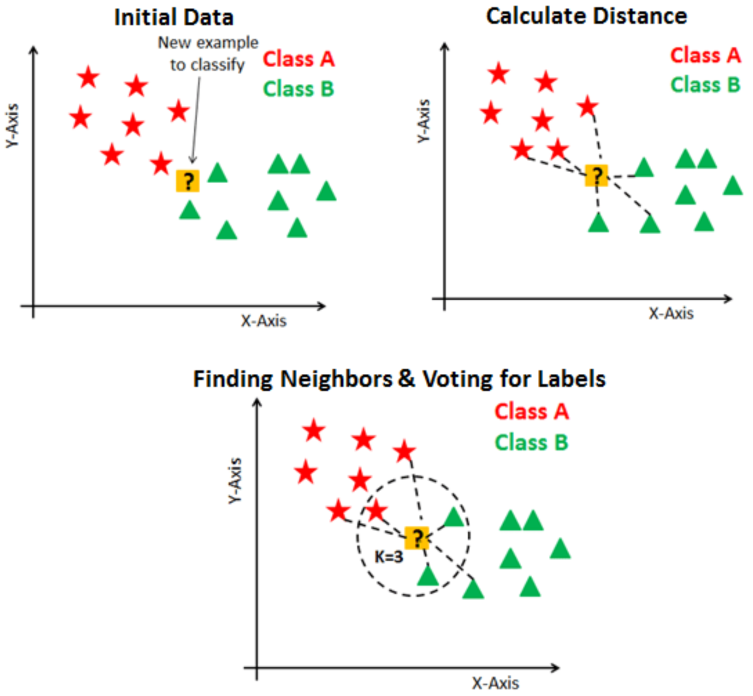
The K-Nearest Neighbors Algorithm

- Recall from last time:
    - **Statistics** is generally about *explanation*
    - **Machine Learning** is generally about *prediction*
- **Binary Classification**: Given a set of information ("features") about an observation (X), predict a yes/no outcome ($y \in \{0, 1\}$) for this observation
    - Example: Given a count of words in an email, classify it as spam ($y = 1$) or not spam ($y = 0$)
- **Multiclass classification**: Classify the observation into one of N categories ($y \in \{0, 1, \ldots, N\}$)
    - Example: Given a handwritten symbol, classify it as a digit ($y = \{0, 1, \ldots, 9\}$)
- K-Nearest Neighbors Intuition: Find the K most similar observations that we've seen before, and have them "majority vote" on the outcome.

K-Nearest Neighbors Example

- The problem: Given a student's GPA, predict whether or not they will graduate
- Many different potential approaches!
- K-Nearest Neighbor Approach:
    - Get a dataset of previous years, students' GPAs and whether or not they graduated
    - Find the $K = 5$ students with GPA closest to the student of interest
    - If a majority of them graduated, predict that the student will graduate. Otherwise, predict that they will not.

Binary Classification with 2 Features



(from [https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn (https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn)](https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn))

Evaluating KNN

- For **binary** classification: Seemingly easy, could just compute # correct, # incorrect
- We generally **DON'T WANT TO DO THIS** (why?)
- Instead, in actual machine learning projects, we use F-score:

$$F_1 = \frac{tp}{tp + \frac{1}{2}(tp + fn)}$$

- Don't worry about the details, the point is that what we **really** want to do is maximize accuracy (tp) **subject to a penalty** for false positives/negatives.
- This generalizes to multiclass classification: each category has its own F-score

Part II: Final Review

Common Misperceptions

0. True or False: The p-value is the probability that the null hypothesis is true.

False. It's the probability that we would obtain a test statistic value this "extreme" if the null hypothesis was true.

1. True or False: A 95% confidence interval means we're 95% confident that the true value of the parameter is between these two values.

False. It's just an interval computed in such a way that if we re-performed the experiment many many times, on average we'd expect the computed interval to contain the true value of the parameter about 95% of the time.

2. True or False: If our p-value is not low enough to meet our significance threshold (say, it's not below 0.05), then we reject the alternative hypothesis and accept the null hypothesis.

False. We never "accept" any hypotheses.

3. True or False: For `pd.read_csv("dataset.csv")` to work correctly, the `dataset.csv` file must be in the same folder as our notebook.

True. Otherwise, we have to specify how to "get to" the .csv file from the notebook's folder

4. What about `pd.read_csv("../dataset.csv")` ?

This tells Pandas to look one level above the folder containing the notebook, in the computer's directory tree. (So, if the notebook was located at `/home/data_science_projects/my_notebook.ipynb` , the above code would look for `dataset.csv` within the `/home` folder.

Regression Questions

We're going to load a dataset with GDP per capita (in USD) and level of inequality (as measured by Gini coefficient -- higher values = more unequal) for each country in the world. Then we'll perform a regression with GDP per capita as our **independent** variable and level of inequality as our **dependent** variable.

5. Write the equation for our **unfitted** model in this case

$$Gini_i = \beta_0 + \beta_1 GDP_i + \varepsilon_i$$

6. What is the **null hypothesis** that this regression is testing, **in terms of this equation**?

$$H_0 : \beta_1 = 0$$

7. What is the **null hypothesis** that this regression is testing, **in words**?

An increase of $1 in GDP per capita is not associated with any change in inequality

8. What is the **alternative hypothesis** that this regression is testing, **in terms of our equation**?

$$H_A : \beta_1 \neq 0$$

9. What is the **alternative hypothesis** that this regression is testing, **in words**?

An increase of $1 in GDP per capita is associated with a change in inequality

In [56]:
```
import pandas as pd
```

In [57]:
```
ineq_df = pd.read_csv("gdp_inequality.csv")
```

In [58]:
```
ineq_df.rename(columns={'Gini coefficient (World Bank (2016))':'gini',
                        'Output-side real GDP per capita (gdppc_o) (PWT 9.1 (2019))':'gd
p'},
               inplace=True)
```

In [59]:
```
ineq_df = ineq_df[~pd.isna(ineq_df["gini"])].copy()
```

In [60]:
```
ineq_df = ineq_df[~pd.isna(ineq_df["gdp"])].copy()
```

In [61]:
```
final_df = ineq_df.groupby("Code").last()
```

In [62]:
```
import statsmodels.formula.api as smf
```

In [63]:
```
result = smf.ols('gini ~ gdp', data=final_df).fit()
```

In [64]:
```
summ = result.summary(); summ.extra_txt = None
```

In [65]:
```
summ
```

Out[65]:

OLS Regression Results

Dep. Variable:	gini	**R-squared:**	0.145
Model:	OLS	**Adj. R-squared:**	0.139
Method:	Least Squares	**F-statistic:**	24.73
Date:	Thu, 03 Dec 2020	**Prob (F-statistic):**	1.83e-06
Time:	17:34:11	**Log-Likelihood:**	-520.62
No. Observations:	148	**AIC:**	1045.
Df Residuals:	146	**BIC:**	1051.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	42.2291	0.932	45.329	0.000	40.388	44.070
gdp	-0.0002	4.59e-05	-4.973	0.000	-0.000	-0.000

Omnibus:	3.685	**Durbin-Watson:**	1.859
Prob(Omnibus):	0.158	**Jarque-Bera (JB):**	3.256
Skew:	0.352	**Prob(JB):**	0.196
Kurtosis:	3.182	**Cond. No.**	2.80e+04

10. What does the -0.0002 in the "coef" column of the "gdp" row mean?

An increase of $1 in GDP per capita is associated with a decrease of 0.0002 in Gini coefficient (inequality)

11. Which column do we look at to obtain our p-value?

The column with header **P>|t|**.

12. True or False: Since the p-value is so low, we can conclude that increases in GDP cause decreases in inequality

False. We should **never** draw causal conclusions from the results of an OLS regression.

13. Where do we look to obtain the F-statistic for our model?

Towards the top -- the third row, right-hand column, lists the F-statistic (in this case, 24.73)

14. What does this value mean?

This is the **test statistic** for the hypothesis that **all** of the coefficients in our model are 0. With reference to our equation above, the null hypothesis for the F-statistic would be

$$H_0 : \beta_0 = 0 \text{ and } \beta_1 = 0$$

and the alternative hypothesis

$$H_A : \beta_0 \neq 0 \text{ or } \beta_1 \neq 0$$

The value 24.73 is high enough that the probability of obtaining an F-statistic as extreme as or more extreme than that is (listed directly underneath the F-statistic in the table) approximately 0. Thus we reject the null hypothesis that all coefficients are zero.

15. True or False: The adjusted R^2 value is lower than the R^2 value because GDP does not explain much of the variance in Gini coefficient

False. Adjusted R^2 is only lower than R^2 because it penalizes you for each new independent variable you introduce into the model.

16. How would the interpretation of our coefficient on GDP change if we added another independent variable?

Whereas in the single-variable case we interpret the coefficient as just the effect of GDP on inequality, if we introduced a new independent variable X_2 we would have to re-interpret our coefficient on GDP as the effect of GDP on inequality **holding $\boldsymbol{X_2}$ constant** (or, if we center our variables, like we really should: the effect of GDP on inequality **at the average $\boldsymbol{X_2}$ value**)

Some Stats

17. If we have an observation $x_5 = 123.45$ in a dataset, and find that the z-score for this value is $z_5 = 2.00$, what does this tell us about the original observation?

It tells us that 123.45 is almost exactly 2 standard deviations above the mean x value.

18. True or False: If the mean of a variable x in our dataset is 50.0 and the standard deviation is 5.0, we know that approximately 68% of the values of x lie within one standard deviation of this mean, i.e., between 45.0 and 55.0.

False. This is important. Because the "68% of the data lie within one standard deviation of the mean" property only holds for **normally-distributed** observations. The question never stated that the data was normally distributed, thus we can't assume the "68% rule" here.

Some Coding

19. What will the following code output?

```
for i in range(100):
    if i % 7 == 0:
        print(i)
```

In [92]:

```
for i in range(100):
    if i % 7 == 0:
        print(i)
```

```
0
7
14
21
28
35
42
49
56
63
70
77
84
91
98
```

20. What will this code output?

```
for i in range(100):
    if i / 7 == 0:
        print(i)
```

In [93]:
```
for i in range(100):
    if i / 7 == 0:
        print(i)
```

```
0
```

21. How many times will the following code print `"hello!"?

```
i = 0
while i < 4:
    for j in range(2):
        print("hello!")
    i = i + 1
```

In [99]:

```
i = 0
while i < 4:
    for j in range(2):
        print("hello!")
    i = i + 1
```

```
hello!
hello!
hello!
hello!
hello!
hello!
hello!
hello!
```

22. How many times will the following code print `"hello!"` ?

```
i = 0
while i < 4:
    for j in range(i):
        print("hello!")
    i = i + 1
```

In [100]:

```python
i = 0
while i < 4:
    for j in range(i):
        print("hello!")
    i = i + 1
```

```
hello!
hello!
hello!
hello!
hello!
hello!
```

Returning to our DataFrame...

In [101]:

```
final_df.head()
```

Out[101]:

	Entity	Year	Total population (Gapminder, HYDE & UN)	Continent	gini	gdp	high_gdp
Code							
AGO	Angola	2008	21696000.0	NaN	42.72	6080.5405	0
ALB	Albania	2012	2914000.0	NaN	28.96	10402.6360	1
ARG	Argentina	2013	42196000.0	NaN	42.28	16920.7560	1
ARM	Armenia	2013	2898000.0	NaN	31.54	9481.5098	0
AUS	Australia	2010	22155000.0	NaN	34.94	44854.9020	1

23. True or False: The following code permanently renames the "Total population" column so it is henceforth named "pop"

In [42]:

```
final_df.rename(columns={'Total population (Gapminder, HYDE & UN)':'pop'})
```

Out[42]:

	Entity	Year	pop	Continent	gini	gdp
Code						
AGO	Angola	2008	21696000.0	NaN	42.72	6080.5405
ALB	Albania	2012	2914000.0	NaN	28.96	10402.6360
ARG	Argentina	2013	42196000.0	NaN	42.28	16920.7560
ARM	Armenia	2013	2898000.0	NaN	31.54	9481.5098
AUS	Australia	2010	22155000.0	NaN	34.94	44854.9020
...	...	...	...	...	...	...
VEN	Venezuela	2006	26850000.0	NaN	46.94	12223.4100
VNM	Vietnam	2012	89802000.0	NaN	38.70	4933.5288
YEM	Yemen	2005	20107000.0	NaN	35.89	3196.2153
ZAF	South Africa	2011	52004000.0	NaN	63.38	11832.0590
ZMB	Zambia	2010	13606000.0	NaN	55.62	2870.8872

148 rows × 6 columns

In [43]:
```
final_df.head()
```

Out[43]:

	Entity	Year	Total population (Gapminder, HYDE & UN)	Continent	gini	gdp
Code						
AGO	Angola	2008	21696000.0	NaN	42.72	6080.5405
ALB	Albania	2012	2914000.0	NaN	28.96	10402.6360
ARG	Argentina	2013	42196000.0	NaN	42.28	16920.7560
ARM	Armenia	2013	2898000.0	NaN	31.54	9481.5098
AUS	Australia	2010	22155000.0	NaN	34.94	44854.9020

In [44]:
```
final_df.rename(columns={'Total population (Gapminder, HYDE & UN)':'pop'}, inplace=True)
```

In [45]:
```
final_df.head()
```

Out[45]:

	Entity	Year	pop	Continent	gini	gdp
Code						
AGO	Angola	2008	21696000.0	NaN	42.72	6080.5405
ALB	Albania	2012	2914000.0	NaN	28.96	10402.6360
ARG	Argentina	2013	42196000.0	NaN	42.28	16920.7560
ARM	Armenia	2013	2898000.0	NaN	31.54	9481.5098
AUS	Australia	2010	22155000.0	NaN	34.94	44854.9020

24. Say we make a new variable `high_gdp` which is 1 if GDP is greater than $10000 and 0 otherwise. What type of variable (not data type) is `high_gdp` ?

It is a [binary] **ordinal** variable, since there is a natural ordering (since we know countries with `high_gdp` = 0 have lower GDPs than countries with `high_gdp` = 1).

In [69]:
```
final_df['high_gdp'] = final_df['gdp'].apply(lambda x: 1 if x > 10000 else 0)
```

In [105]:
```
final_df.head()
```

Out[105]:

	Entity	Year	Total population (Gapminder, HYDE & UN)	Continent	gini	gdp	high_gdp
Code							
AGO	Angola	2008	21696000.0	NaN	42.72	6080.5405	0
ALB	Albania	2012	2914000.0	NaN	28.96	10402.6360	1
ARG	Argentina	2013	42196000.0	NaN	42.28	16920.7560	1
ARM	Armenia	2013	2898000.0	NaN	31.54	9481.5098	0
AUS	Australia	2010	22155000.0	NaN	34.94	44854.9020	1

25. Say we filled in the `Continent` column, so that e.g. Africa = 0, Asia = 1, South America = 2, North America = 3, Europe = 4, Oceania = 5. What type of variable (not data type) would `Continent` be in this case?

In this case `Continent` would be a **categorical** variable, not an ordinal variable, since there is no natural ordering of the values. We could just as easily have labeled the continents so that Europe = 0, Oceania = 1, Asia = 2, Africa = 3, North America = 4, South America = 5, without losing any information that this variable is supposed to hold.

In [72]:

```
sorted_df = final_df.sort_values(by="gini").copy()
```

In [80]:

```
us_gini = sorted_df.loc["USA"]["gini"]
us_gini
```

Out[80]:

```
41.06
```

In [84]:

```
import numpy as np
sorted_df['gini_vs_us'] = sorted_df['gini'] - us_gini
```

In [88]:

```
sorted_df.iloc[83:95]
```

Out[88]:

	Entity	Year	Total population (Gapminder, HYDE & UN)	Continent	gini	gdp	high_gdp	gini_vs_us
Code								
TUR	Turkey	2012	7.465100e+07	NaN	40.17	20904.22300	1	-0.89
TTO	Trinidad and Tobago	1992	1.237000e+06	NaN	40.27	9583.64550	0	-0.79
SEN	Senegal	2011	1.303400e+07	NaN	40.28	2735.58670	0	-0.78
MDG	Madagascar	2010	2.115200e+07	NaN	40.63	1459.91550	0	-0.43
MAR	Morocco	2007	3.116400e+07	NaN	40.72	4984.77200	0	-0.34
TKM	Turkmenistan	1998	4.413000e+06	NaN	40.77	6358.13180	0	-0.29
USA	United States	2013	3.164010e+08	NaN	41.06	51547.74600	1	0.00
RUS	Russia	2012	1.439940e+08	NaN	41.59	26074.13100	1	0.53
URY	Uruguay	2013	3.389000e+06	NaN	41.87	18652.68600	1	0.81
CHN	China	2010	1.368811e+09	NaN	42.06	9337.29000	0	1.00
COD	Democratic Republic of Congo	2012	6.902100e+07	NaN	42.10	740.51245	0	1.04
GAB	Gabon	2005	1.391000e+06	NaN	42.18	15793.24700	1	1.12

25. If we only knew the US's gini coefficient (and not its GDP), and we used the K-Nearest Neighbors algorithm with $K = 5$ to try and predict whether it was low or high GDP, which would we predict?

- 1st closest neighbor: Turkmenistan (low GDP)
- 2nd closest neighbor: Morocco (low GDP)
- 3rd closest neighbor: Madagascar (low GDP)
- 4th closest neighbor: Russia (high GDP)
- 5th closest neighbor: Senegal (low GDP)

4 out of 5 are low GDP $\implies$ we predict low GDP for US

26. What about with $K = 7$?

(The above 5 closest neighbors, plus:)

- 6th closest neighbor: Trinidad and Tobago (low GDP)
- 7th closest neighbor: Uruguay (high GDP)

5 out of 7 are low GDP $\implies$ we predict low GDP for US again

27. Why do we always pick odd numbers as values for K?

Because we need to take a majority vote of the K neighbors, so odd numbers ensure that we won't encounter any ties.