



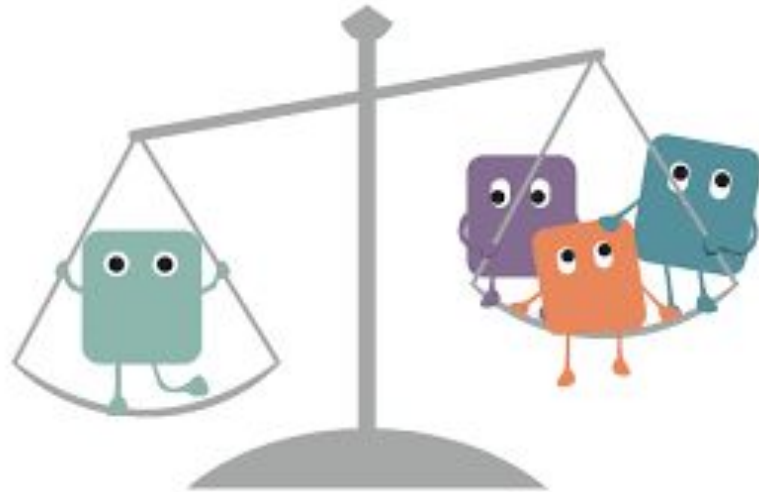
Lesson #07

Feature Selection & Pipelines

- Hypothesis test
 - Significant test
 - Chi-squared test
- Feature Selection for ML
 - Univariate selection
 - Recursive feature elimination
- Pipelines

A common problem in applied machine learning is determining whether **input features** are **relevant to the outcome to be predicted**

Feature selection



WHAT IS A HYPOTHESIS?



"A hypothesis is an idea that
can be tested"



Did raising the price of a product cause a meaningful drop in sales?



Has a new banner ad on a website caused a meaningful drop in the user engagement?

Null vs. Alternative Hypothesis

Null Hypothesis

$$H_0$$

A statement about a population parameter.

We test the likelihood of this statement being true in order to decide whether to accept or reject our alternative hypothesis.

Can include =, \leq , or \geq sign.

Alternative Hypothesis

$$H_a$$

A statement that directly contradicts the null hypothesis.

We determine whether or not to accept or reject this statement based on the likelihood of the null (opposite) hypothesis being true.

Can include a \neq , $>$, or $<$ sign.





Dude, I think data
scientists make more
than \$125,000!

Hm...





$H_0 : \mu_0$

$> \$125,000$


$H_1 : \mu_0$

$\leq \$125,000$

**THE NULL HYPOTHESIS IS THE STATEMENT WE ARE TRYING TO REJECT.
THEREFORE THE NULL IS THE PRESENT STATE OF AFFAIRS WHILE THE
ALTERNATIVE IS OUR PERSONAL OPINION.**



EXAMPLE


$$H_0: \mu_0 = \$125,000$$

Accept if: \bar{x} is close enough to the true mean

Reject if: \bar{x} is too far from the true mean



A new weight loss pill helped people lose more weight:

H_0 : patients who went on the weight loss pill lost no more weight than those who didn't.

H_1 : patients who went on the weight loss pill lost more weight than those who didn't

*New Weight
Loss Procedure*

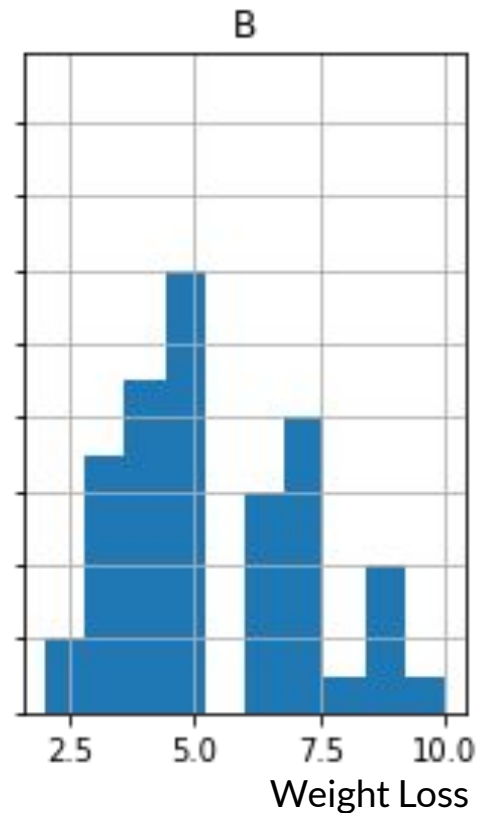
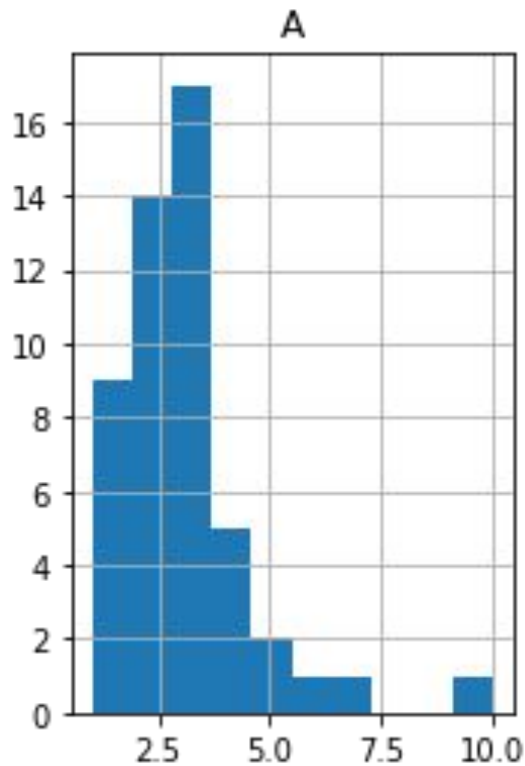


**Lose the Pounds
Without the Knife**

Blind Experiment

- Group A was given a placebo, or fake, pill and instructed to consume it on a daily basis.
- Group B was given the actual weight loss pill and instructed to consume it on a daily basis.





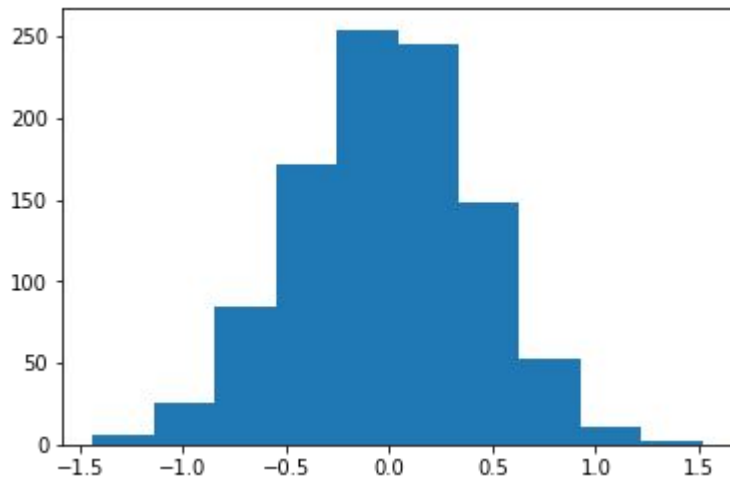
Null hypothesis: $\bar{x}_b - \bar{x}_a = 0$

Alternative hypothesis: $\bar{x}_b - \bar{x}_a > 0$

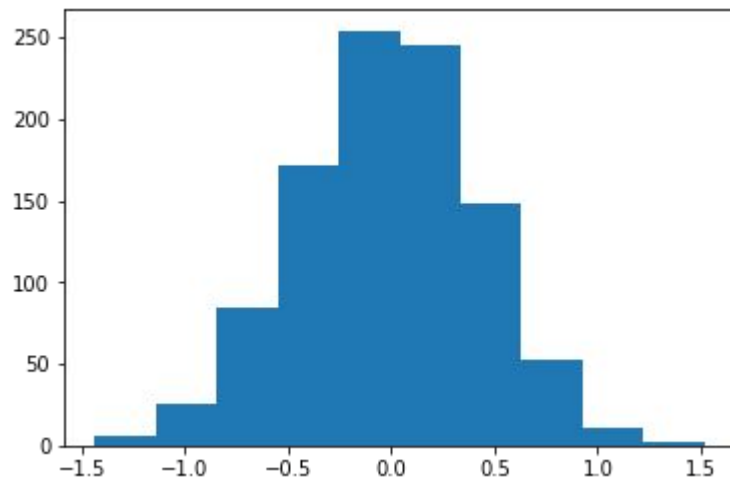
$$\bar{x}_b - \bar{x}_a = 2.52$$

Permutation Test

```
mean_difference = 2.52
mean_differences = []
for i in range(1000):
    group_a = []
    group_b = []
    for value in all_values:
        assignment_chance = np.random.rand()
        if assignment_chance >= 0.5:
            group_a.append(value)
        else:
            group_b.append(value)
    iteration_mean_difference = np.mean(group_b) - np.mean(group_a)
    mean_differences.append(iteration_mean_difference)
plt.hist(mean_differences)
```



```
[ (0.036814725890355504, 8),  
  (-0.160000000000000014, 7),  
  (-0.3709353673223603, 6),  
  (-0.4471153846153846, 6),  
  (-0.046474358974359475, 6),  
  (0.17021276595744705, 6),  
  (0.16326530612244916, 6),  
  (0.08992372541148086, 5),  
  (0.18840579710144967, 5),  
  (-0.0305098354074671, 5)]
```



```
frequencies = []  
for sp in sampling_distribution.keys():  
    if sp >= 2.52:  
        frequencies.append(sampling_distribution[sp])  
p_value = np.sum(frequencies) / 1000  
p_value  
0.0
```

In general, it's good practice to set the **p value** threshold before conducting the study:

- if the **p value** is less than the threshold, we:
 - **reject the null hypothesis** that there's no difference in mean amount of weight lost by participants in both groups,
 - **accept the alternative hypothesis** that the people who consumed the weight loss pill lost more weight,
 - conclude that the weight loss pill does affect the amount of weight people lost.
- if the **p value** is greater than the threshold, we:
 - **accept the null hypothesis** that there's no difference in the mean amount of weight lost by participants in both groups,
 - **reject the alternative hypothesis** that the people who consumed the weight loss pill lost more weight,
 - conclude that the weight loss pill doesn't seem to be effective in helping people lose more weight.

	age	workclass	education_num	marital_status	occupation	relationship	race	sex	high_income
0	39	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	<=50K
1	50	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	<=50K
2	38	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	<=50K
3	53	Private	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	<=50K
4	28	Private	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	<=50K

	Male	Female	Total
Observed	21790	10771	32561
Expected	16280.50	16280.50	32561

We don't have any way to determine if there's a statistically significant difference between the two groups, and if we need to investigate further

Chi-Squared Test

we need a way to figure out what the chi-squared value represents

Chi-Squared Value

$$\chi^2 = \sum_{i=1}^k \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$$

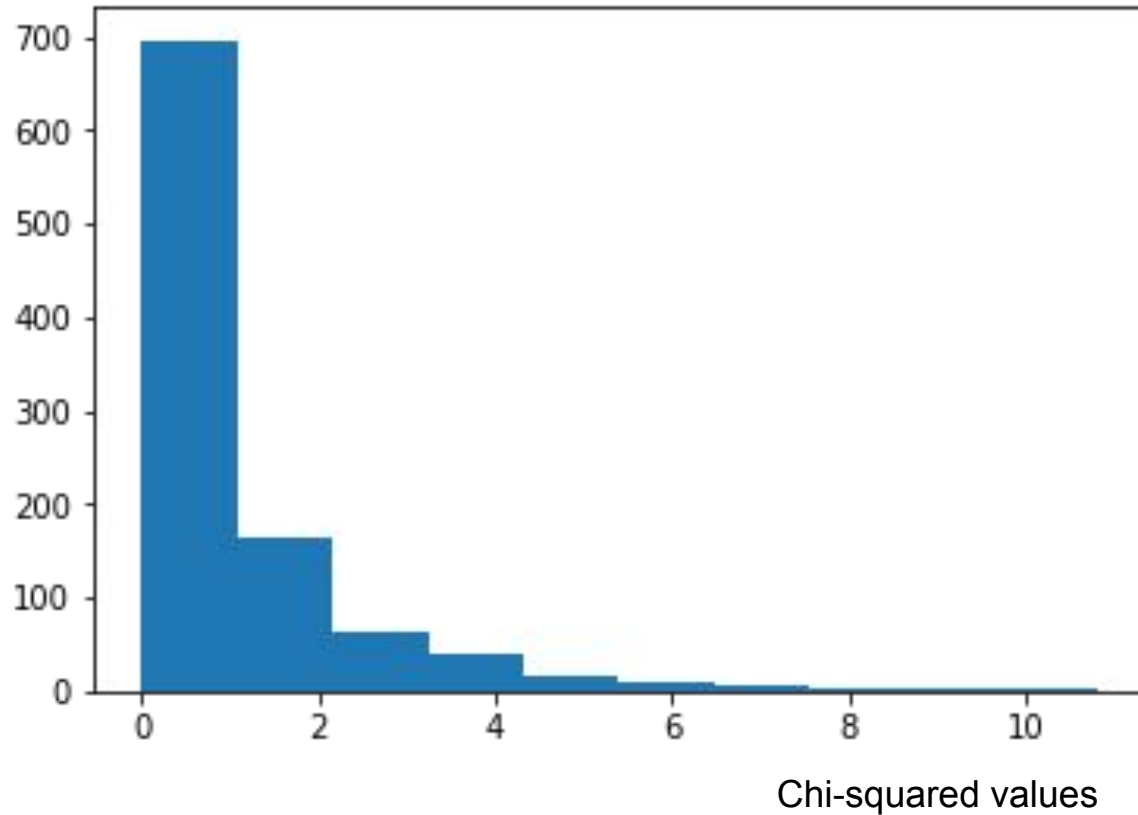
```
female_diff = (data.sex.value_counts()[1] - data.shape[0]* 0.5)**2/(data.shape[0]*0.5)
male_diff = (data.sex.value_counts()[0] - data.shape[0]* 0.5)**2/(data.shape[0]*0.5)
chi2_census = female_diff + male_diff
chi2_census
```

3728.950615767329

```
chi_squared_values = []
```

Generating a
distribution

```
for i in range(1000):  
    sequence = random((32561,))  
    sequence[sequence < .5] = 0  
    sequence[sequence >= .5] = 1  
    male_count = len(sequence[sequence == 0])  
    female_count = len(sequence[sequence == 1])  
    male_diff = (male_count - 16280.5) ** 2 / 16280.5  
    female_diff = (female_count - 16280.5) ** 2 / 16280.5  
    chi_squared = male_diff + female_diff  
    chi_squared_values.append(chi_squared)
```



How many values are greater than 3728.95 (our chi-squared value)?

p-value = 0

This would indicate that we need to investigate our data collection techniques more closely to figure out why such a result occurred.

	Male	Female	Total
Observed	21790	10771	32561
Expected	16280.50	16280.50	32561

```
import numpy as np
from scipy.stats import chisquare

observed = np.array([21790, 10771])
expected = np.array([16280.50, 16280.50])
chisquare_value, pvalue = chisquare(observed, expected)

3728.950615767329, 0
```

How two categorical columns interact?

Chi-Square statistic will test whether there is a significant difference (dependent vs independent) in the observed vs the expected frequencies of both variables.

```
pd.crosstab(data["sex"], [data["high_income"]], margins=True, normalize=True)
```

sex	high_income		All
	<=50K	>50K	
Female	9592	1179	10771
Male	15128	6662	21790
All	24720	7841	32561

sex	high_income		All
	<=50K	>50K	
Female	0.295	0.036	0.331
Male	0.465	0.205	0.669
All	0.759	0.241	1.000

- The Null hypothesis is that there is NO association between both variables.
- The Alternate hypothesis says there is evidence to suggest there is an association between the two variables.

If we reject the null hypothesis, it's an important variable (dependent) to use in your model.

To reject the null hypothesis, the calculated P-Value needs to be below a defined threshold.

Expected values

```
males_over50k = .669 * .241 * 32561
males_under50k = .669 * .759 * 32561
females_over50k = .331 * .241 * 32561
females_under50k = .331 * .759 * 32561
```

high_income	<=50K	>50K	All
sex			
Female	0.295	0.036	0.331
Male	0.465	0.205	0.669
All	0.759	0.241	1.000

```
observed = np.array([9592, 1179, 15128, 6662])
expected = np.array([8180.27, 2597.42, 16533.53, 5249.78])

chisq_value, pvalue_gender_income = chisquare(observed, expected)

1517.595316564686, 0
```

Be Pythonic

```
chisq_value, pvalue_gender_race, df, expected = chi2_contingency(pd.crosstab(data["sex"],  
                                                                              [data["high_income"]]))
```

The function return:

- Chi-squared value
- P-value
- Degree of freedom
- Expected value

Feature selection

```
# all chi-squared values
chi_dict = {}

# only categorical columns are used in chi-squared test
columns = income.select_dtypes("object").columns.to_list()[:-1]

# eliminate the high_income column
for name in columns:
    chisq_value, pvalue_all, df, expected = chi2_contingency(pd.crosstab(income[name], [income["high_income"]]))
    chi_dict[name] = (chisq_value, pvalue_all)

sorted(chi_dict.items(), key=lambda kv: kv[1][0], reverse=True)

[('relationship', (6699.07689685885, 0.0)),
 ('marital_status', (6517.741653663022, 0.0)),
 ('education', (4429.653302288619, 0.0)),
 ('occupation', (4031.974280247181, 0.0)),
 ('sex', (1517.813409134445, 0.0)),
 ('workclass', (1045.7085997281692, 2.026505431120716e-220)),
 ('race', (330.9204310085741, 2.305960610160958e-70)),
 ('native_country', (317.2303857833171, 2.2113858852543023e-44))]
```



Feature selection

- Univariate Selection
- Recursive Feature Elimination
- Feature Importance
- Principal Component Analysis (PCA)

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

# univariate selection
fsel_model = SelectKBest(score_func=chi2, k=5)
new_income = fsel_model.fit_transform(income.drop(labels=["high_income"],axis=1),
                                     income.high_income)

from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression

# recursive feature elimination
model = LogisticRegression(solver="liblinear")
fsel = RFE(model,5)
new_income = fsel.fit_transform(income.drop(labels=["high_income"],axis=1),
                               income.high_income)
```

```
# columns that were selected
```

```
# univariate selection
```

```
income.loc[:,fsel_model.get_support()].columns
```

```
['age', 'fnlwgt', 'capital_gain', 'capital_loss', 'hours_per_week']
```

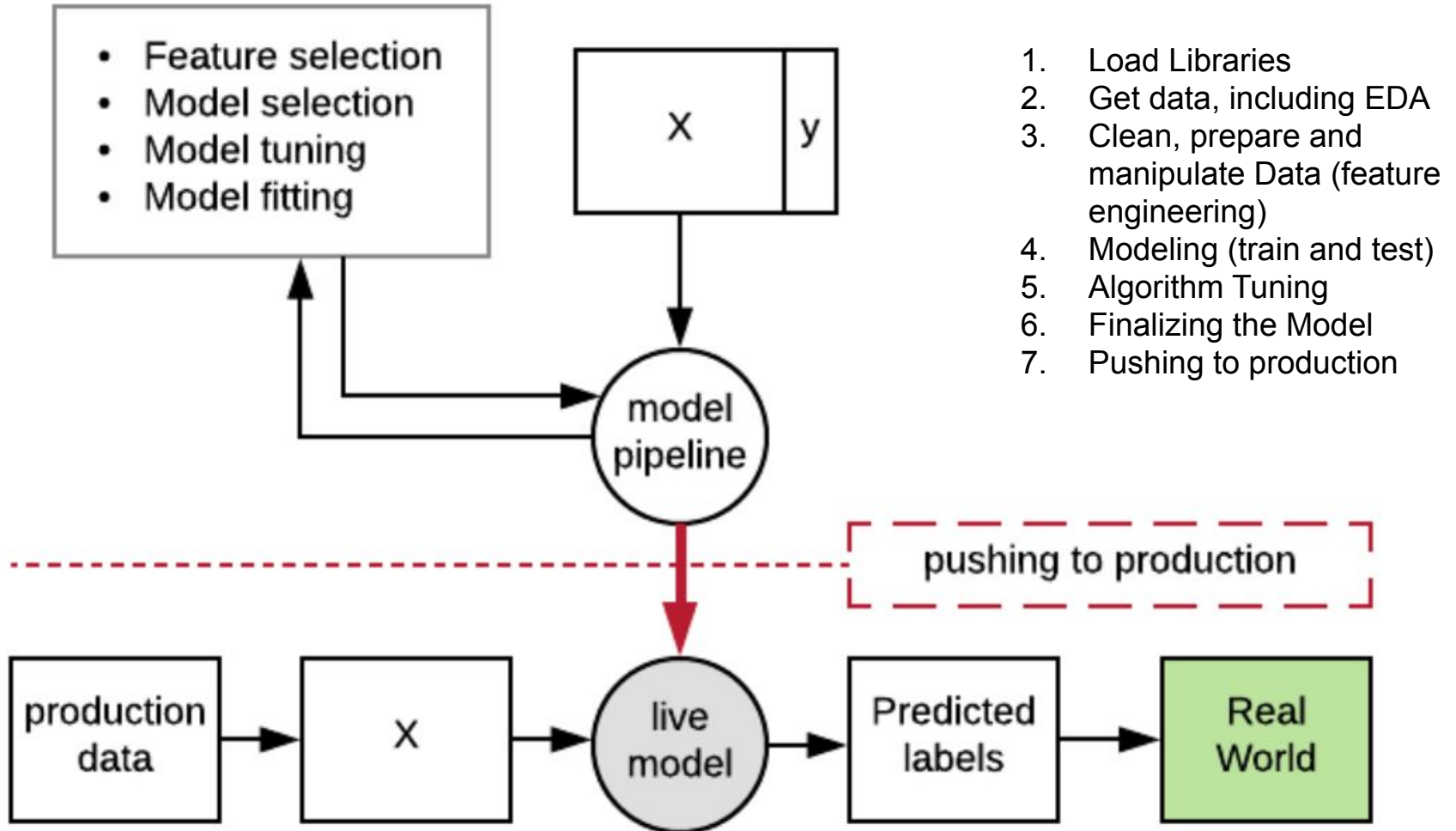
```
# recursive feature elimination
```

```
income.loc[:,fsel.support_].columns
```

```
['education_num', 'marital_status', 'relationship', 'race', 'sex']
```

Everything ends in
Pipelines





Section 2.4

Exercise:

- Tuning Decision Tree
 - "Max_depth": range(3, 21, 3)
- Tuning Logistic Regression
 - "Penalty": L1, L2
- Tuning Features Selection
- Analyze overfitting

