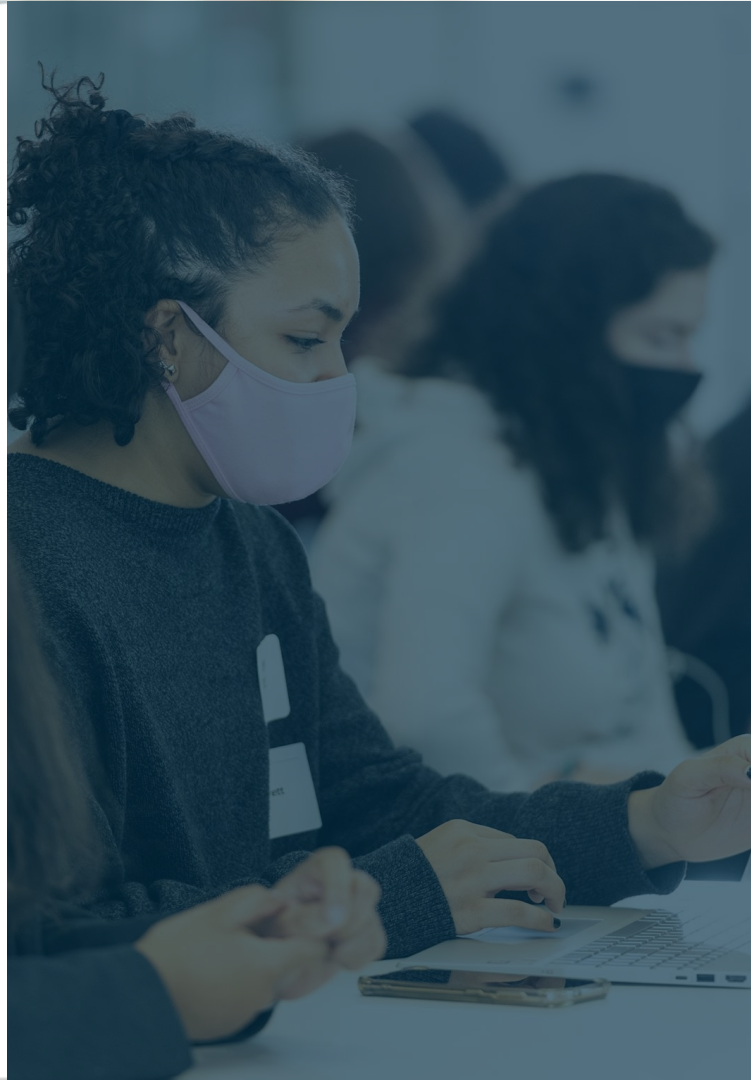
Machine Learning Foundations

Lab 2

Today's Agenda

Icebreaker	(10 minutes)
Week 2 Overview + Q&A	(30 minutes)
Breakout Groups: Big Picture Questions	(20 minutes)
Class Discussion	(10 minutes)
Break	(10 minutes)
Breakout Groups: Lab Assignment Working Session	(80 minutes)
Working Session Debrief	(10 minutes)
Concluding Remarks & Survey	(10 minutes)



Icebreaker: “Get Pumped!”

Icebreaker: Get Pumped!

Objectives:

- Share a “pump up song” that inspires, builds your confidence, or otherwise brings you energy

Icebreaker: Get Pumped! Instructions

- Think about your own “pump up song”. Remember, this song might:
  - Inspire you, give you energy, or confidence
  - Be related to a personal value that is important to you
  - Remind you of a person, place, or thing

- Once you’ve picked your Pump Up Song, enter it into the Google Form linked in the chat. The Break Through Tech team will make a playlist with your Pump Up Songs for upcoming Maker Day and other events!

- When you’re finished, share your Pump Up Song in the chat and why it gives you energy.

https://forms.gle/R8PvWbCTn1oKgFZV8

Icebreaker: Get Pumped! Share out

- Who wants to share a clip from their “Pump Up” song?
- Why is this song meaningful to you?

https://docs.google.com/spreadsheets/d/1-Ef6Wsrd5dipaxsWoyq4ktpFvbKJ1A3XuJB6KclGRwQ/edit?resourcekey#gid=1549963278

Week 2 Overview + Q&A

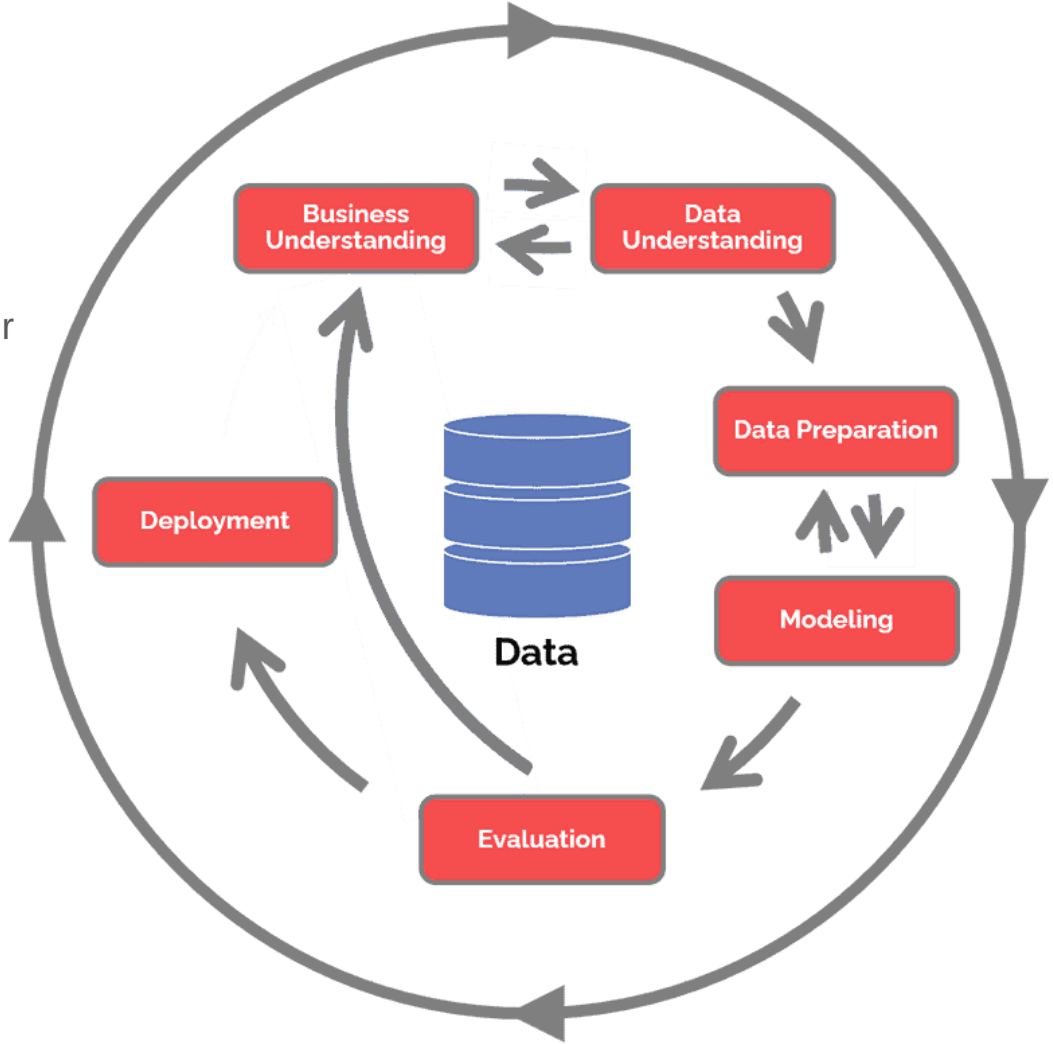
Week 2 Overview

This week you explored a number of topics. To refresh your memory, your goals were to:

- Build a data set suitable for ML applications.
- Create an appropriate label for supervised learning.
- Create features that are suitable for ML applications.
- Use exploratory analysis to understand your data.
- Identify and fix issues with your data.

Recap: ML Life Cycle

- CRISP-DM
  - Cross-Industry Standard Process for Data Mining



Data Matrix

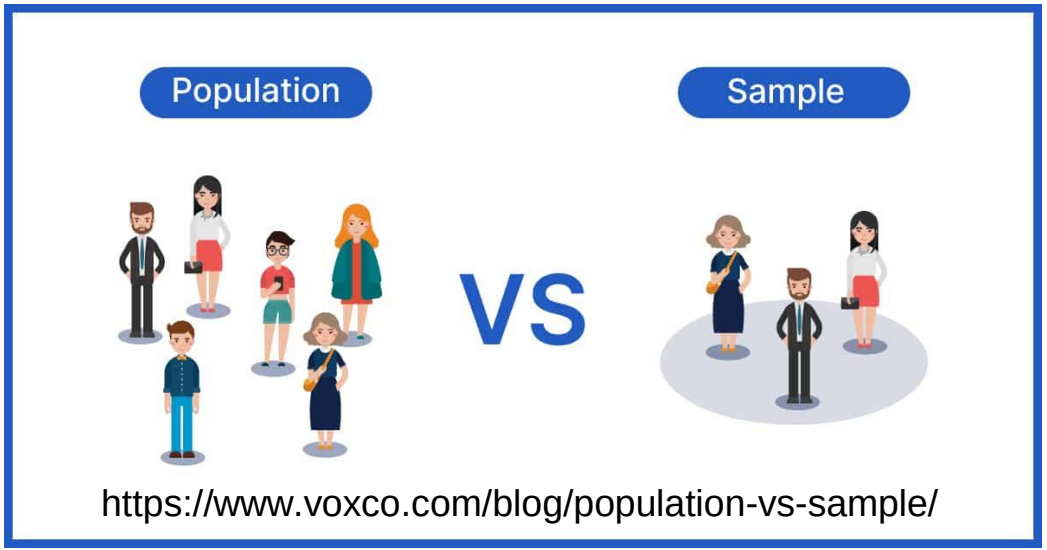
- Rows: data points
- Columns: features/attributes

Sample dataset

Name	Income	Credit Score	Occupation	Job Sector	Loan Status
John Doe	$76,000	650	Engineer	Engineering	Good
Gill Bates	$85,000	760	Nurse	Healthcare	Defaulted
Jane Doe	“95000.00”	0	Banker	Financial	Good
John Doe	$76,000	650	Engineer	Engineering	Good
Melon Usk		810	Flight Attendant	Transportation	Excellent
Barren Wuffet	5000/mo	35000	Contractor	Construction	Defaulted

Sample vs. Population

- Population:
  - All the data points, e.g., everyone
- Samples:
  - A subset of data points, e.g., the ones in your database



Data Exploration

- Understand your data
  - What are input and what are output
  - Statistics summary from the data, e.g., mean, standard deviation
  - Visualization of the data, e.g., scatter plot, histogram

Sample dataset

Name	Income	Credit Score	Occupation	Job Sector	Loan Status
John Doe	$76,000	650	Engineer	Engineering	Good
Gill Bates	$85,000	760	Nurse	Healthcare	Defaulted
Jane Doe	“95000.00”	0	Banker	Financial	Good
John Doe	$76,000	650	Engineer	Engineering	Good
Melon Usk		810	Flight Attendant	Transportation	Excellent
Barren Wuffet	5000/mo	35000	Contractor	Construction	Defaulted

Data Preparation/Data preprocessing

- What we want for a supervised task?
  - $D = \{(x, y)\}$
  - Examples:
    - Salary prediction: x are a collection of features related to salary; y is salary

← Features →					Label
Position	**Experience**	**Skill**	**Country**	**City**	**Salary ($)**
Developer	0	1	USA	New York	103100
Developer	1	1	USA	New York	104900
Developer	2	1	USA	New York	106800
Developer	3	1	USA	New York	108700
Developer	4	1	USA	New York	110400
Developer	5	1	USA	New York	112300
Developer	6	1	USA	New York	114200
Developer	7	1	USA	New York	116100
Developer	8	1	USA	New York	117800
Developer	9	1	USA	New York	119700
Developer	10	1	USA	New York	121600

https://www.i2tutorials.com/what-do-you-mean-by-features-and-labels-in-a-dataset/

Feature Engineering

- Turn everything into a nice numerical matrix

Technique	Description	Input	Output
Binary indicators	Transform data to binary based on meeting a true/false condition	Categorical/Numeric	Binary (0/1)
One-hot-encoding	Transform K categories into K-1 binary indicators	Categorical	Binary (0/1)
Functional Transforms	Transform a numeric input X into a new numeric value based on f(X)	Numeric	Numeric
Interaction Terms	Take the multiplication of two numeric types	Numeric	Numeric

Examples

- Occupation
  - If there are 5 occupations in total, make a 5-dimensional binary vector (10000, 01000, etc.)
- Income
  - Make a log transformation

Sample dataset

Name	Income	Credit Score	Occupation	Job Sector	Loan Status
John Doe	$76,000	650	Engineer	Engineering	Good
Gill Bates	$85,000	760	Nurse	Healthcare	Defaulted
Jane Doe	“95000.00”	0	Banker	Financial	Good
John Doe	$76,000	650	Engineer	Engineering	Good
Melon Usk		810	Flight Attendant	Transportation	Excellent
Barren Wuffet	5000/mo	35000	Contractor	Construction	Defaulted

More examples

- Problem: Recommend new twitter users that I should follow

Why do I follow someone?

1. This person looks interesting.
2. The person posts about topics that I am interested in.
3. This person followed me.
4. People I follow also follow this person
5. This person is very popular with a lot of followers

TwitterID	topic_overlap_pct	user_follows_me	num_follows_follow	num_followers
1234	0.1	0	5	1199
1345	0.2	0	1	230
1099	0.7	1	24	100
2011	0.8	1	46	10145
2900	0.5	0	51	51454

Know your data again

- Basic statistics using Pandas describe()

```
In [27]: describe_vars = ['isbuyer','buy_freq','expected_time_buy','uniq_urls','num_checkins','y_buy']
         df[describe_vars].describe()
```

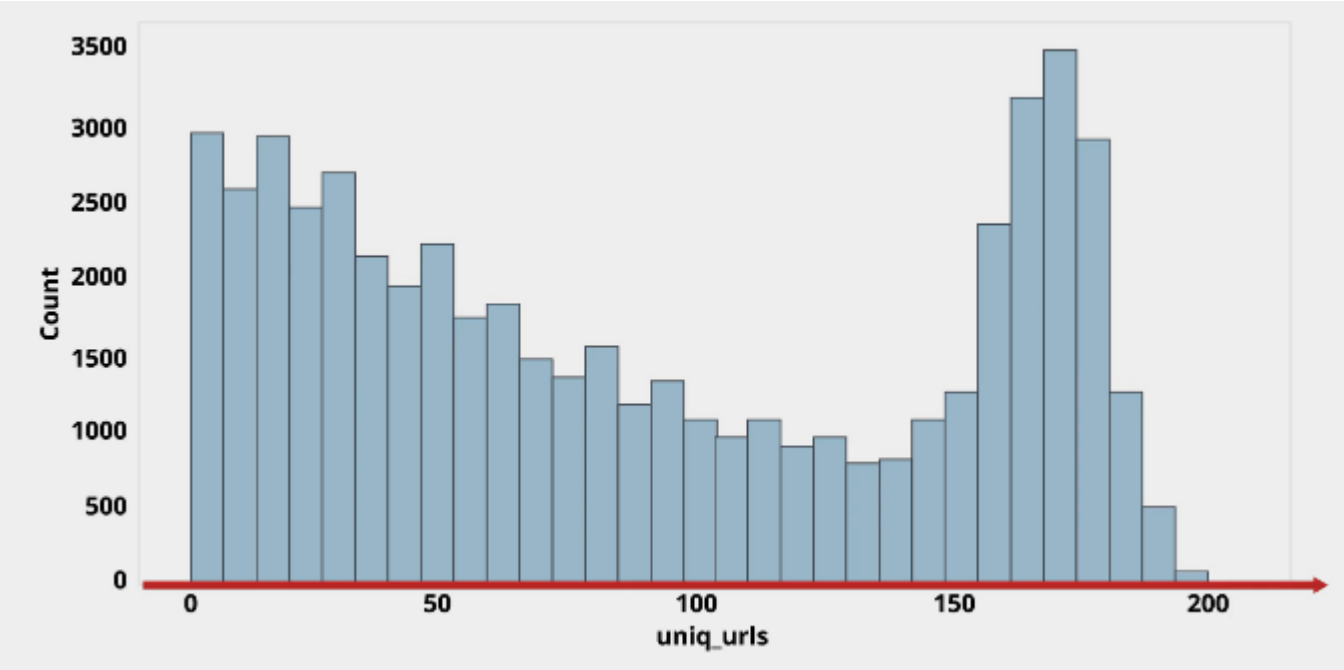
Out[27]: Output

	isbuyer	buy_freq	expected_time_buy	uniq_urls	num_checkins	y_buy
count	54584.000000	2327.000000	54584.000000	54584.000000	54584.000000	54584.000000
mean	0.042632	1.240653	-0.198040	86.569343	720.657592	0.004635
std	0.202027	0.782228	4.997792	61.969765	1275.727306	0.067924
min	0.000000	1.000000	-181.923800	-1.000000	1.000000	0.000000
25%	0.000000	1.000000	0.000000	30.000000	127.000000	0.000000
50%	0.000000	1.000000	0.000000	75.000000	319.000000	0.000000
75%	0.000000	1.000000	0.000000	155.000000	802.000000	0.000000
max	1.000000	15.000000	84.285710	206.000000	37091.000000	1.000000

Standard Distribution Statistics

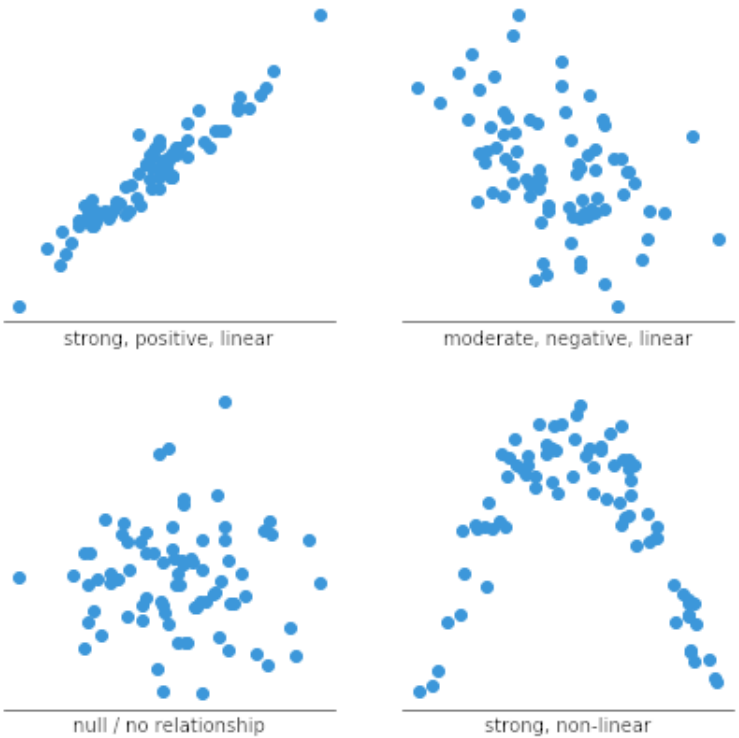
Visualization – univariate plotting

- Histogram



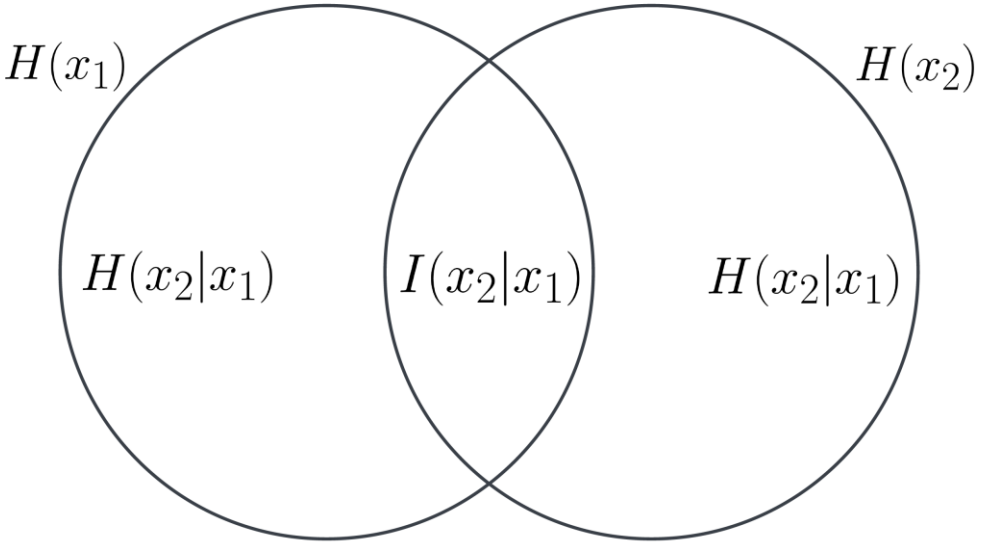
Visualization – bivariate plotting

- Scatter plot



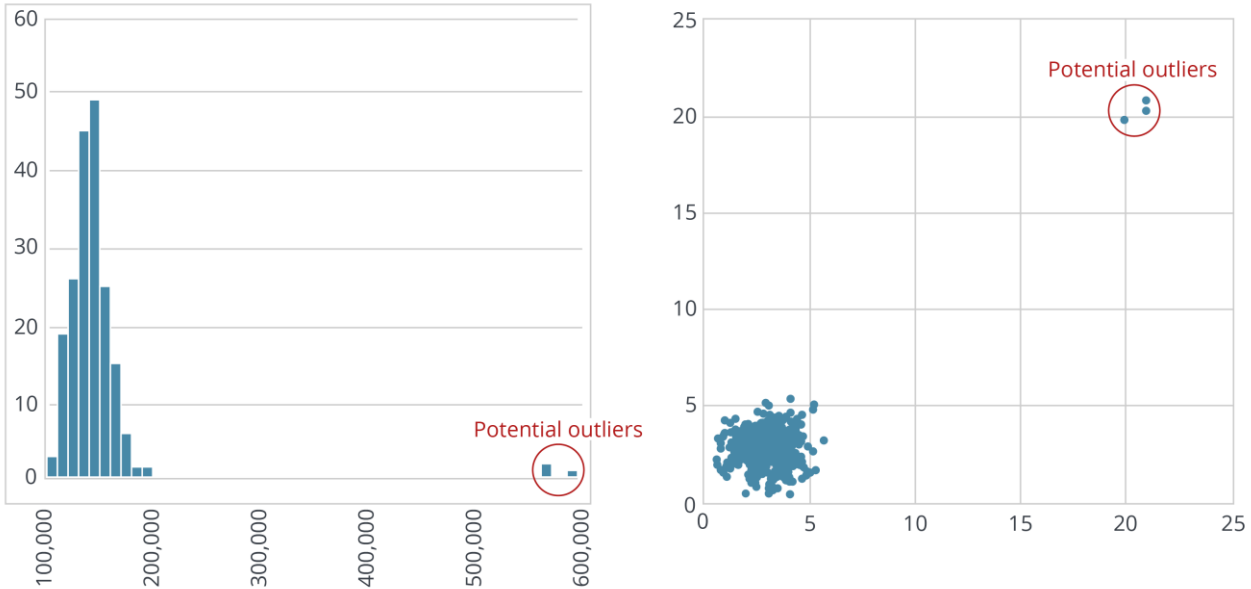
Dependency between variables

- Correlation and Covariance
- Mutual information



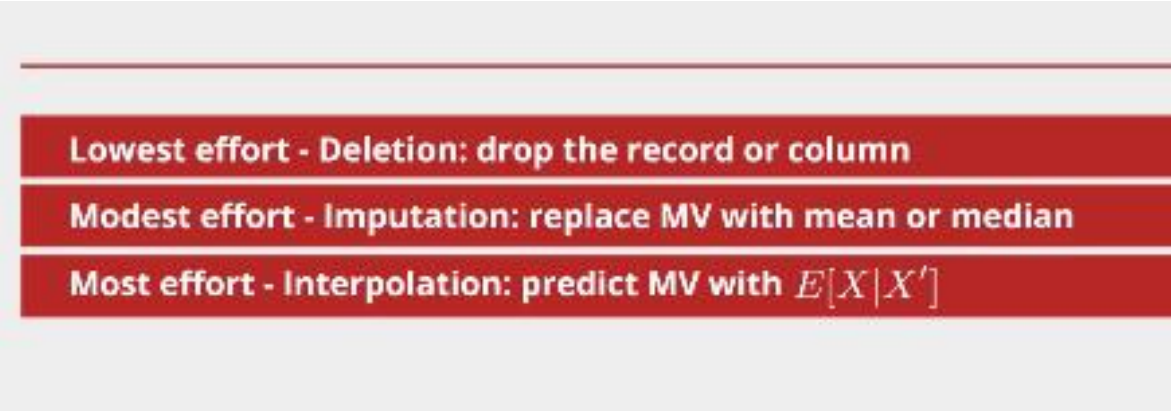
Data cleaning

- Outlier detection



Data cleaning

- Missing values



Build Your Data Matrix: Applying Filters and Building a Balanced Dataset

Pandas DataFrame: Common Methods :

- DataFrames
  - df.head(10)
  - Properties:
    - df.shape
    - df.index
    - df.loc[5]
    - df.dtypes – returns type of each column

```
import pandas as pd

data = {'Name': ['John', 'Jane', 'Mike'],
        'Age': [25, 30, 35],
        'City': ['New York', 'London', 'Paris']}

df = pd.DataFrame(data)

print(df.index)
```

Output:

```
arduino                                                    Copy code

RangeIndex(start=0, stop=3, step=1)
```

Sampling:

- Taking a Sample of Data
  - indices = np.random.choice(df.index, size=100, replace=False)
    df_subset = df.loc[indices]
  - df_subset = df.sample(100)

Build Your Data Matrix: Applying Filters and Building a Balanced Dataset

- Filtering data
  - condition1 = df['workclass'] == 'Private' - returns series of True/False values
  - condition2 = df_subset['sex_selfID'].isnull() – returns True/False values
  - df_filter = df[condition1 & ~condition2]

- Groups within a column
  - df_subset['sex_selfID'].unique() – returns unique values in column
  - counts = df_subset['sex_selfID'].value_counts()
  - df_subset.groupby(['sex_selfID', 'label']).size()

- Modifying/Merging labels
  - condition = columns_not_self_employed & columns_not_null
    df['workclass'] = np.where(condition, 'Not-self-emp', df['workclass'])

python Copy code

```python
import pandas as pd

data = {'workclass': ['Private', 'Self-emp', 'Private', 'Self-emp', 'Private
        'sex_selfID': ['Male', 'Female', None, 'Male', 'Female'],
        'age': [25, 30, 35, 40, 45]}

df = pd.DataFrame(data)

condition1 = df['workclass'] == 'Private'
condition2 = df['sex_selfID'].isnull()

df_filter = df[condition1 & ~condition2]

print(df_filter)
```

Output:

vbnet Copy code

```
  workclass sex_selfID  age
0   Private       Male   25
2   Private       None   35
4   Private     Female   45
```

Build Your Data Matrix: Applying Filters and Building a Balanced Dataset

- Filtering data
  - condition1 = df['workclass'] == 'Private' - returns series of True/False values
  - condition2 = df_subset['sex_selfID'].isnull() – returns True/False values
  - df_filter = df[condition1 & ~condition2]

- Groups within a column
  - df_subset['sex_selfID'].unique() – returns unique values in column
  - counts = df_subset['sex_selfID'].value_counts()
  - df_subset.groupby(['sex_selfID', 'label']).size()

- Modifying/Merging labels
  - condition = columns_not_self_employed & columns_not_null
    df['workclass'] = np.where(condition, 'Not-self-emp', df['workclass'])

python Copy code

```python
import pandas as pd

data = {'sex_selfID': ['Male', 'Male', 'Female', 'Female', 'Male', 'Female',
        'label': ['A', 'B', 'A', 'A', 'B', 'B', 'A']}

df_subset = pd.DataFrame(data)

counts = df_subset['sex_selfID'].value_counts()

print(counts)
```

Output:

vbnet Copy code

```
Female        3
Male          3
Non-Binary    1
Name: sex_selfID, dtype: int64
```

python Copy code

```python
import pandas as pd

data = {'sex_selfID': ['Male', 'Male', 'Female', 'Female', 'Male', 'Female',
        'label': ['A', 'B', 'A', 'A', 'B', 'B', 'A']}

df = pd.DataFrame(data)

grouped_data = df.groupby(['sex_selfID', 'label']).size()

print(grouped_data)
```

Output:

css Copy code

```
sex_selfID  label
Female      A        2
            B        2
Male        A        1
            B        1
Non-Binary  A        1
dtype: int64
```

Build Your Data Matrix: Applying Filters and Building a Balanced Dataset

- Filtering data
  - condition1 = df['workclass'] == 'Private' - returns series of True/False values
  - condition2 = df_subset['sex_selfID'].isnull() – returns True/False values
  - df_filter = df[condition1 & ~condition2]

- Groups within a column
  - df_subset['sex_selfID'].unique() – returns unique values in column
  - counts = df_subset['sex_selfID'].value_counts()
  - df_subset.groupby(['sex_selfID', 'label']).size()

- Modifying/Merging labels
  - condition = columns_not_self_employed & columns_not_null
    df['workclass'] = np.where(condition, 'Not-self-emp', df['workclass'])

python

```python
import pandas as pd
import numpy as np

data = {'workclass': ['Private', 'Self-emp', None, 'Private', 'Self-emp'],
        'age': [25, 30, 35, 40, 45]}

df = pd.DataFrame(data)

columns_not_self_employed = df['workclass'] != 'Self-emp'
columns_not_null = df['workclass'].notnull()

condition = columns_not_self_employed & columns_not_null

df['workclass'] = np.where(condition, 'Not-self-emp', df['workclass'])

print(df)
```

Output:

rust

```
      workclass  age
0  Not-self-emp   25
1      Self-emp   30
2           None   35
3  Not-self-emp   40
4      Self-emp   45
```

Create Labels and Features: Clean and Convert Data

- Cast column to type int
  - df['col'] = df['col'].astype(int)

- Creating new sample that doesn't include original sample
  - df_never_sampled = df.drop(labels=df_subset.index, axis=0, inplace=False)
  - df = df.drop(['col1', col2'], axis=1) - drop col1 and col2 from df

- Ordering categorical data

  edu = ['Preschool', '1st-4th', '5th-6th', '7th-8th', '9th', '10th', '1th', '12th', 'HS-grad', 'Prof-school', 'Assoc-acdm', 'Assoc-voc', 'Some-college', 'Bachelors', 'Masters', 'Doctorate']

  df['education'] = pd.Categorical(df['education'], ordered=True, categories=edu)

- Converting categorical data to binary (Feature engineering: one-hot encoding)
  - df_binary = pd.get_dummies(df)

python Copy code

```python
import pandas as pd

# Sample DataFrame
data = {'education': ['Bachelors', 'Masters', 'HS-grad', 'Doctorate', 'Bache
df = pd.DataFrame(data)

# Define education categories for ordering
edu = ['Preschool', '1st-4th', '5th-6th', '7th-8th', '9th', '10th', '11th',

# Convert 'education' column to ordered categorical variable
df['education'] = pd.Categorical(df['education'], ordered=True, categories=e

# Perform one-hot encoding
df_binary = pd.get_dummies(df)

print(df_binary)
```

Output:

Copy code

```
ion_Some-college  education_Bachelors  education_Masters  education_Doctorate
                0                    1                  0
                0                    0                  1
                0                    0                  0
                0                    0                  0
                0                    1                  0
```

Explore Your Data: Common Data Exploration Functions

- df.describe() – returns per column statistics about df
- Finding column with highest variance
  - df_summ = df.describe()
  - df_summ.loc['std'].idxmax(axis=1)
  - df_summ.idxmax(axis = 1)['std']

- Does any column have negative values
  - np.any(df_summ.loc['min'] < 0)
  - np.any(condition)

```
In [11]: df_summ = df.describe()
         df_summ
```

Out[11]:

	age	fnlwgt	education-num	capital-gain	capital-loss	hours-per-week
count	7000.000000	7.000000e+03	7000.000000	7000.000000	7000.000000	7000.000000
mean	38.596714	1.924335e+05	10.049857	1079.000429	84.970286	40.107143
std	13.745594	1.063365e+05	2.580982	7011.160679	400.142351	12.323946
min	17.000000	1.882700e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.202478e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.821170e+05	10.000000	0.000000	0.000000	40.000000
75%	47.000000	2.402370e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.268339e+06	16.000000	99999.000000	4356.000000	99.000000

```
In [16]: df_summ.loc['std'].idxmax(axis = 1)
Out[16]: 'fnlwgt'
```

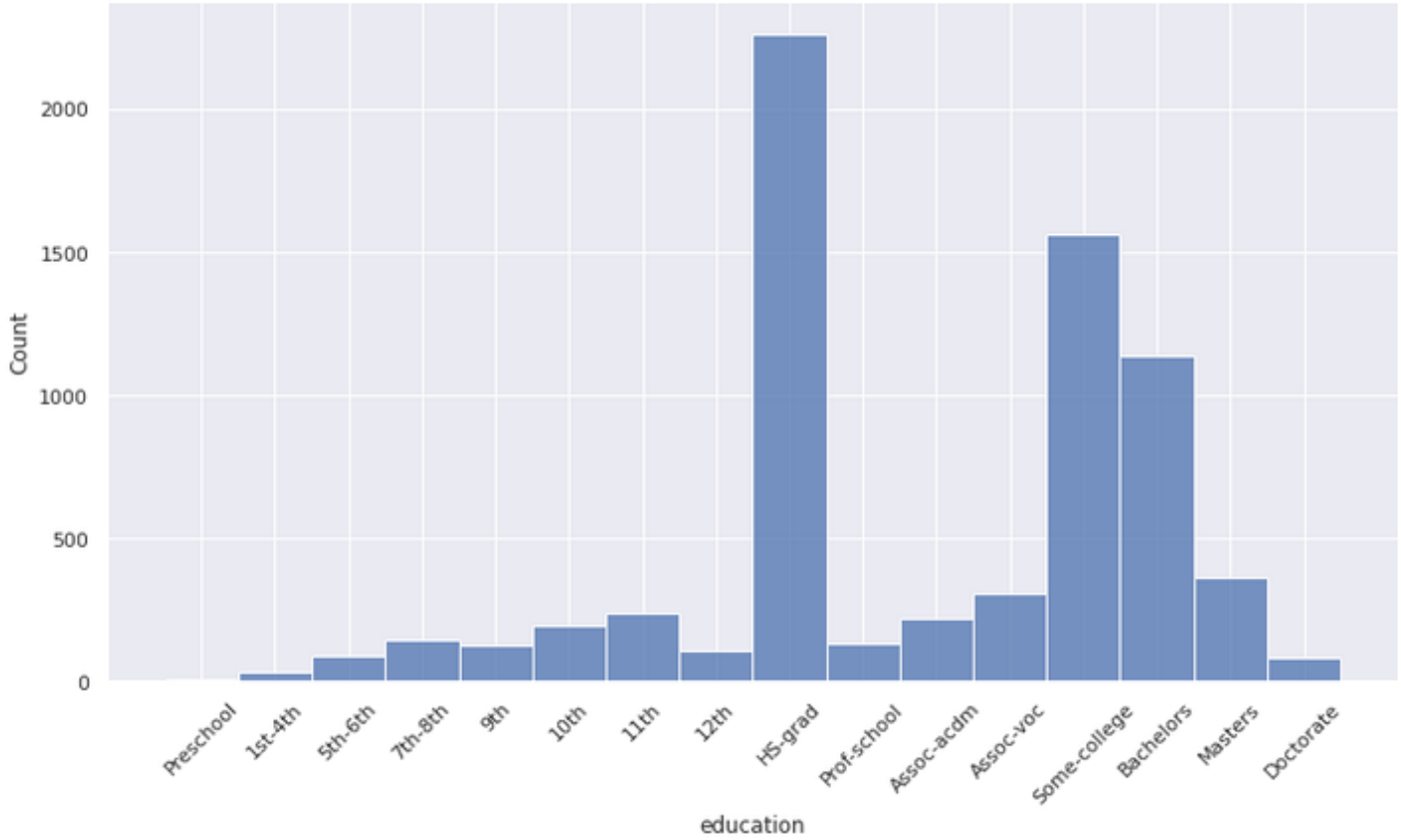
Explore Your Data:
Visualize Data using Seaborn and Matplotlib

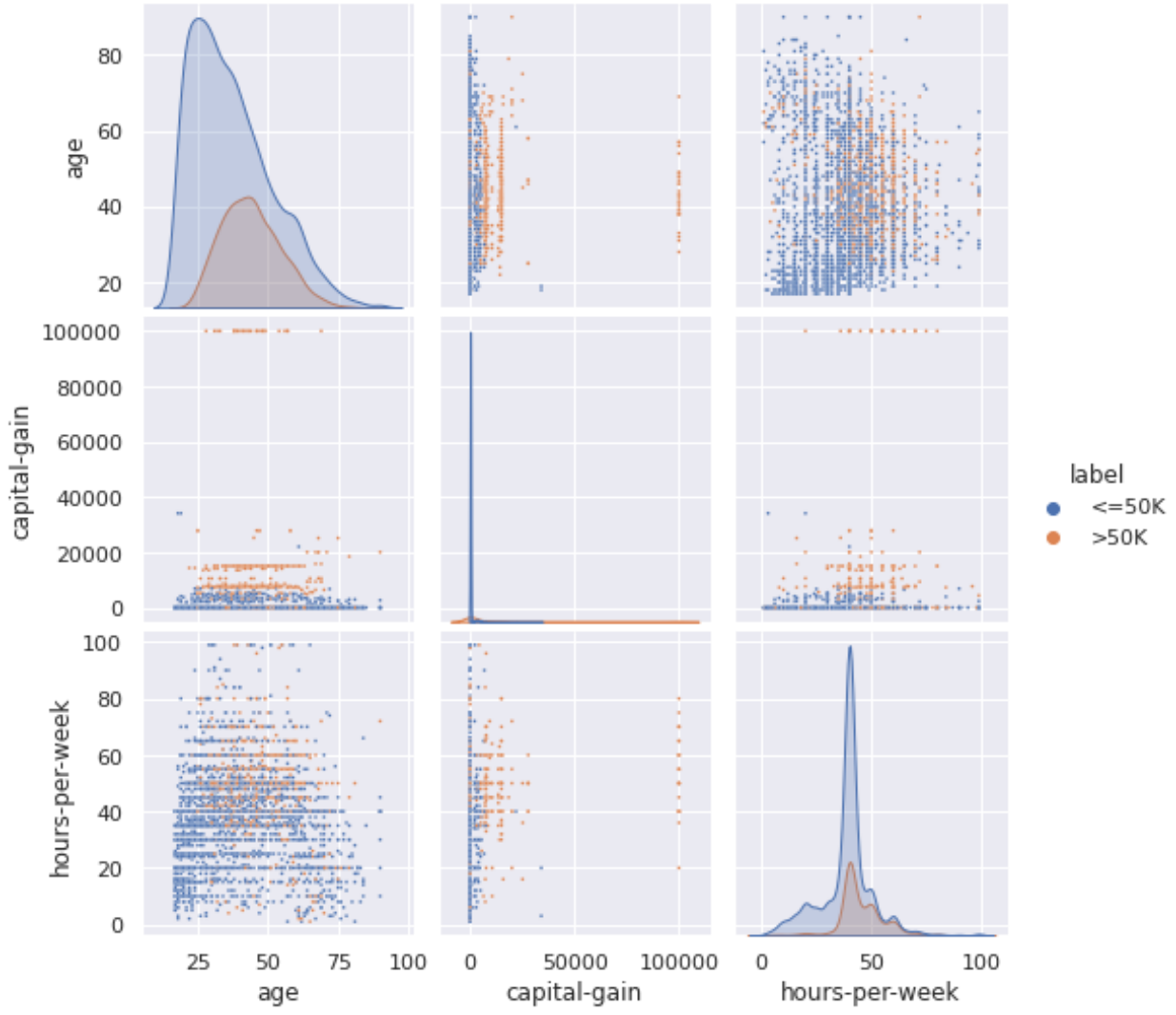
- Histogram, Pairplot, and Barplot
  - sns.histplot(data=df, x="age")
  - sns.pairplot(data=df, hue='label') – pairwise scatterplots of each pair of columns, and color by 'label'
  - sns.pairplot(data=df, kind='kde', corner=True) - use kernel density estimator type plot
  - sns.barplot(data = df_sub, x='education',' y='label') - shows average of labels for each value of x

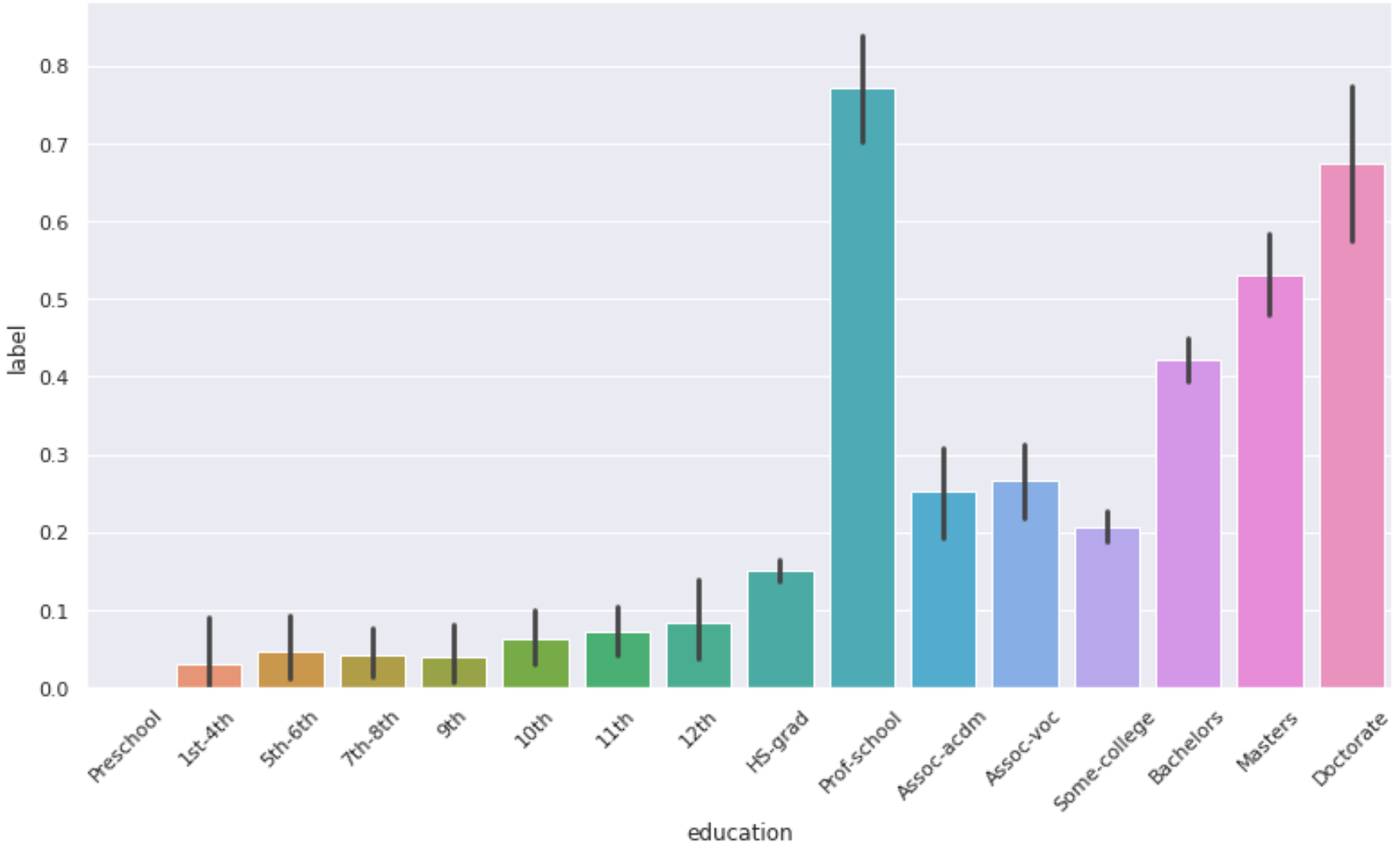
- Editing figure
  - plt.figure(figsize=(13,7)) – set width, height in inches
  - plt.ylim(0, 600) – can use this to zoom in on a smaller region of the y axis
  - plt.xticks(rotation=45)

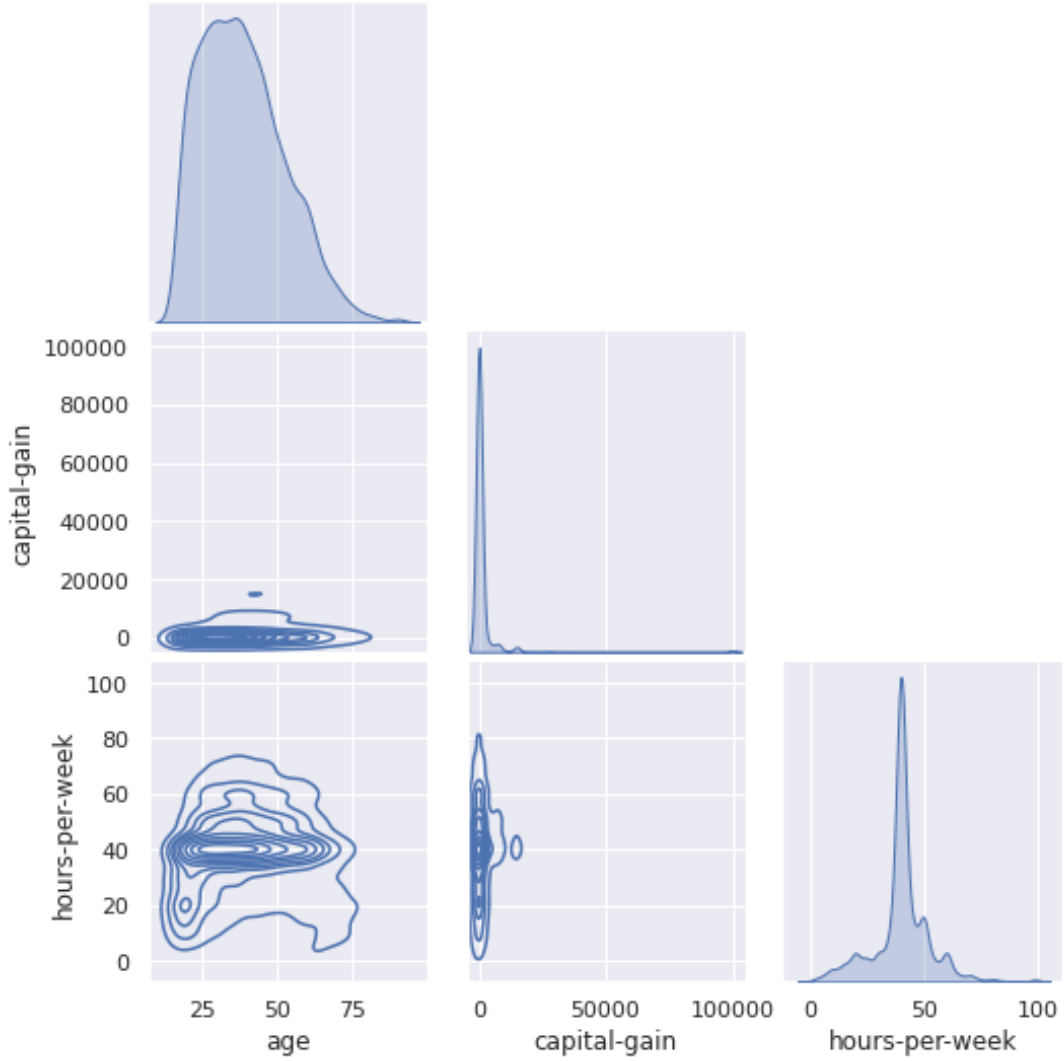
In [18]:

```
fig2 = plt.figure(figsize=(13,7))
ax = sns.histplot(data=df, x="education")
t2 = plt.xticks(rotation=45)
```




age
capital-gain
hours-per-week
label
<=50K
>50K


label
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Preschool
1st-4th
5th-6th
7th-8th
9th
10th
11th
12th
HS-grad
Prof-school
Assoc-acdm
Assoc-voc
Some-college
Bachelors
Masters
Doctorate
education


100000
80000
60000
40000
20000
0
capital-gain
100
80
60
40
20
0
hours-per-week
25
50
75
age
0
50000
100000
capital-gain
0
50
100
hours-per-week

Explore Your Data: Correlation

- df.corr()['label'] – returns correlation of each feature with the label
  - exclude = ['label', 'non_winsorized_label']
    corrs = df.corr()['label'].drop(exclude, axis=0)

- Sort correlations in descending order
  corrs_sorted = corrs.sort_values(ascending=False)
  col_names = corrs_sorted.index - returns column names in descending order of correlation with label.

Finding Outliers and Missing Data: Replacing Outliers

- Value corresponding to x percentile.
  - val = np.percentile(df['col_name'], x)

- Scipy to winsorize data (remove outliers)
  - scipy.stats as stats
  - df['col-win'] = stats.mstats.winsorize(df['col'], limits=[0.01, 0.01]) – replace lower and top 1% with values at 1% and 99% respectively.

- zscore = (value – mean)/std - measures how far away each point is from the mean – zscore of 1 means point is 1 std away from mean.
  - zscores = stats.zscore(df['col'])

Finding Outliers and Missing Data: Replacing Missing Values

- Find and count missing values
  - df.isnull()
  - nan_count = np.sum(df.isnull())

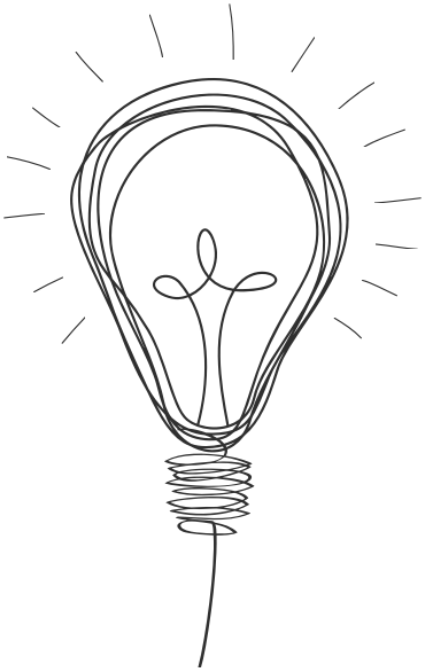
- Replace missing values with mean

  mean_ages=df['age'].mean()

  df['age'].fillna(value=mean_ages, inplace=True)

Questions & Answers

What questions do you have about the online content this week?



Breakout Groups: Big Picture Questions

Big Picture Questions

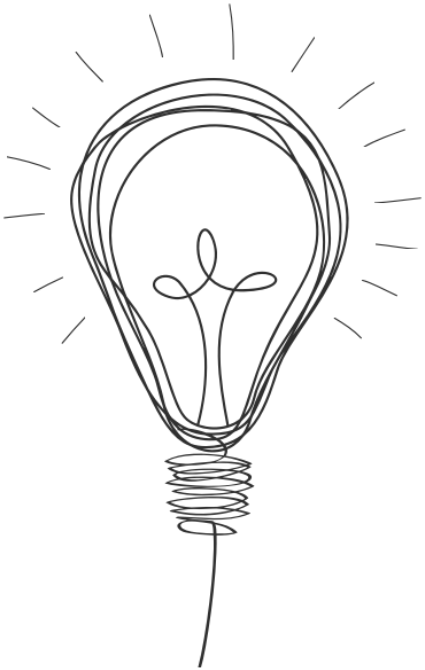
You have 20 minutes to discuss the following questions within your breakout groups:

- Why is data preparation so important to the machine learning development process?
- Considering that data preparation often takes the majority of model development time, how would you communicate to stakeholders (bosses, product managers, leadership, etc.) why you need to budget time for data preparation?
- What does it mean to have a “modeling dataset”?
- What is the difference between nominal data and ordinal data? Explain with an example.
- Why is data visualization an important part of the data preparation process?
- Name a few libraries used for data analysis and visualization and explain when you would use each library.

Class Discussion

Class Discussion: Responses to Big Picture Questions

Let’s hear your classmates’ responses.



Break!

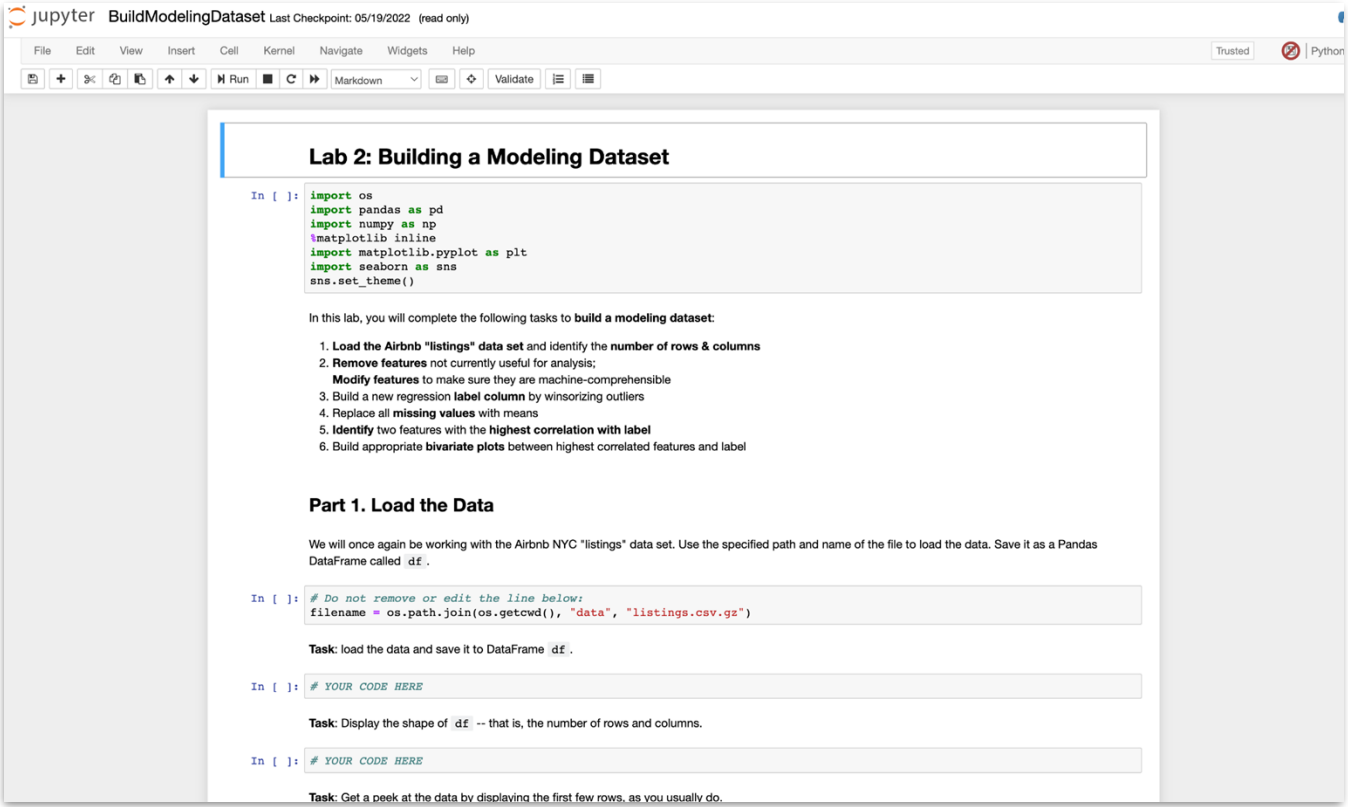
Breakout Groups:
Lab Assignment

Lab 2

In this lab, you will:

- Load data and identify the number of records & columns
- Remove features that are not currently useful for analysis
- Modify features to make sure they are machine-comprehensible
- Build a new regression label column by winsorizing outliers
- Replace all missing values with means
- Identify two variables with the highest correlation with the label
- Build appropriate bivariate plots between the highest correlated features and the label

Lab 2


jupyter BuildModelingDataset Last Checkpoint: 05/19/2022 (read only)
File Edit View Insert Cell Kernel Navigate Widgets Help
Trusted Python
Run Markdown Validate
Lab 2: Building a Modeling Dataset
In [ ]:
import os
import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme()
In this lab, you will complete the following tasks to build a modeling dataset:
1. Load the Airbnb "listings" data set and identify the number of rows & columns
2. Remove features not currently useful for analysis;
Modify features to make sure they are machine-comprehensible
3. Build a new regression label column by winsorizing outliers
4. Replace all missing values with means
5. Identify two features with the highest correlation with label
6. Build appropriate bivariate plots between highest correlated features and label
Part 1. Load the Data
We will once again be working with the Airbnb NYC "listings" data set. Use the specified path and name of the file to load the data. Save it as a Pandas DataFrame called df .
In [ ]:
# Do not remove or edit the line below:
filename = os.path.join(os.getcwd(), "data", "listings.csv.gz")
Task: load the data and save it to DataFrame df .
In [ ]:
YOUR CODE HERE
Task: Display the shape of df -- that is, the number of rows and columns.
In [ ]:
YOUR CODE HERE
Task: Get a peek at the data by displaying the first few rows, as you usually do.

Working Session Debrief

Lab Debrief

So far,

- What did you enjoy about this lab?
- What did you find difficult about this lab?
- What questions do you still have about this lab?
- How did you approach problem-solving during the exercise?
- What would you do differently if you were to repeat the exercise?

Concluding Remarks

Concluding Remarks

- Key takeaways
- Additional resources

Next week

In the following week, you will:

- Define the core foundational elements of model training and evaluation
- Develop intuition for different classes of algorithms
- Analyze the mechanics of two popular supervised learning algorithms: decision trees and k-nearest neighbors
- Develop intuition on trade-offs between different algorithmic choices

And in the lab, you will:

- Convert categorical features to one-hot encoded values.
- Train decision tree classifiers with various hyperparameter values.
- Train KNN classifiers with various hyperparameter values.
- Visualize the models' accuracies.

Content + Lab Feedback Survey

Content + Lab Feedback Survey

To complete your lab, please answer the following questions about BOTH your online modules and your lab experience. Your input will help pay it forward to the Break Through Tech student community by enabling us to continuously improve the learning experience that we provide to our community.

Thank you for your thoughtful feedback!

https://forms.gle/xdfN3Vy1BYMUHFvu8