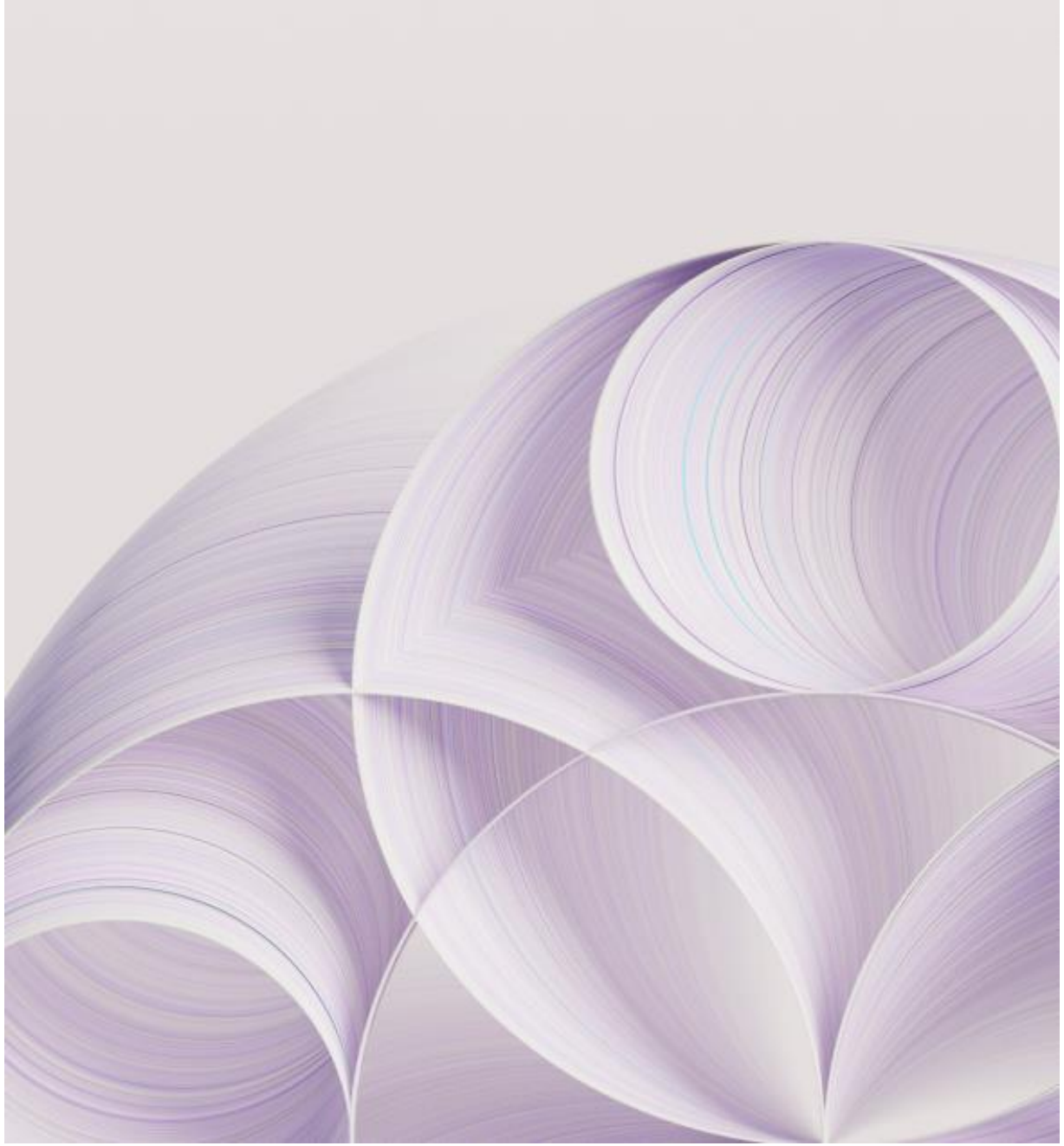
Automated Survey Comment Coding with Naïve Bayes & Python

By Tyler Waite
Advisory Data Scientist, AI, Automation & Data Platform
Finance and Operations

October 2024



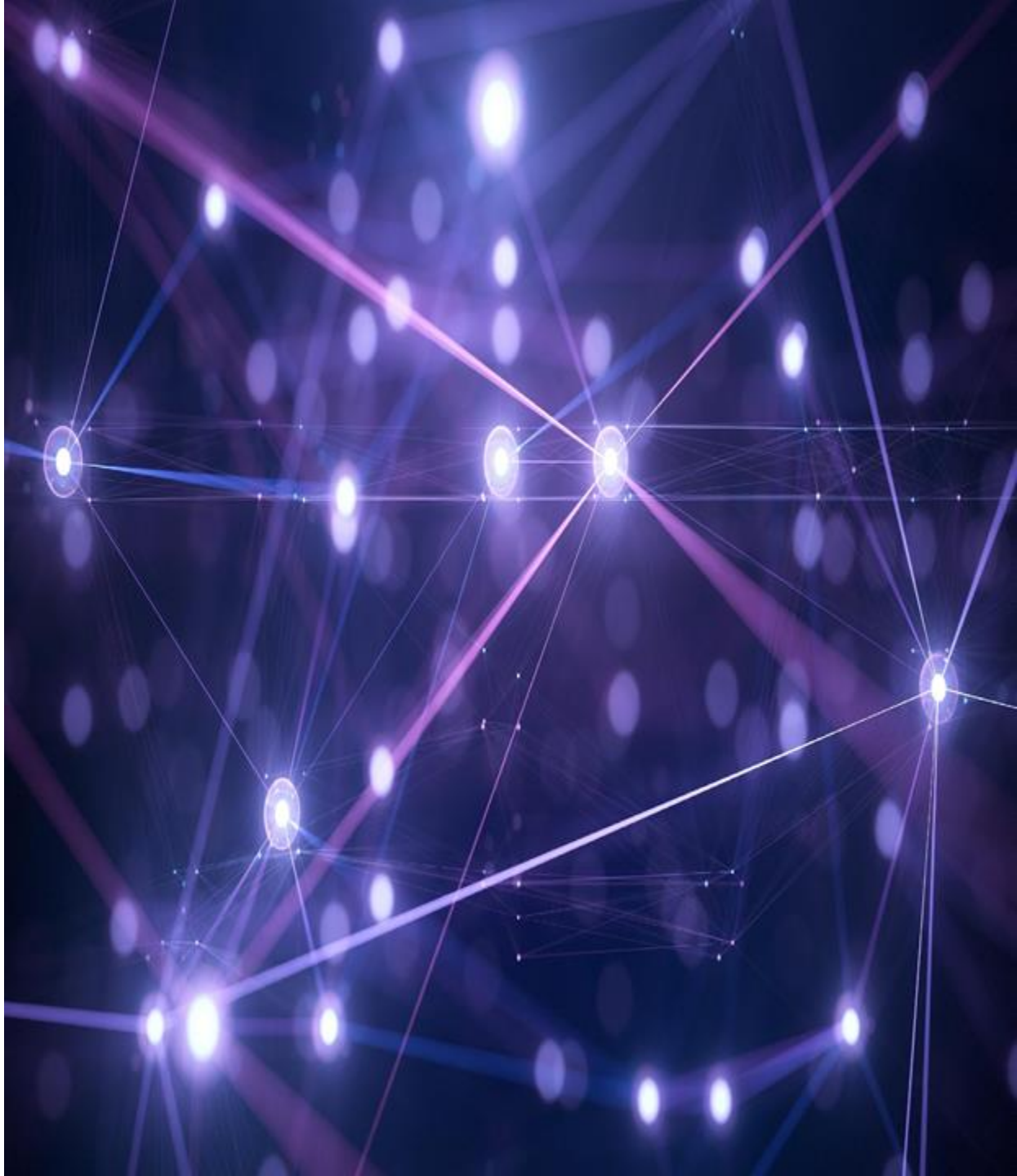
What is Naïve Bayes?

- Uses a csv training file of previously coded survey comments to predict how a new statement should be coded.
- Requires minimal knowledge of programming.
- Works best with longitudinal surveys where the same question is asked each time.
- Training file can be used across multiple questions that tend to elicit similar topic categories.
- Can be used for sentiment analysis to determine if comments are positive, negative or neutral.
- Accuracy of comment category prediction should increase as number of comments in training file increases.
- It helps classify more comments faster with human defined categories.

"Naive Bayes is the most straightforward and fast classification algorithm, which is suitable for a large chunk of data. Naive Bayes classifier is successfully used in various applications such as spam filtering, text classification, sentiment analysis, and recommender systems. It uses Bayes theorem of probability for prediction of unknown class." - DataCamp.com

Naïve Bayes Limitations

- Does not work with new questions that lack coded comments from prior surveys.
- It requires a minimum of 3,000 previously coded comments, and categories with at least 100-200 examples.
- While the analysis of new comments is fast, creating and validating the training file can be time consuming.
- It is still necessary to review the output for accuracy. I have found it to have an accuracy of 60-70%.
- It has a bias towards larger comment categories and works best with more conceptually discrete topics.
- It does not split up longer comments that touch on multiple topics.
- Accuracy of categorization can vary depending on the size training file, the size of the category examples, and the accuracy of the coded comments.



Categorization Method Comparison

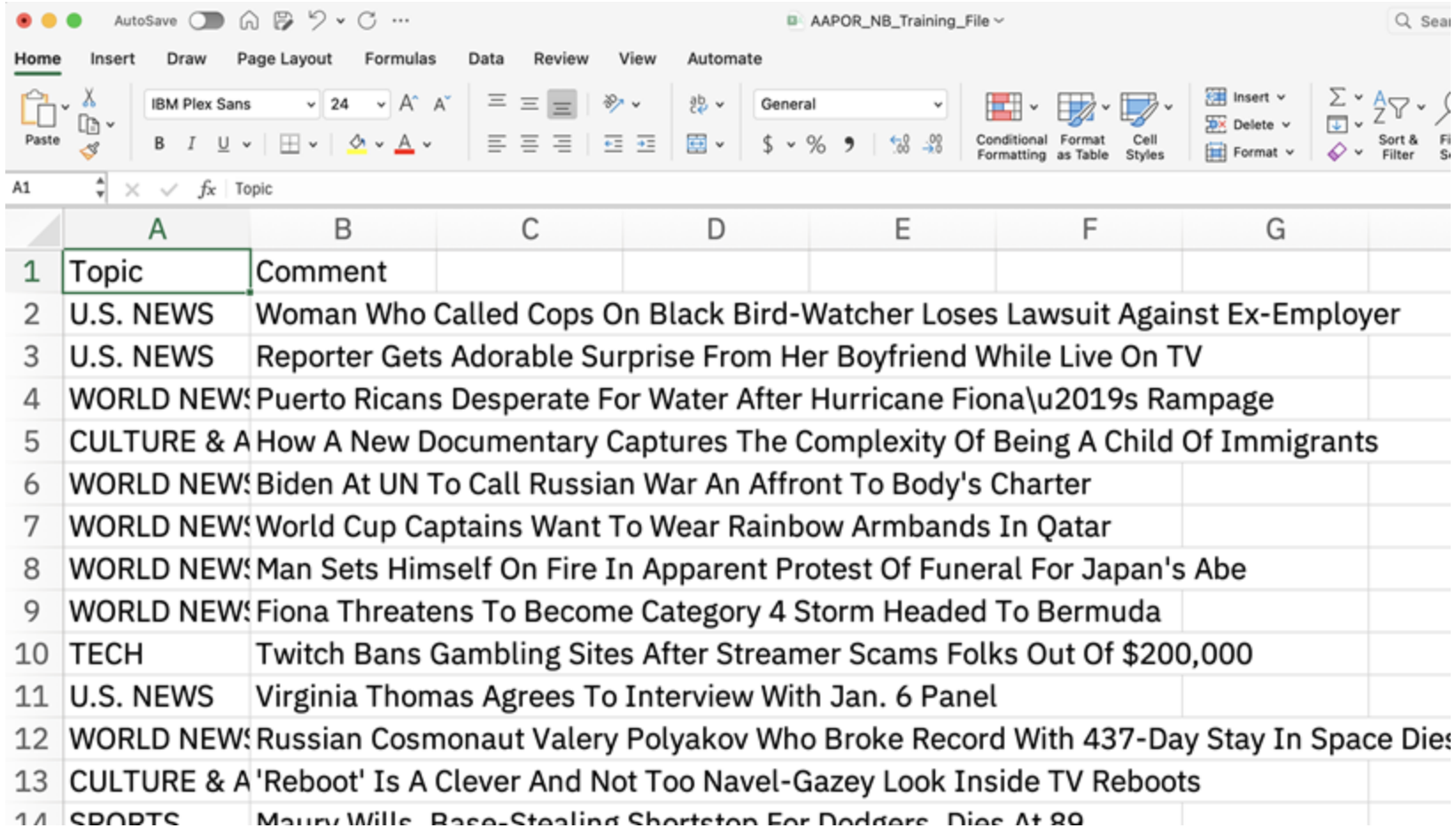
Topics	**Manual Coding** (Total = 1547)	**Comment Sampling** (K-α =.591)	**KPA** (K-α =.436)	**Regex** (K-α =.556)	**Naïve Bayes** (K-α = .585)
Laptops & Accessories	329	24 **70%** match	95 **28%** match	409 **81%** match	249 **85%** match
Network / CICD	346	17 **45%** match	140 **33%** match	216 **36%** match	168 **68%** match
Satisfied	209	19 **73%** match	210 **58%** match	191 **17%** match	118 **43%** match
Tools	407	40 **80%** match	140 **56%** match	480 **42%** match	357 **82%** match
IT Support	182	19 **54%** match	101 **31%** match	95 **18%** match	117 **59%** match
Documentation	18			22 **54%** match	1 **6%** match
Not categorized		1420	235	472	0

“Alpha [0.67 - 0.79]: This range is often considered the lower bound for tentative conclusions. A Krippendorff's Alpha in this range suggests moderate agreement; thus, outcomes should be interpreted with concern, questioning the roots of such diverging ratings.” - www.k-alpha.org

Creating the Training File

Preparing the training file

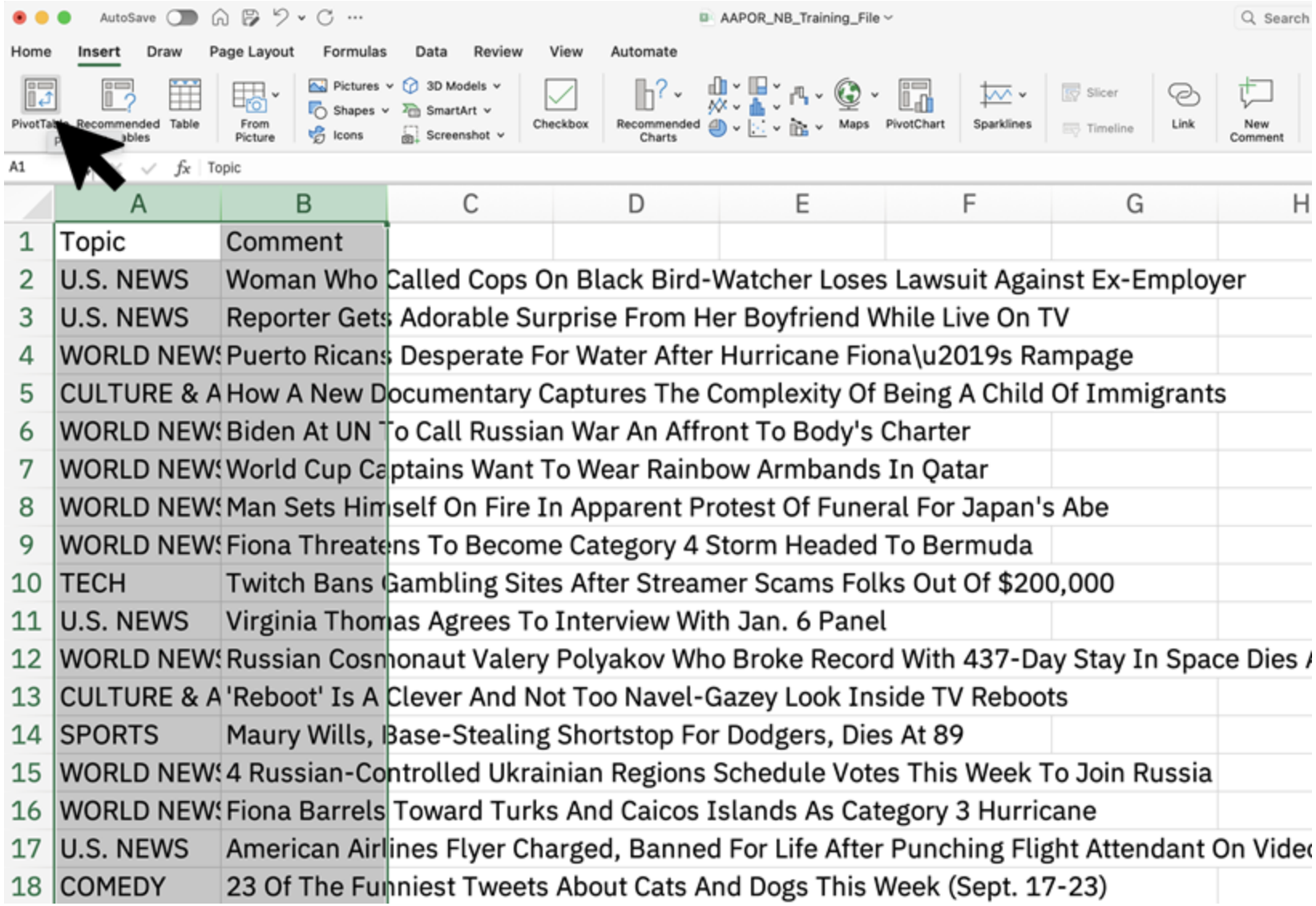
- Open Excel
- Paste in comments from previous survey.
- Label first column Topic and second column Comment.
- You can use other column labels if you prefer (e.g., Category, Text), just remember to use those in the python script.



	A	B
1	Topic	Comment
2	U.S. NEWS	Woman Who Called Cops On Black Bird-Watcher Loses Lawsuit Against Ex-Employer
3	U.S. NEWS	Reporter Gets Adorable Surprise From Her Boyfriend While Live On TV
4	WORLD NEWS	Puerto Ricans Desperate For Water After Hurricane Fiona\u2019s Rampage
5	CULTURE & A	How A New Documentary Captures The Complexity Of Being A Child Of Immigrants
6	WORLD NEWS	Biden At UN To Call Russian War An Affront To Body's Charter
7	WORLD NEWS	World Cup Captains Want To Wear Rainbow Armbands In Qatar
8	WORLD NEWS	Man Sets Himself On Fire In Apparent Protest Of Funeral For Japan's Abe
9	WORLD NEWS	Fiona Threatens To Become Category 4 Storm Headed To Bermuda
10	TECH	Twitch Bans Gambling Sites After Streamer Scams Folks Out Of $200,000
11	U.S. NEWS	Virginia Thomas Agrees To Interview With Jan. 6 Panel
12	WORLD NEWS	Russian Cosmonaut Valery Polyakov Who Broke Record With 437-Day Stay In Space Dies
13	CULTURE & A	'Reboot' Is A Clever And Not Too Navel-Gazey Look Inside TV Reboots
14	SPORTS	Maury Wills, Base-Stealing Shortstop For Dodgers, Dies At 89

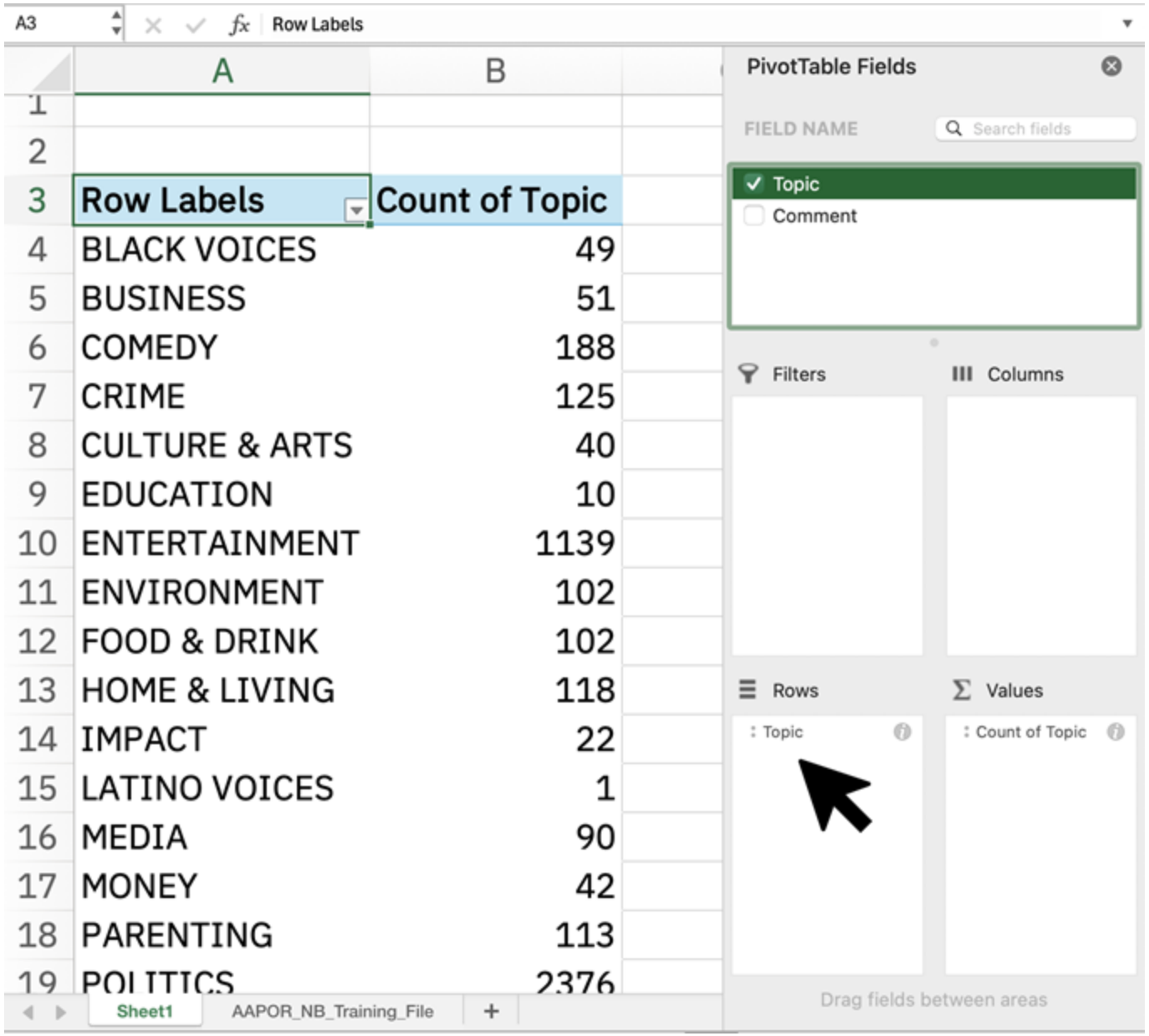
Creating a Pivot Table

- Select the two columns and click Pivot Table.



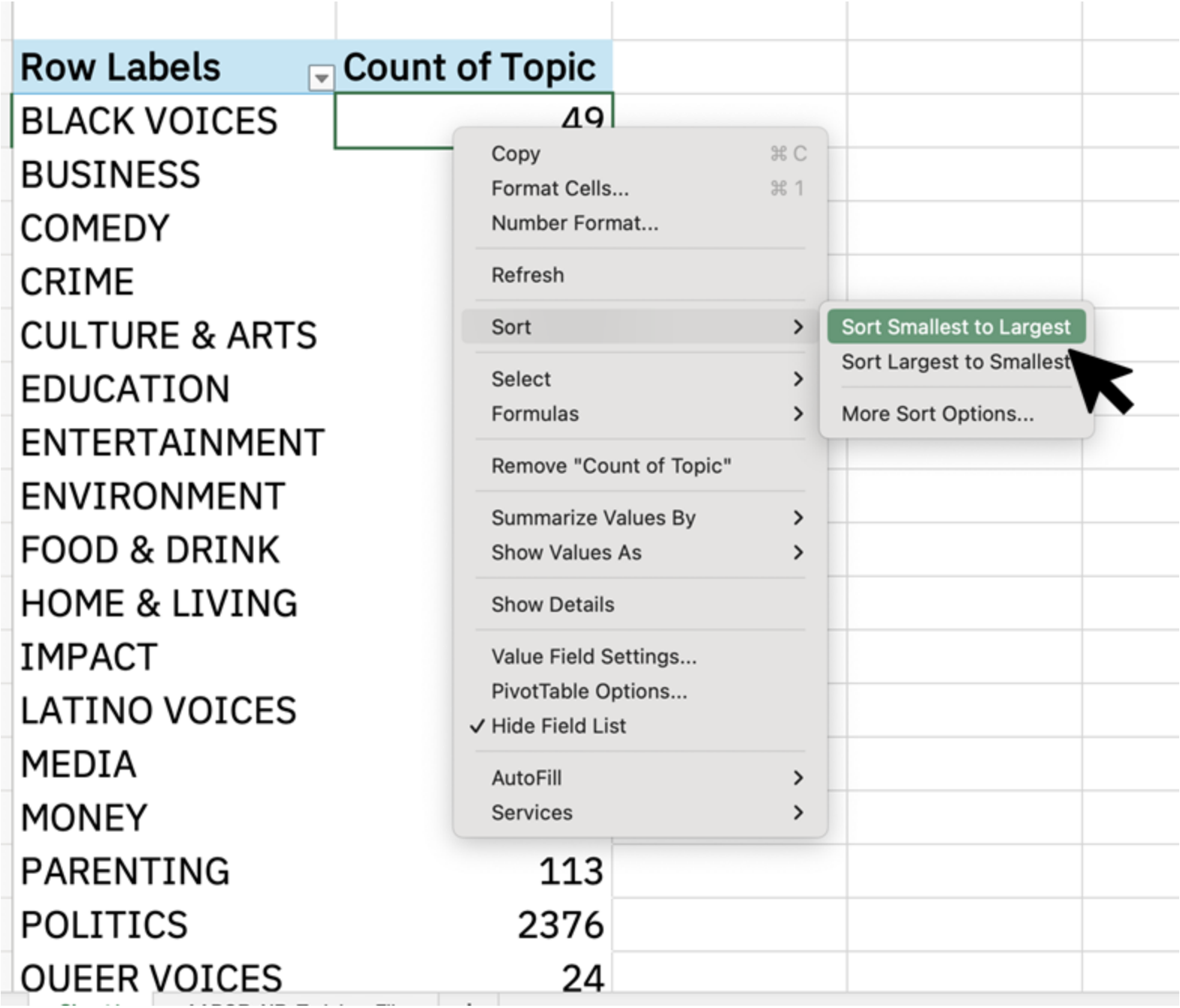
Using the Pivot table

- Drag the Topic column into the Rows and Values area.
- Look at the number of comments in each topic.



Sorting the category count

- Click on a cell that contains a topic count.
- Select “Sort” and “Sort Smallest to Largest.”



Combining Categories

- Look at the number of comments in each topic.
- Look for smaller categories that can be combined into larger categories.
- Look for large categories that you might want to split.
- Category counts should reflect typical distribution of topics, but small categories are less likely to be predicted.

Row Labels	Count of Topic
(blank)	
WEDDINGS	1
LATINO VOICES	1
EDUCATION	10
TRAVEL	13
RELIGION	14
TECH	19
IMPACT	22
QUEER VOICES	24
SCIENCE	24
CULTURE & ARTS	40
MONEY	42
BLACK VOICES	49
BUSINESS	51
WOMEN	62
MEDIA	90
WEIRD NEWS	95
ENVIRONMENT	102
FOOD & DRINK	102
PARENTING	113
WELLNESS	113
HOME & LIVING	118
CRIME	125
STYLE & BEAUTY	144
SPORTS	173
COMEDY	188
WORLD NEWS	839
ENTERTAINMENT	1139
U.S. NEWS	1142
POLITICS	2376
Grand Total	**7231**

Check to see if there are blank rows that should be deleted.

Education / Parenting

Tech / Science

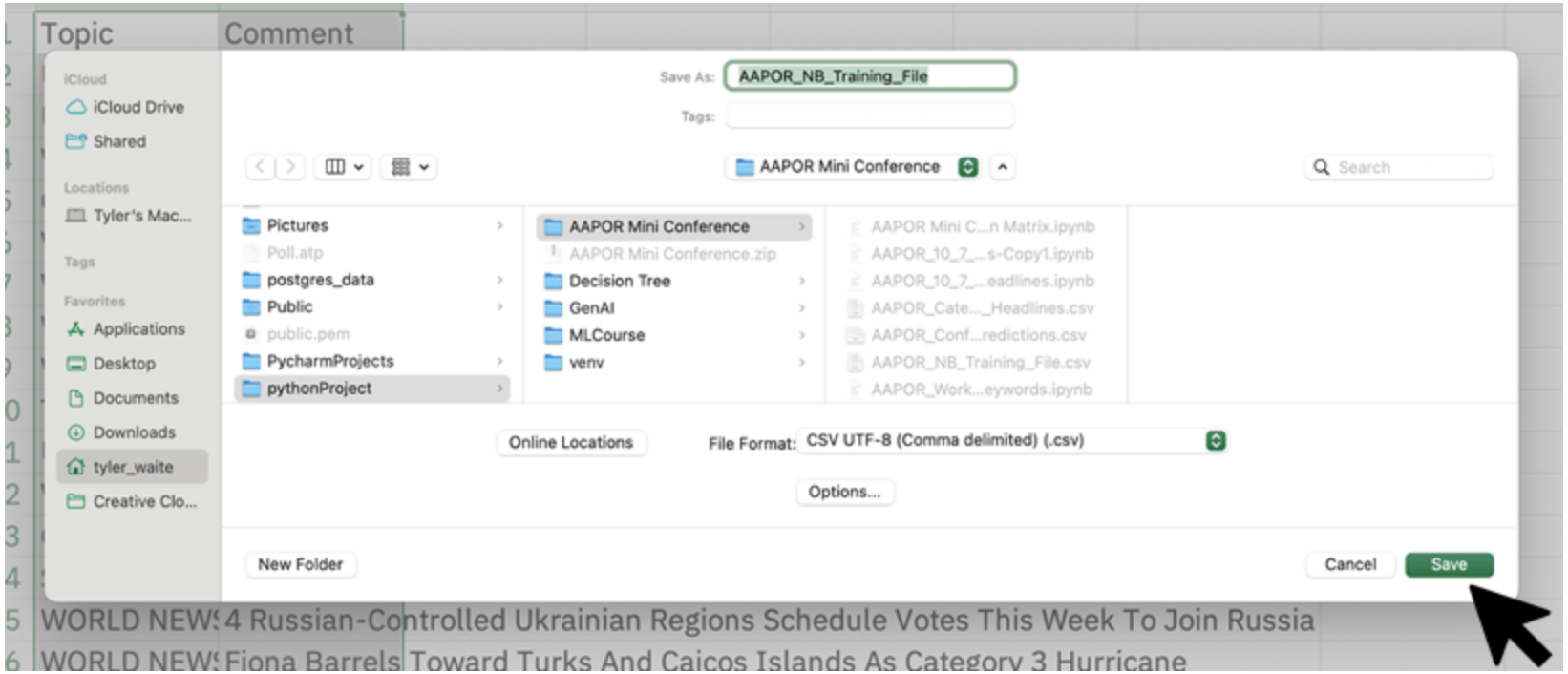
Money / Business

Media / Entertainment

World Politics
US Politics
State Politics

Save the training file

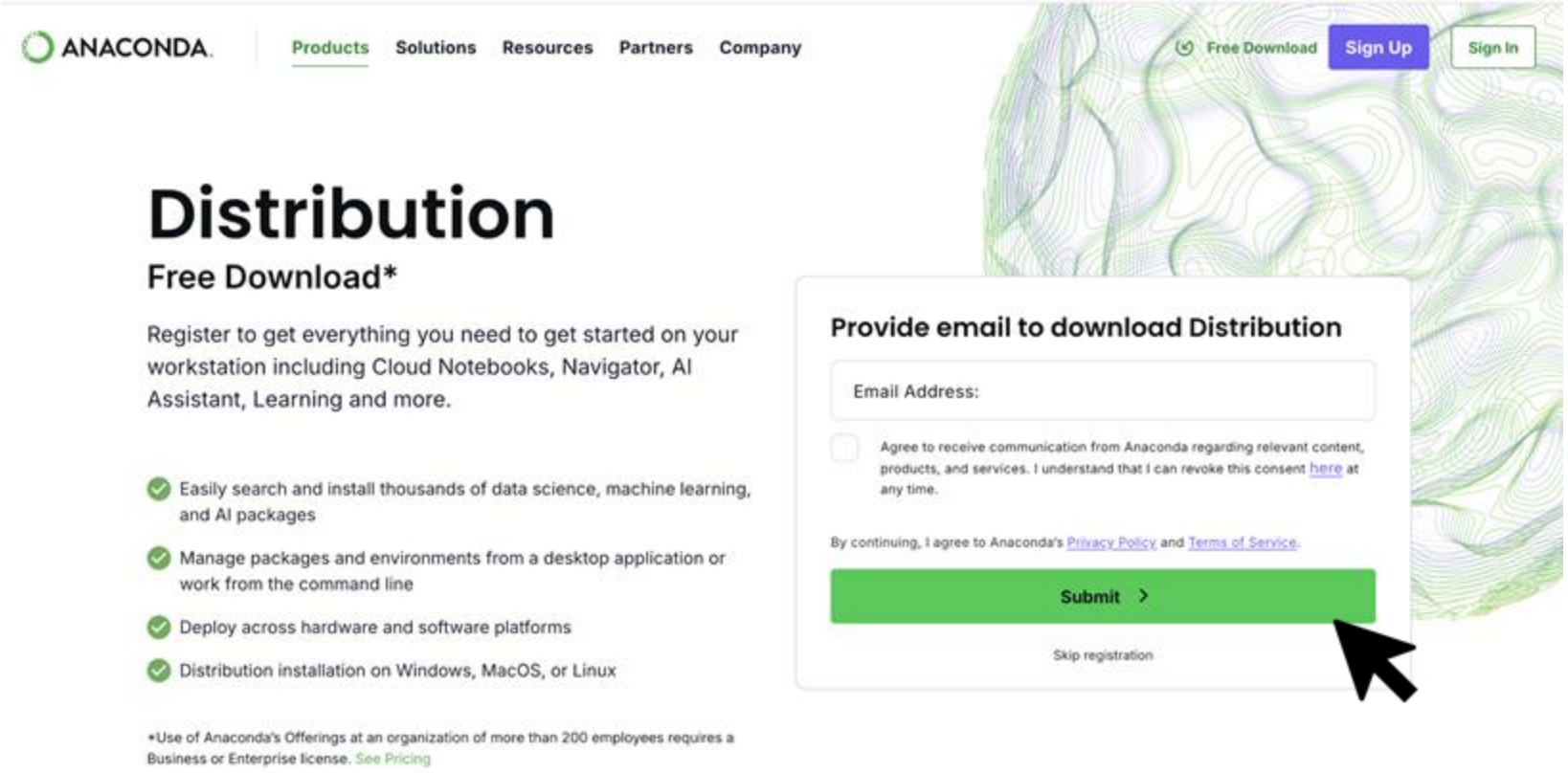
- Once you are happy with your categories delete the sheet with the Pivot Table.
- Be sure to save the file as csv.



Anaconda, NLTK, Scikit-Learn & Jupyter Notebooks

Installing Anaconda

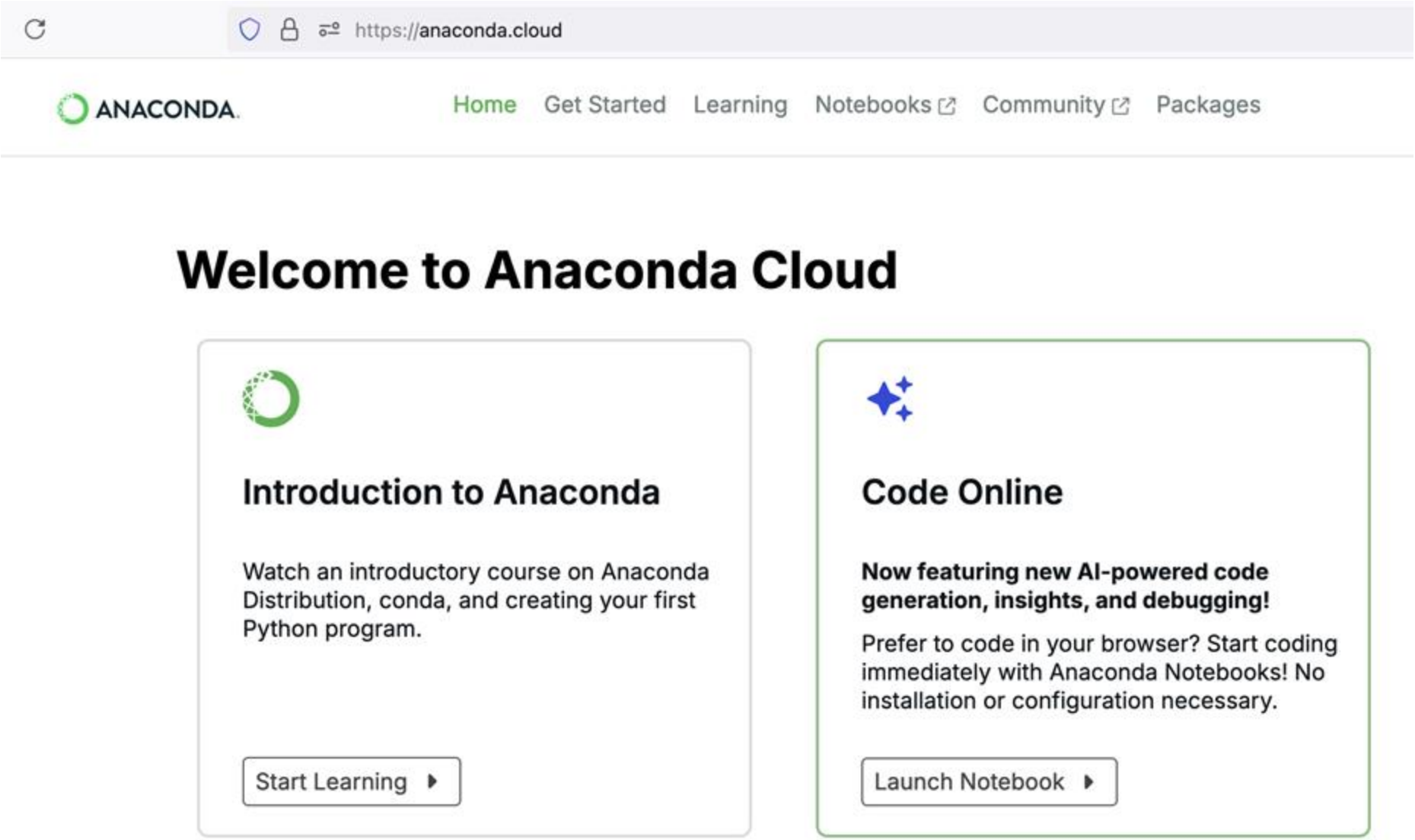
- Anaconda is a free open-source online application that you can use to launch a Jupyter notebook.



https://www.anaconda.com/download

Using Anaconda online

- Anaconda now has an online version of jupyter notebooks you can use if you don't want to install any libraries on your computer.
- After creating an account with Anaconda sign in and select the "Code Online" option.



Installing NLTK

- If you want to launch Jupyter Notebooks from the Anaconda application, you will need to install some libraries.
- The NLTK is a natural language processing library that we will use to help categorize the comments.

Mac/Unix

1. Install NLTK: run `pip install --user -U nltk`
2. Install Numpy (optional): run `pip install --user -U numpy`
3. Test installation: run `python` then type `import nltk`

For older versions of Python it might be necessary to install setuptools (see **https://pypi.python.org/pypi/setuptools**) and to install pip (`sudo easy_install pip`).

Windows

These instructions assume that you do not already have Python installed on your machine.

32-bit binary installation

1. Install Python 3.8: **https://www.python.org/downloads/** (avoid the 64-bit versions)
2. Install Numpy (optional): **https://numpy.org/install/**
3. Install NLTK: **https://pypi.python.org/pypi/nltk**
4. Test installation: `Start>Python38`, then type `import nltk`

https://www.nltk.org/install.html

Installing SciKit-Learn libraries

- Python is a powerful language because there are so many libraries that can be used to perform various functions on your data.

Installing the latest release

pip | **conda**

Install conda using the Anaconda or miniconda installers or the miniforge installers (no administrator permission required for any of those). Then run:

```
$ conda create -n sklearn-env -c conda-forge scikit-learn
$ conda activate sklearn-env
```

In order to check your installation, you can use:

```
$ conda list scikit-learn  # show scikit-learn version and location
$ conda list               # show all installed packages in the environment
$ python -c "import sklearn; sklearn.show_versions()"
```

https://scikit-learn.org/stable/install.html

Installing Numpy

- Numpy should be installed with Anaconda, but if for some reason it isn't try these commands.

CONDA

If you use `conda`, you can install NumPy from the `defaults` or `conda-forge` channels:

```
# Best practice, use an environment rather than install in the base env
conda create -n my-env
conda activate my-env
# If you want to install from conda-forge
conda config --env --add channels conda-forge
# The actual install command
conda install numpy
```

PIP

If you use `pip`, you can install NumPy with:

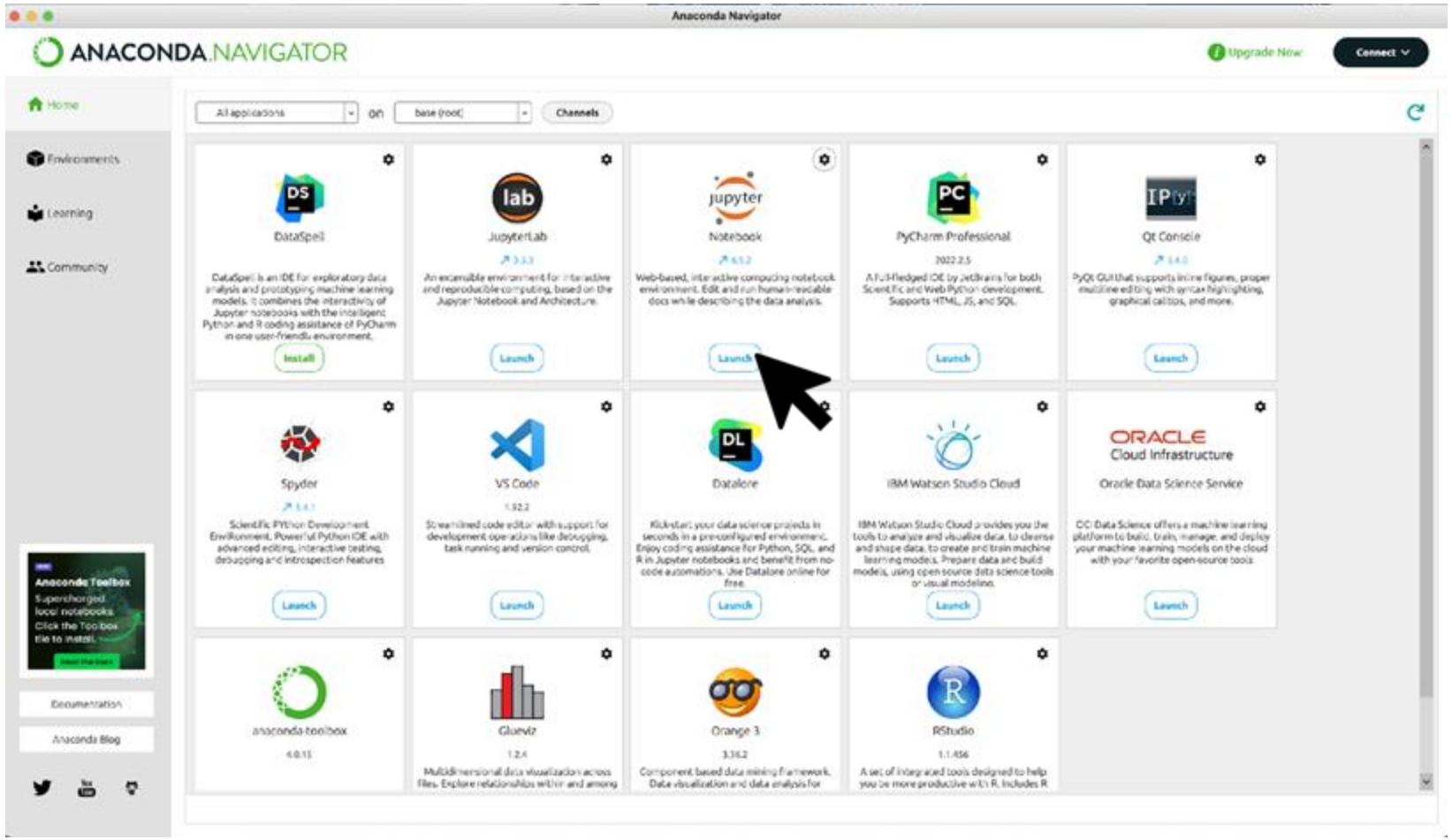
```
pip install numpy
```

https://numpy.org/install/

Jupyter Notebooks

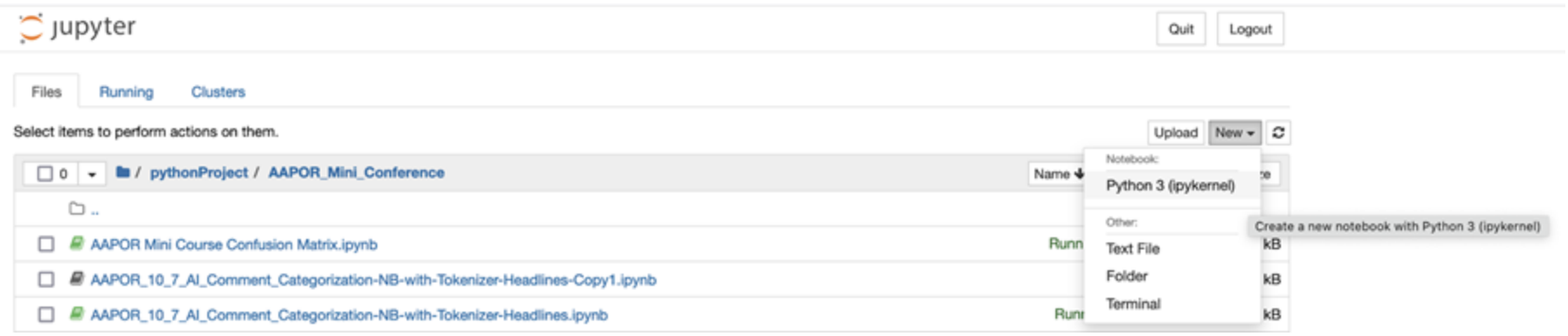
Launching Jupyter Notebook

- Once Anaconda is installed you should be able to open it and launch a Jupyter notebook.
- Jupyter Notebooks are where we will be creating the Python script.



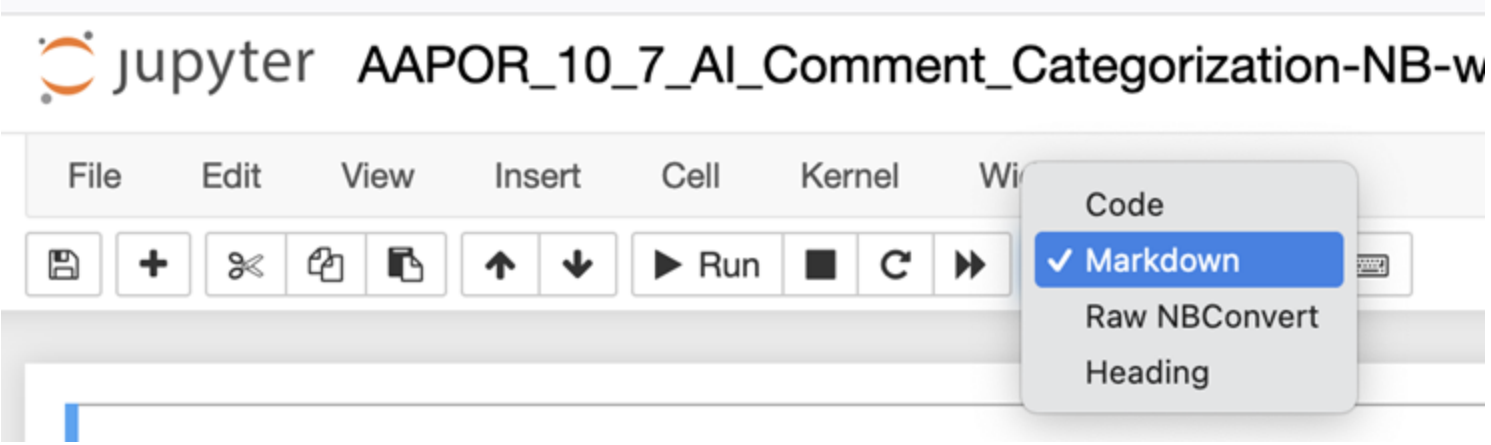
Creating a new Python notebook

- Navigate to the folder where you want to keep your Python scripts.
- Select the New drop-down menu and Python 3 (ipykernel).



Exploring the Jupyter Notebook Interface

- The Jupyter Notebook interface is very similar in many ways to a text editor.
- You can cut, copy and paste.
- You can search and replace text.
- You can reorder sections of the script.
- You can add sections for explanatory text (markdown) or code.
- When you want to run the code in a section you press Run.



Markdown text

- With Jupyter Notebooks you can add explanatory text outside of the code sections and format it using Markdown.

```
# This is a level 1 heading

## This is a level 2 heading

This is some plain text that forms a paragraph. Add emphasis via **bold** or __bold__, and *italic* or _itali

Paragraphs must be separated by an empty line.

* Sometimes we want to include lists.
* Which can be bulleted using asterisks.

1. Lists can also be numbered.
2. If we want an ordered list.

[It is possible to include hyperlinks](https://www.dataquest.io)

Inline code uses single backticks: `foo()`, and code blocks use triple backticks:
```
bar()
```
Or can be indented by 4 spaces:
```
    foo()
```


And finally, adding images is easy: ![Alt text](https://www.dataquest.io/wp-content/uploads/2023/02/DQ-Logo.s
```

https://www.dataquest.io/blog/jupyter-notebook-tutorial/

Markdown text

- Press play to view your formatted text.

Here's how that Markdown would look once you run the cell to render it:

This is a level 1 heading

This is a level 2 heading

This is some plain text that forms a paragraph. Add emphasis via **bold** or **bold**, and *italic* or *italic*.

Paragraphs must be separated by an empty line.

- Sometimes we want to include lists.
- Which can be bulleted using asterisks.

1. Lists can also be numbered.
2. If we want an ordered list.

It is possible to include hyperlinks

Inline code uses single backticks: `foo()`, and code blocks use triple backticks:

```
bar()
```

Or can be indented by 4 spaces:

```
    foo()
```

And finally, adding images is easy:

DATAQUEST

https://www.dataquest.io/blog/jupyter-notebook-tutorial/

Writing code

- Use a # If you want to add a comment or prevent a line of code from being run.
- All code in a section will be run at the same time.
- Breaking up code into smaller chunks can help with bug checks and validating that data was loaded properly.
- Python remembers the most recent value of the variables in each section that has been run.
- The number in brackets to the left of the code section tells you the order in which the sections have been run.

```
In [1]: print('Hello World!')

        Hello World!

In [2]: import time
        time.sleep(3)

In [3]: def say_hello(recipient):
            return 'Hello, {}!'.format(recipient)
        say_hello('Tim')

Out[3]: 'Hello, Tim!'

In [4]: import numpy as np
        def square(x):
            return x * x

In [5]: x = np.random.randint(1, 10)
        y = square(x)
        print('%d squared is %d' % (x, y))

        7 squared is 49
```

https://www.dataquest.io/blog/jupyter-notebook-tutorial/

Creating the script

Loading the libraries

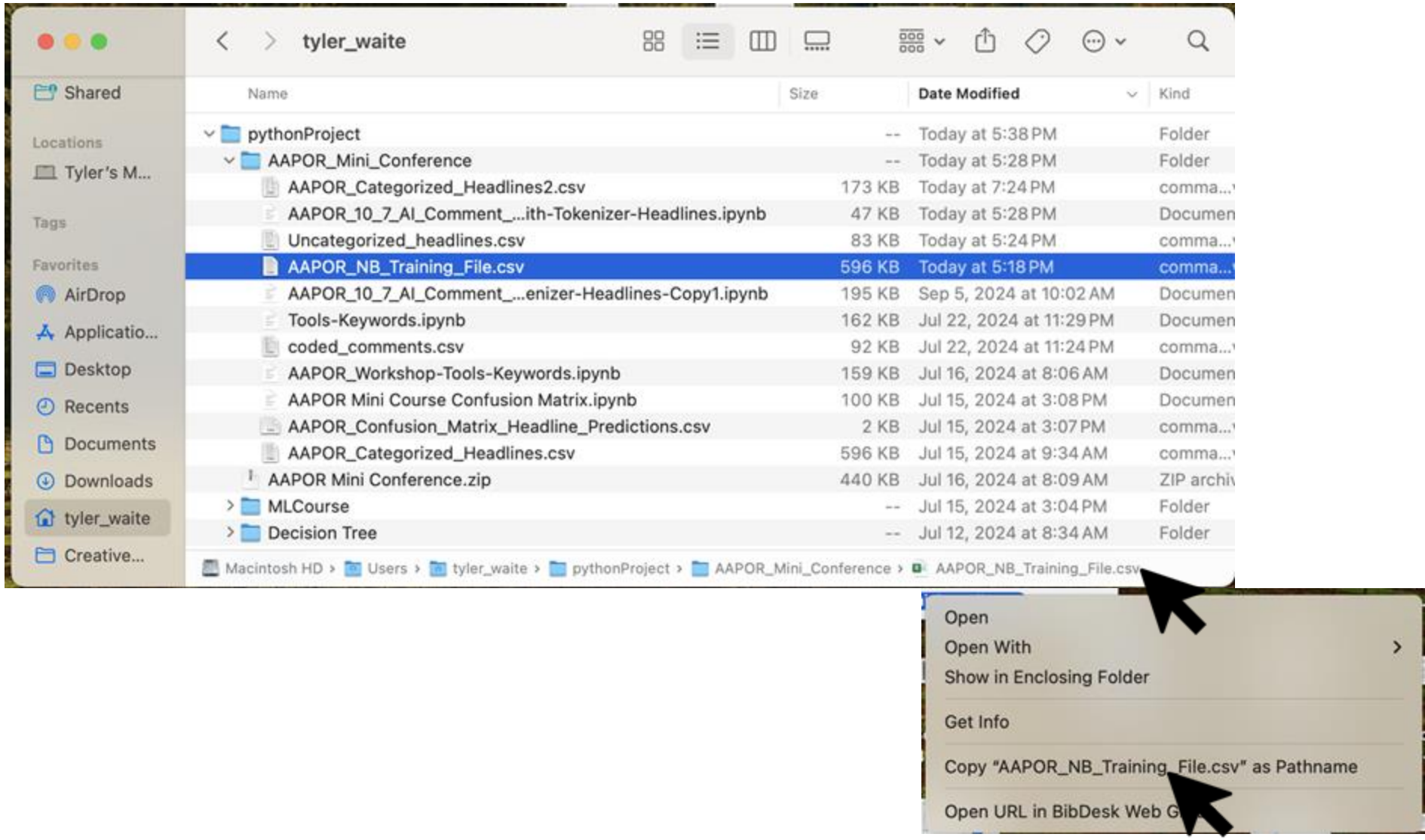
- Don’t change these libraries.
- Don’t worry about the pink section after you click run.

```python
# Python libraries used to run the scripts.  Do not change.
import os
import io
import numpy
import pandas as pd
import re
import os.path
from pandas import DataFrame
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
import nltk
nltk.download('all')
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
```

```
[nltk_data] Downloading collection 'all'
[nltk_data]    |
[nltk_data]    | Downloading package abc to
[nltk_data]    |     /Users/tyler_waite/nltk_data...
[nltk_data]    |   Package abc is already up-to-date!
[nltk_data]    | Downloading package alpino to
[nltk_data]    |     /Users/tyler_waite/nltk_data...
[nltk_data]    |   Package alpino is already up-to-date!
[nltk_data]    | Downloading package averaged_perceptron_tagger to
[nltk_data]    |     /Users/tyler_waite/nltk_data...
[nltk_data]    |   Package averaged_perceptron_tagger is already up-
[nltk_data]    |       to-date!
[nltk_data]    | Downloading package averaged_perceptron_tagger_eng to
[nltk_data]    |     /Users/tyler_waite/nltk_data...
[nltk_data]    |   Package averaged_perceptron_tagger_eng is already
[nltk_data]    |       up-to-date!
[nltk_data]    | Downloading package averaged_perceptron_tagger_ru to
[nltk_data]    |     /Users/tyler_waite/nltk_data...
[nltk_data]    |   Package averaged_perceptron_tagger_ru is already
[nltk_data]    |       up-to-date!
```

Copying file pathname

- An easy way to copy a files path name is to right click or control click on the file name at the bottom of the finder window, then click Copy.



Loading the training file

- Replace the file name with the path to your training file.
- If you want to see more rows, put the number of rows you want between the parenthesis: head(15).

Loads file containing categorized comments from previous surveys.
Make sure file is in the same folder as your python script or use full pathname for file.

```
comments_df = pd.read_csv(r"AAPOR_NB_Training_File.csv")
```

Use this to view the dataframe so you can make sure the comments were correctly loaded.

```
comments_df.head()
```

	Topic	Comment
0	U.S. NEWS	Woman Who Called Cops On Black Bird-Watcher Lo...
1	U.S. NEWS	Reporter Gets Adorable Surprise From Her Boyfr...
2	WORLD NEWS	Puerto Ricans Desperate For Water After Hurric...
3	CULTURE & ARTS	How A New Documentary Captures The Complexity ...
4	WORLD NEWS	Biden At UN To Call Russian War An Affront To ...

Vectorizing the training file

- This is the Naïve Bayes process that creates the prediction values for the categories from the training file.

This creates the comment classifier.

```
vectorizer = CountVectorizer()
```

When using this with your own files, update 'Comment' to match the column name for your comments.

```
counts = vectorizer.fit_transform(comments_df['Comment'].values)
classifier = MultinomialNB()
```

When using this with your own files, update 'Topic' to match the column name for your categories or topics.

```
targets = comments_df['Topic'].values
classifier.fit(counts, targets)
```

```
▾ MultinomialNB
MultinomialNB()
```

Loading the uncategorized comments

- If you have demographic data associated with the comments, it is OK to include those columns in the uncategorized file. The Naïve Bayes category column is appended to the existing columns in the uploaded file.

This loads the file containing the new uncategorized comments. Make sure file is in the same folder as your python script or use full pathname for file.

```
new_comments_df = pd.read_csv(r"Uncategorized_headlines.csv")
```

View the dataframe to make sure the comments were correctly loaded.

```
new_comments_df.head()
```

	Comment
0	brown labels democrats new slogan awful
1	victorians honoured with queens birthday awards
2	charges to be laid over barcaldine train crash
3	jim morrison surfaces in pre doors short film
4	scientists converge on nt for eclipse

This removes all rows without a comment

```
new_comments_df['Comment'].fillna(0, inplace = True)
new_comments_df = new_comments_df[new_comments_df.Comment != 0]
```

Lemmatization

- This takes the comments and removes words like "of" "or" "and" "the" which are very common and not useful for categorization. It also converts all capital letters to lowercase and removes endings like "ing" "ed" to convert words to their root form which will be easier to match with categorized text.
- This helps improve matching new comments to similar previously categorized comments.

```
text = list(new_comments_df['Comment'])

lemmatizer = WordNetLemmatizer()

corpus = []

for i in range(len(text)):

    r = re.sub('[^a-zA-Z]', ' ', text[i])
    r = r.lower()
    r = r.split()
    r = [word for word in r if word not in stopwords.words('english')]
    r = [lemmatizer.lemmatize(word) for word in r]
    r = ' '.join(r)
    corpus.append(r)
```

This puts the lemmatized comment into a new column so that you still have the original, easier to read, version.

```
new_comments_df['lemm_text'] = corpus
new_comments_df.head()
```

	Comment	lemm_text
0	brown labels democrats new slogan awful	brown label democrat new slogan awful
1	victorians honoured with queens birthday awards	victorian honoured queen birthday award
2	charges to be laid over barcaldine train crash	charge laid barcaldine train crash
3	jim morrison surfaces in pre doors short film	jim morrison surface pre door short film
4	scientists converge on nt for eclipse	scientist converge nt eclipse

Outputting the categorized comments

- This converts the lemmatized new comments into an array and a predicted category classification is assigned.
- Once the categorization of the new comments is complete it writes the results to a new column that is added to the data frame with the comments.
- The last step outputs the data frame to a csv file.

This writes the dataframe to an array

```
new_comments = new_comments_df['Comment'].to_numpy()
new_comments
```

```
array(['brown labels democrats new slogan awful',
       'victorians honoured with queens birthday awards',
       'charges to be laid over barcaldine train crash', ...,
       'wheatbelt towns propose amalgamation',
       'shoppers boost property trusts bottom line',
       'elton john becomes a father'], dtype=object)
```

Assigns predicted categories to comments

```
new_comments_counts = vectorizer.transform(new_comments)
new_categories = classifier.predict(new_comments_counts)
new_categories
```

```
array(['POLITICS', 'ENTERTAINMENT', 'U.S. NEWS', ..., 'POLITICS',
       'STYLE & BEAUTY', 'ENTERTAINMENT'], dtype='<U14')
```

This adds a column with the new predicted topic category.

```
categories_df = pd.DataFrame(new_categories, columns=['Topics'])
categorized_comments = categories_df.join(new_comments_df)
```

This outputs comment classifications to csv

```
categorized_comments.to_csv(r"AAPOR_Categorized_Headlines.csv", index=False)
```

Viewing the results

Reviewing the categorization

- Notice how Lemmatization changed the text by removing words like “with”, “to”, “be”, “over”, “in” and “on.”
- You can delete this column if you like.

Topics	Comment	lemm_text
POLITICS	brown labels democrats new slogan awful	brown label democrat new slogan awful
ENTERTAINMENT	victorians honoured with queens birthday awards	victorian honoured queen birthday award
U.S. NEWS	charges to be laid over barcaldine train crash	charge laid barcaldine train crash
ENTERTAINMENT	jim morrison surfaces in pre doors short film	jim morrison surface pre door short film
POLITICS	scientists converge on nt for eclipse	scientist converge nt eclipse

Reviewing the categorization

- Notice how the NB was biased towards the larger categories.
- Notice how the top four categories map to the top four predicted categories.
- Notice how only categories with over 100 examples had any predictions.
- This is why it is important to roll up smaller categories in your training data into larger groups and why you may want to split up large categories into smaller sets.

Original Categories

Row Labels	Count of Topic
POLITICS	2376
U.S. NEWS	1142
ENTERTAINMENT	1139
WORLD NEWS	839
COMEDY	188
SPORTS	173
STYLE & BEAUTY	144
CRIME	125
HOME & LIVING	118
WELLNESS	113
PARENTING	113
FOOD & DRINK	102
ENVIRONMENT	102
WEIRD NEWS	95
MEDIA	90
WOMEN	62
BUSINESS	51
BLACK VOICES	49
MONEY	42
CULTURE & ARTS	40
SCIENCE	24
QUEER VOICES	24
IMPACT	22
TECH	19
RELIGION	14
TRAVEL	13
EDUCATION	10
LATINO VOICES	1
WEDDINGS	1
(blank)	
Grand Total	7231

Predicted Categories

Row Labels	Count of Topics
POLITICS	969
U.S. NEWS	517
WORLD NEWS	340
ENTERTAINMENT	160
SPORTS	5
PARENTING	2
STYLE & BEAUTY	2
COMEDY	2
ENVIRONMENT	1
HOME & LIVING	1
(blank)	
Grand Total	1999

Improving the accuracy of the training file

Confusion Matrix

- The Confusion Matrix takes a sample of the training file and then tries to predict what the category should be.
- It then creates a table that show you how many predictions were correct and how many predictions were assigned to other categories.
- This can help you identify categories that might need closer review to make sure the comments are correctly categorized.

```python
#This imports the python libraries that will be used in this script.  Do not change.

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn import metrics
from sklearn import datasets, svm
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.model_selection import train_test_split
```

```python
#Loads the training file containing categorized comments.  Replace the pathname with the pathname to your file.
data = pd.read_csv('AAPOR_Categorized_Headlines.csv', encoding='latin-1')

#Column names from your training file.  Change 'Headline' and 'Category' to match your data file if needed.
data.columns = ['Category', 'Headline']

#This lets you view your imported data.  Change number to view more or fewer rows.
data.head(10)
```

	Category	Headline
0	U.S. NEWS	Woman Who Called Cops On Black Bird-Watcher Lo...
1	U.S. NEWS	Reporter Gets Adorable Surprise From Her Boyfr...
2	WORLD NEWS	Puerto Ricans Desperate For Water After Hurric...
3	CULTURE & ARTS	How A New Documentary Captures The Complexity ...
4	WORLD NEWS	Biden At UN To Call Russian War An Affront To ...
5	WORLD NEWS	World Cup Captains Want To Wear Rainbow Armban...
6	WORLD NEWS	Man Sets Himself On Fire In Apparent Protest O...
7	WORLD NEWS	Fiona Threatens To Become Category 4 Storm Hea...
8	TECH	Twitch Bans Gambling Sites After Streamer Scam...
9	U.S. NEWS	Virginia Thomas Agrees To Interview With Jan. ...

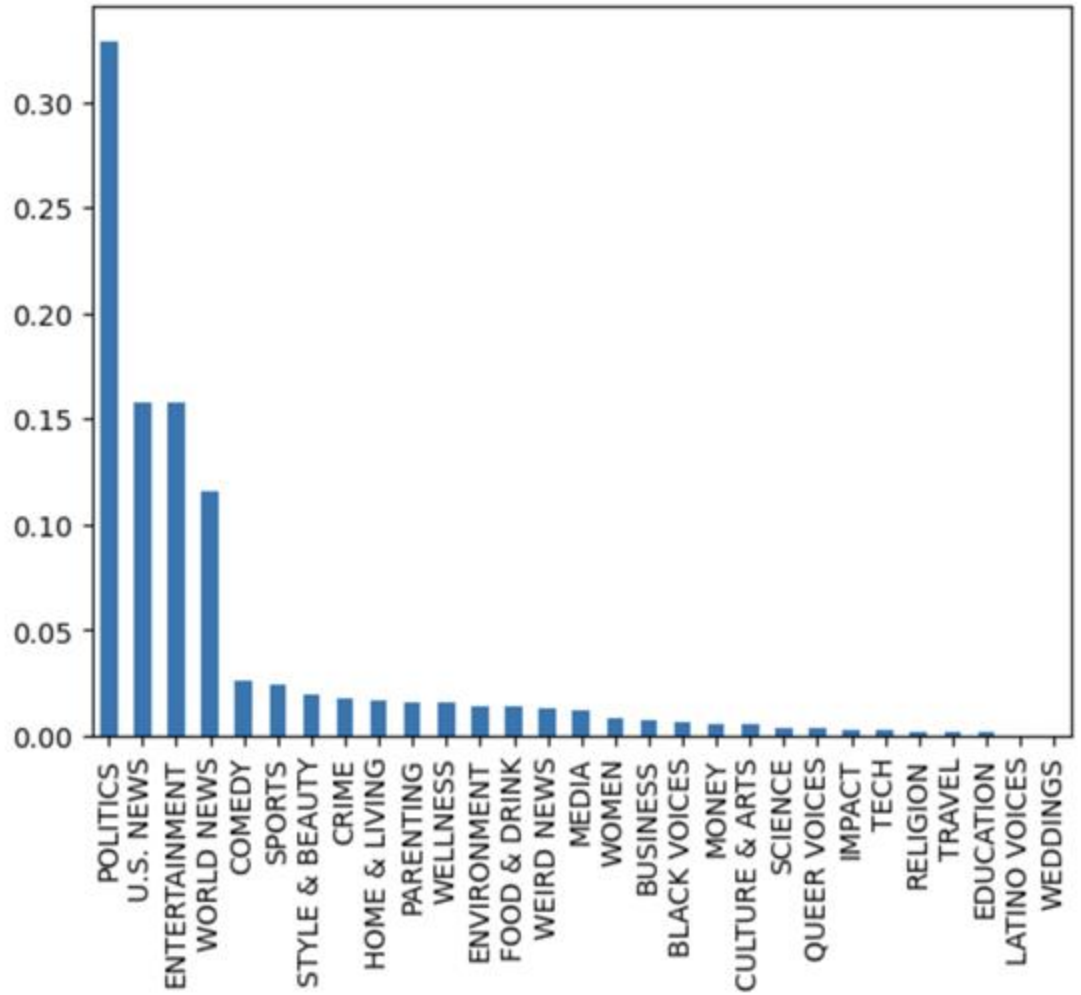
Frequency Chart

- Creating a bar chart of the comment category frequencies shows that there are a few large categories.

```
# This checks the frequency of the categories in the training file.
# Because NB uses probability, predictions will be skewed towards larger categories.
# Change 'Category' to match your column name.

data['Category'].value_counts(normalize = True).plot.bar()
```

```
<Axes: >
```



Predicting the training categories

- This code randomly selects 33% of the training file to put into the test data.
- It then vectorizes the test file to predict the categories using the remaining training data.
- The predicted category is then compared with the original category.

```
# This creates the feature and label sets.  Only change 'Heading'and 'Category' to match your file.

X = data['Headline']
y = data['Category']

# train test split (66% train - 33% test)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=123)

print('Training Data :', X_train.shape)
print('Testing Data : ', X_test.shape)
```

```
Training Data : (4844,)
Testing Data :  (2387,)
```

```
# Train Bag of Words model.  Do not change.

from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()

X_train_cv = cv.fit_transform(X_train)
X_train_cv.shape
```

```
(4844, 10272)
```

```
# Training Logistic Regression model. Do not change this block.

from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()
lr.fit(X_train_cv, y_train)

# transform X_test using CV
X_test_cv = cv.transform(X_test)

# generate predictions
predictions = lr.predict(X_test_cv)

#predictions
predictions_df = pd.DataFrame(data = predictions)
predictions_df.head(20)
```

Labeling the rows and columns

- When outputting the matrix you need to name all the columns.
- The column names should be in alphabetical order even if that is not the order of the columns in the original file.
- The index (rows) names should match the order of the column names.

```
# This creates the confusion matrix table.
# index and columns text should match the categories in your training file.
# The same names should be used in both the index and columns section.
# The category names should be in alphabetical order.
# The last category should not have a ,

df = pd.DataFrame(metrics.confusion_matrix(y_test,predictions),index=[
    'BLACK VOICES','BUSINESS','COMEDY','CRIME','CULTURE & ARTS','EDUCATION','ENTERTAINMENT','ENVIRONMENT',
    'FOOD & DRINK','HOME & LIVING','IMPACT','MEDIA','MONEY','PARENTING','POLITICS','QUEER VOICES','RELIGION',
    'SCIENCE','SPORTS','STYLE & BEAUTY','TECH','TRAVEL','U.S. NEWS','WEIRD NEWS','WELLNESS','WOMEN',
    'WORLD NEWS'
], columns=[
    'BLACK VOICES','BUSINESS','COMEDY','CRIME','CULTURE & ARTS','EDUCATION','ENTERTAINMENT','ENVIRONMENT',
    'FOOD & DRINK','HOME & LIVING','IMPACT','MEDIA','MONEY','PARENTING','POLITICS','QUEER VOICES','RELIGION',
    'SCIENCE','SPORTS','STYLE & BEAUTY','TECH','TRAVEL','U.S. NEWS','WEIRD NEWS','WELLNESS','WOMEN',
    'WORLD NEWS'
])

df
```

Interpreting the Confusion Matrix

- The rows are the original categories. The columns are the predicted categories.
- The higher the number at the intersection of the category column and row pairs the more accurate the prediction.
- If the count is low look for high counts in other categories (e.g., Comedy).

	BLACK VOICES	BUSINESS	COMEDY	CRIME	CULTURE & ARTS	EDUCATION	ENTERTAINMENT	ENVIRONMENT	FOOD & DRINK	HOME & LIVING	...	SCIENCE	SPORTS
BLACK VOICES	0	0	0	0	0	0	7	0	0	0	...	0	1
BUSINESS	0	1	0	0	0	0	1	0	0	0	...	0	0
COMEDY	0	0	29	0	0	0	20	1	0	0	...	0	0
CRIME	0	0	0	5	0	0	0	0	0	0	...	0	0
CULTURE & ARTS	0	0	0	0	0	0	9	0	0	0	...	0	0
EDUCATION	0	0	0	0	0	0	0	0	0	0	...	0	0
ENTERTAINMENT	1	0	9	0	0	0	248	0	1	1	...	0	4
ENVIRONMENT	0	0	0	0	0	0	3	0	0	0	...	0	0
FOOD & DRINK	0	0	0	0	0	0	5	0	10	0	...	0	0
HOME & LIVING	0	0	0	0	0	0	5	0	1	27	...	0	0
IMPACT	0	0	0	0	0	0	0	0	1	0	...	0	0
MEDIA	0	1	1	0	0	0	3	0	0	0	...	0	0
MONEY	0	0	0	0	0	0	2	0	2	0	...	0	0
PARENTING	0	0	0	0	0	0	8	0	1	1	...	0	0

Outputting the Confusion Matrix

- This script will output the matrix to a csv.
- When we look at the four largest categories in Excel and highlight the cells with the largest counts, we see that the majority of headlines in those categories were correctly predicted.
- If we then take the count at the intersection of the categories and divide it by the total for the column we can see the overall accuracy.
- The “U.S. News” category in the training file should probably be reviewed to see if the categorization was accurate.

```
# This saves the confusion matrix to a csv file.
# Change the pathname to where you want to save the file and the file name.
# Divide the number where the row and column names match by the total count for the column to determine accuracy.

matrix = df
matrix.to_csv(r"AAPOR_Confusion_Matrix_Headline_Predictions.csv", index=True)
```

	ENTERTAINMENT	POLITICS	U.S. NEWS	WORLD NEWS
BLACK VOICES	7	3	5	0
BUSINESS	1	2	5	0
COMEDY	20	9	0	0
CRIME	0	7	25	4
CULTURE & ARTS	9	3	5	0
EDUCATION	0	2	4	0
ENTERTAINMENT	248	45	37	10
ENVIRONMENT	3	8	23	6
FOOD & DRINK	5	8	3	0
HOME & LIVING	5	2	2	1
IMPACT	0	2	0	0
MEDIA	3	8	3	0
MONEY	2	4	3	1
PARENTING	8	5	1	2
POLITICS	21	670	69	22
QUEER VOICES	2	1	2	0
RELIGION	0	0	1	1
SCIENCE	3	0	2	0
SPORTS	16	6	12	8
STYLE & BEAUTY	12	7	5	0
TECH	0	3	4	1
TRAVEL	1	0	0	0
U.S. NEWS	31	91	221	27
WEIRD NEWS	8	5	15	0
WELLNESS	1	4	1	2
WOMEN	3	4	4	0
WORLD NEWS	16	40	55	178
Total	425	939	507	263
Accuracy	58%	71%	44%	68%

Contact

Tyler Waite
Advisory Data Scientist
CIO, AI, Automation & Data Platform
Tyler.Waite@ibm.com

More reading on Python, Naïve Bayes and comment coding

2022, November). Understanding Text Classification in Python [Review of Understanding Text Classification in Python]. Data Camp. https://www.datacamp.com/tutorial/text-classification-python

Friedman, R., Dankin, L., Hou, Y., Aharonov, R., Katz, Y., & Slonim, N. (2021, November 1). Overview of the 2021 Key Point Analysis Shared Task (K. Al-Khatib, Y. Hou, & M. Stede, Eds.). ACLWeb; Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.argmining-1.16

How to interpret a confusion matrix for a machine learning model. (n.d.). Www.evidentlyai.com. https://www.evidentlyai.com/classification-metrics/confusion-matrix

Kane, F., & Kane, F. (2020). Machine Learning, Data Science and Deep Learning with Python. Udemy; Udemy. https://www.udemy.com/course/data-science-and-machine-learning-with-python-hands-on/

Krippendorff's Alpha Calculator (K-Alpha) - Official Website. (n.d.). Www.k-Alpha.org. Retrieved May 8, 2024, from https://www.k-alpha.org/krippendorffs-alpha-calculator-k-alpha-official-website

MeasuringU: How to Code & Analyze Verbatim Comments. (2017). Measuringu.com. https://measuringu.com/code-verbatim/

pandas.Series.str.contains — pandas 1.4.2 documentation. (n.d.). Pandas.pydata.org. https://pandas.pydata.org/docs/reference/api/pandas.Series.str.contains.html

Reaching saturation point in qualitative research. (2016, July 21). Quirkos Blog. https://www.quirkos.com/blog/post/saturation-qualitative-research-guide/

S, Y. (2020, May 8). An Introduction to Naïve Bayes Classifier. Medium. https://towardsdatascience.com/introduction-to-na%C3%AFve-bayes-classifier-fa59e3e24aaf

Pryke, B. (2019, August 22). *Jupyter Notebook for Beginners Tutorial*. Dataquest. https://www.dataquest.io/blog/jupyter-notebook-tutorial/

Keyword Search

- This script searches the comments for terms that are frequently found in survey comment responses.
- If a comment contains one of the keywords the comment is assigned a category label and written to a data frame for that category.
- This script assigns both primary and secondary categories to comments.
- If a comment does not match any of the defined categories it is placed in the "Other" category so that the researcher can review those comments to identify new topics or new string patterns to add to the existing categories.
- This script is useful for responses that contain specific product names.

```
# Import Python libraries
import pandas as pd
import numpy as np
import re

# Reading csv file into a data frame
df = pd.read_csv(r"comment_file.csv")

#How to add column to data frame if you want to
df['Survey Period'] = '1Q24'

#Removes all rows without a comment
df['Comment_Column'].fillna(0, inplace = True)
df = df[df.Comment_Column != 0]

#Remove unwanted columns from data frame and only keeps the ones specified between the brackets
df = df.reindex(columns=['Survey Period','response_id','Comment_Column'])

#Finds comments that mention these products and copies them to a new data frame with their product label.
anaconda_df = all_surveys_df[all_surveys_df['Tools_Needed'].str.contains('anaconda', na=False, case=False)]
anaconda_df['Topic'] = 'Anaconda'

angular_df = all_surveys_df[all_surveys_df['Tools_Needed'].str.contains('angular', na=False, case=False)]
angular_df['Topic'] = 'Angular'

#Merging all the topic data frames together
frames = [ anaconda_df, angular_df ]

#Writes labeled comments to csv
coded_tools_df = pd.concat(frames)
coded_tools_df.to_csv(r"/coded_comments.csv",index=False)

#Identifies comments that were not coded using existing keyword scripts so script can be updated to find new keywords
cond = all_surveys_df['Tools_Needed'].isin(coded_tools_df['Tools_Needed'])
all_surveys_df.drop(all_surveys_df[cond].index, inplace = True)
uncat_df = all_surveys_df
uncat_df['Topic'] = 'Other'
uncat_df.to_csv(r"/uncategorized_comments.csv",index=False)

# Writes all comments to single csv file
frames = [coded_tools_df, uncat_df]
all_comments_df = pd.concat(frames)
all_comments_df = all_comments_df.reindex(columns=['Survey Period','response_id','Comment_Column','Topic'])
all_comments_df.to_csv(r"/Tools_Needed_all.csv",index=False)
```

Exploring Jupyter Notebooks

- Go to: https://jupyter.org/try-jupyter/lab/