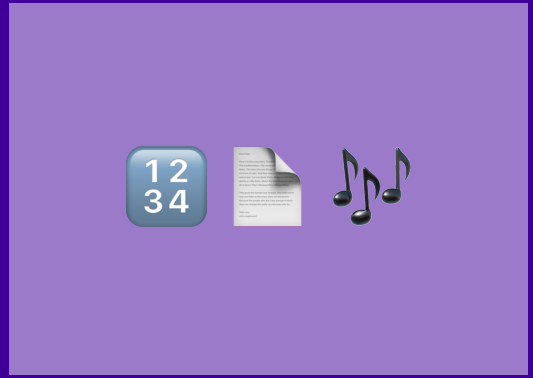
SPICED

Spiced Data Science

Week 4: Text Classification

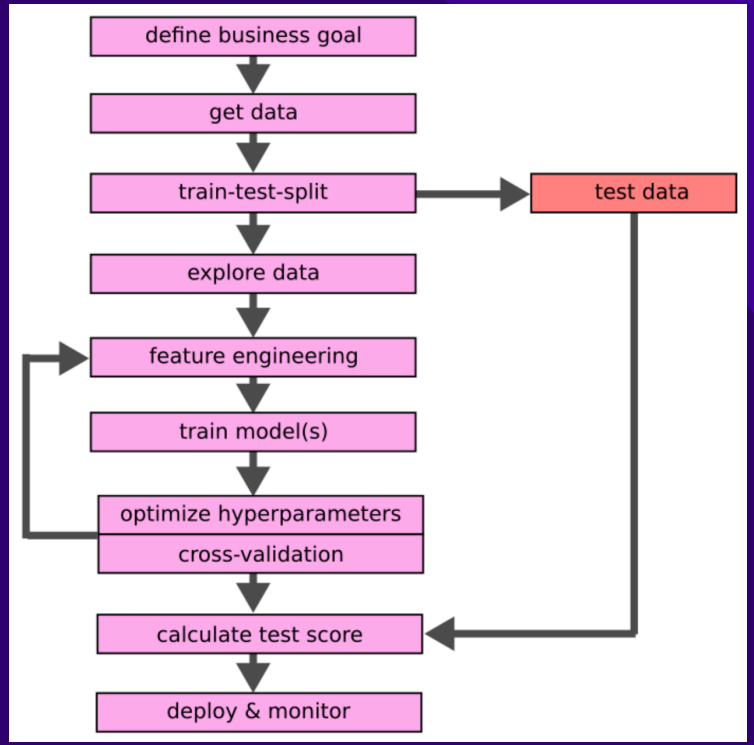


SPICED

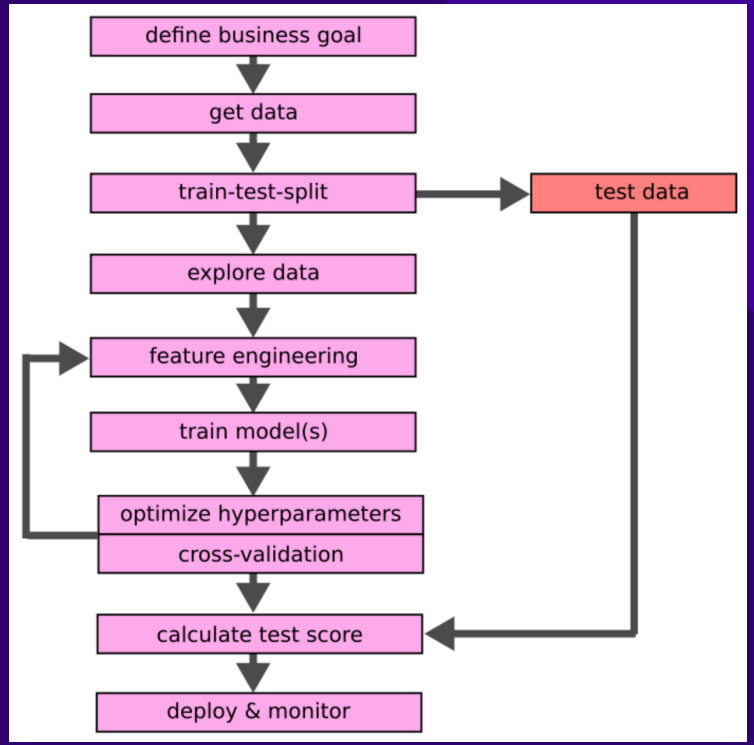
Project of the Week:

```
🍺 $ python song_classifier.py -t 'inconceivable hummingbird strawberries'

The model predicts that the sample text belongs to Fleet Foxes,
with a probability of 80.3%.

🍺 $
```

Roadmap for Text Classification



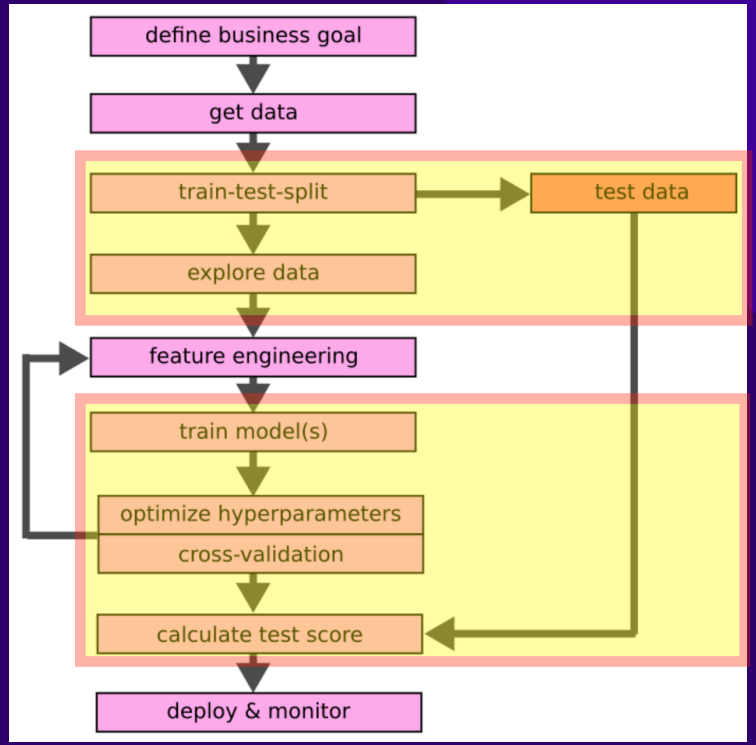
Roadmap for Text Classification



Your Schedule:

	AM	PM
Mon.	Web Scraping	Regular Expressions
Tues.	Parsing HTML	Bag of Words
Wed.	Class Imbalance	Writing Python Functions
Thurs.	Recap Session	Command-Line Interface
Fri.	Naive Bayes	Project Review / Presentations

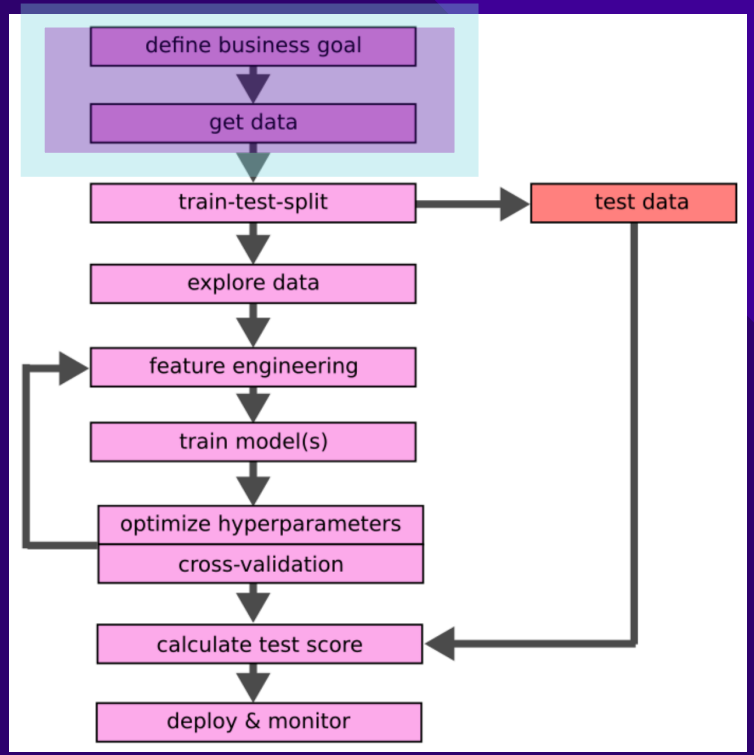
Roadmap for Text Classification



Your Schedule:

	AM	PM
Mon.	Web Scraping	Regular Expressions
Tues.	Parsing HTML	Bag of Words
Wed.	Class Imbalance	Writing Python Functions
Thurs.	Recap Session	Command-Line Interface
Fri.	Naive Bayes	Project Review / Presentations

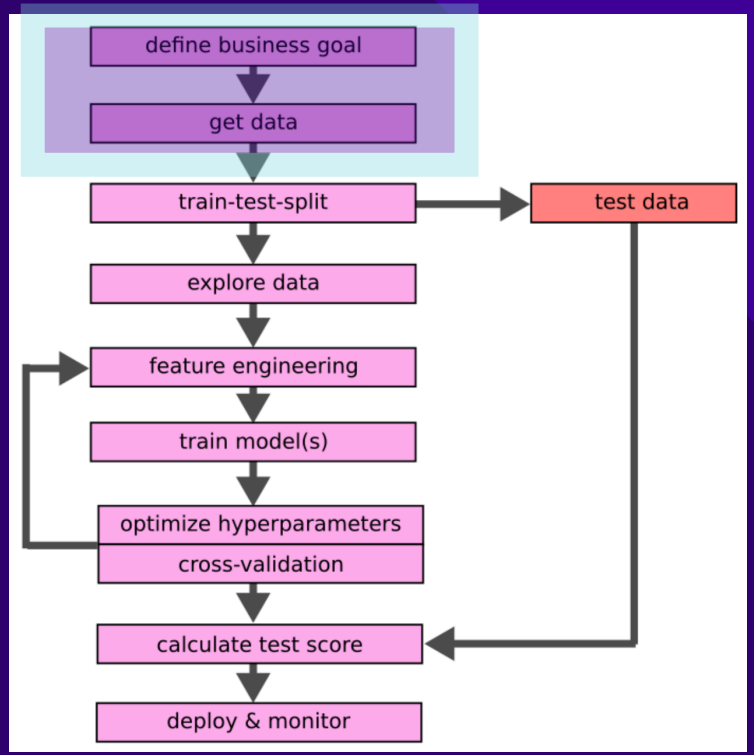
Roadmap for Text Classification



Your Schedule:

	AM	PM
Mon.	Web Scraping	Regular Expressions
Tues.	Parsing HTML	Bag of Words
Wed.	Class Imbalance	Writing Python Functions
Thurs.	Recap Session	Command-Line Interface
Fri.	Naive Bayes	Project Review / Presentations

Roadmap for Text Classification

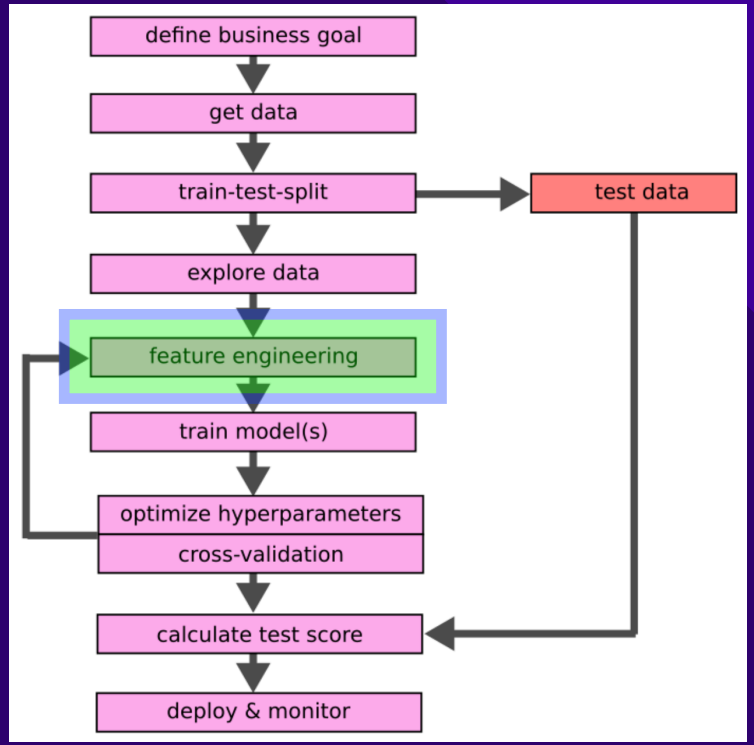


Web Scraping: download pages with lyrics
Regular Expressions: extract links to songs
Parsing HTML: extract lyrics from song pages

Your Schedule:

	AM	PM
Mon.	Web Scraping	Regular Expressions
Tues.	Parsing HTML	Bag of Words
Wed.	Class Imbalance	Writing Python Functions
Thurs.	Recap Session	Command-Line Interface
Fri.	Naive Bayes	Project Review / Presentations

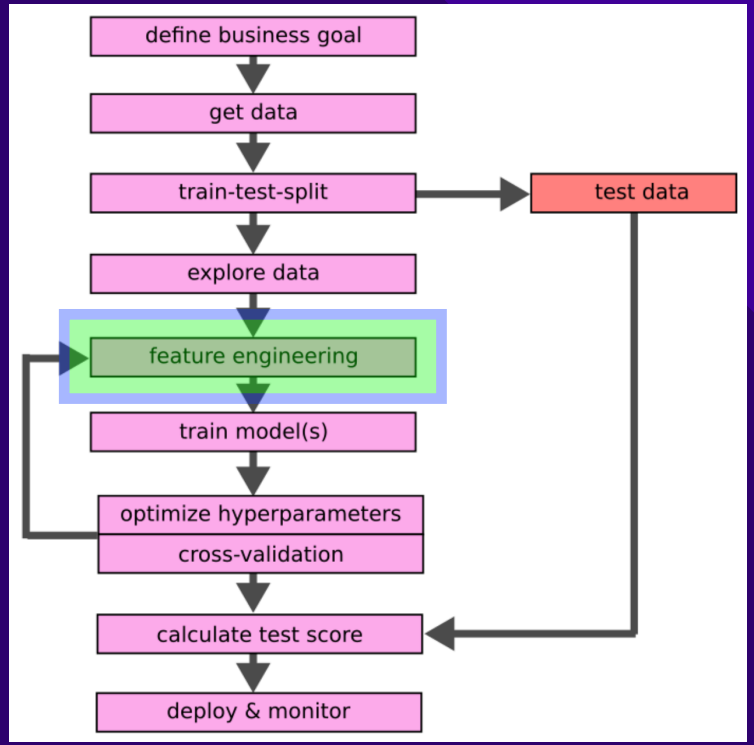
Roadmap for Text Classification



Your Schedule:

	AM	PM
Mon.	Web Scraping	Regular Expressions
Tues.	Parsing HTML	Bag of Words
Wed.	Class Imbalance	Writing Python Functions
Thurs.	Recap Session	Command-Line Interface
Fri.	Naive Bayes	Project Review / Presentations

Roadmap for Text Classification

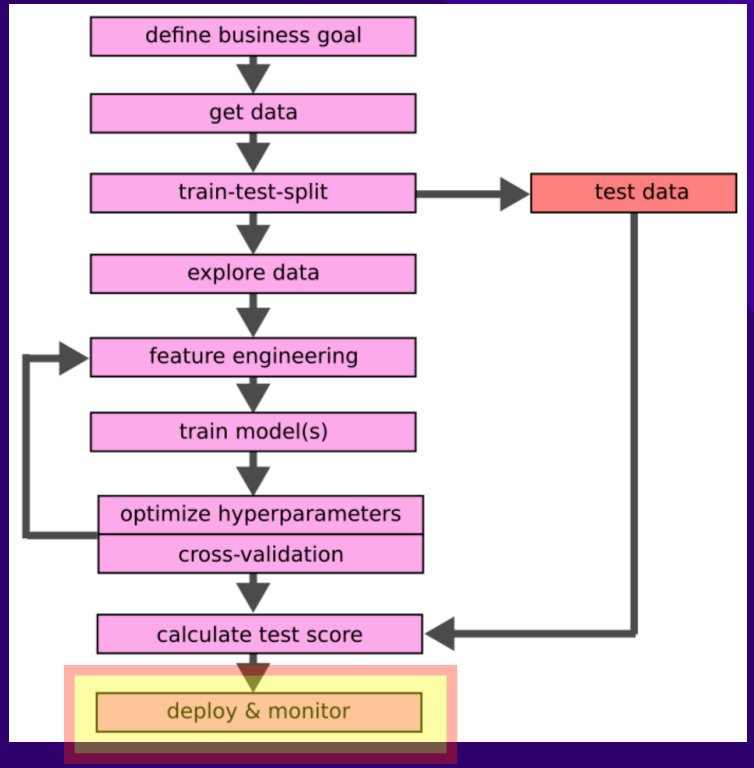


Bag of Words: feature engineer / vectorize corpus
Class Imbalance: balance the dataset

Your Schedule:

	AM	PM
Mon.	Web Scraping	Regular Expressions
Tues.	Parsing HTML	Bag of Words
Wed.	Class Imbalance	Writing Python Functions
Thurs.	Recap Session	Command-Line Interface
Fri.	Naive Bayes	Project Review / Presentations

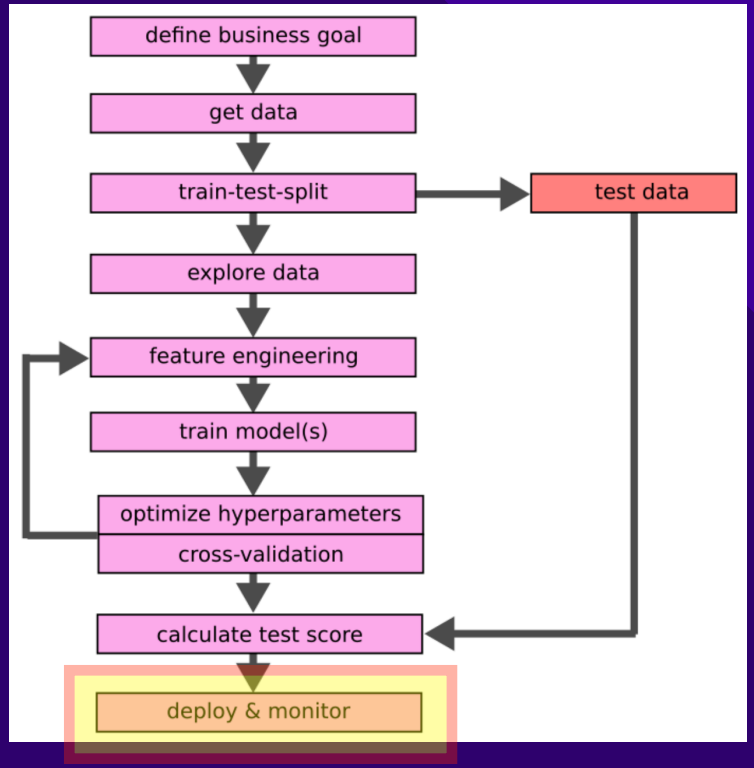
Roadmap for Text Classification



Your Schedule:

	AM	PM
Mon.	Web Scraping	Regular Expressions
Tues.	Parsing HTML	Bag of Words
Wed.	Class Imbalance	Writing Python Functions
Thurs.	Recap Session	Command-Line Interface
Fri.	Naive Bayes	Project Review / Presentations

Roadmap for Text Classification

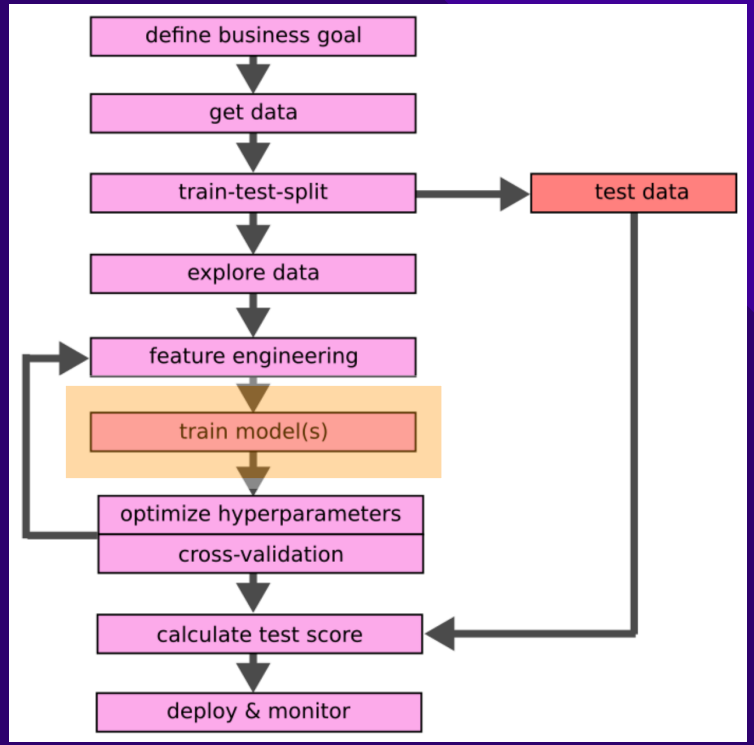


Python Functions: structure your code
Command Line Interface: write a CLI

Your Schedule:

	AM	PM
Mon.	Web Scraping	Regular Expressions
Tues.	Parsing HTML	Bag of Words
Wed.	Class Imbalance	Writing Python Functions
Thurs.	Recap Session	Command-Line Interface
Fri.	Naive Bayes	Project Review / Presentations

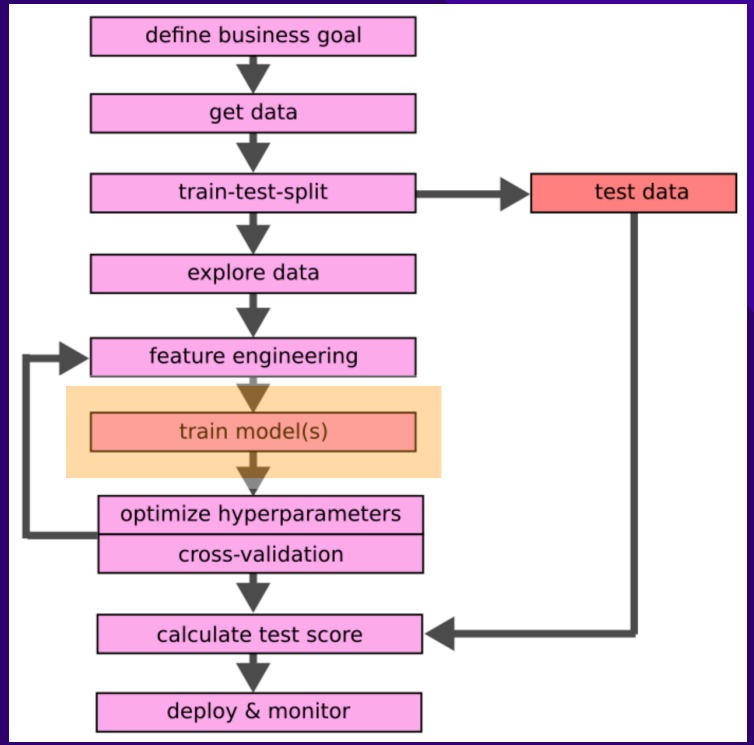
Roadmap for Text Classification



Your Schedule:

	AM	PM
Mon.	Web Scraping	Regular Expressions
Tues.	Parsing HTML	Bag of Words
Wed.	Class Imbalance	Writing Python Functions
Thurs.	Recap Session	Command-Line Interface
Fri.	Naive Bayes	Project Review / Presentations

Roadmap for Text Classification

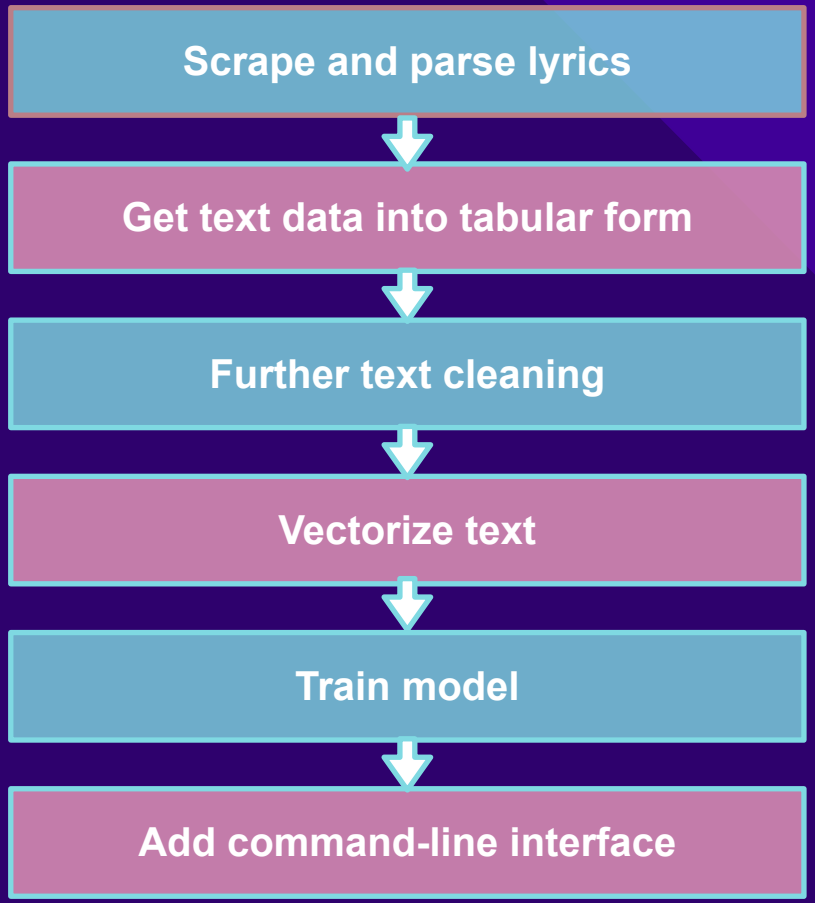


Naive Bayes: train a MultinomialNB model

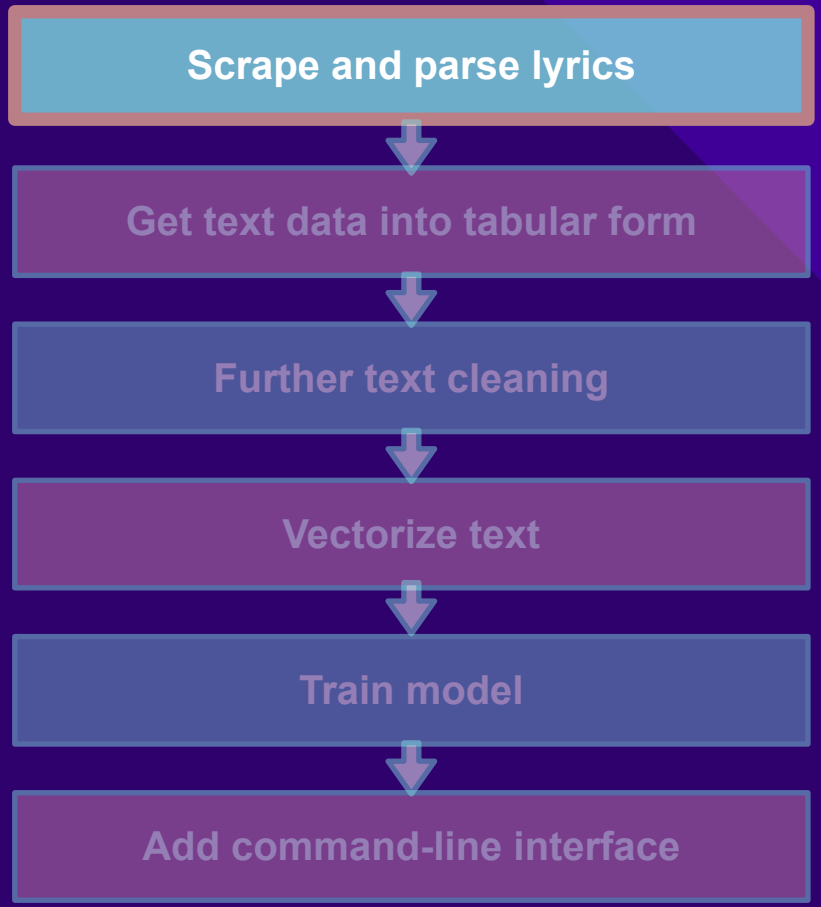
Your Schedule:

	AM	PM
Mon.	Web Scraping	Regular Expressions
Tues.	Parsing HTML	Bag of Words
Wed.	Class Imbalance	Writing Python Functions
Thurs.	Recap Session	Command-Line Interface
Fri.	Naive Bayes	Project Review / Presentations

More Detailed Roadmap



More Detailed Roadmap



Tools used:

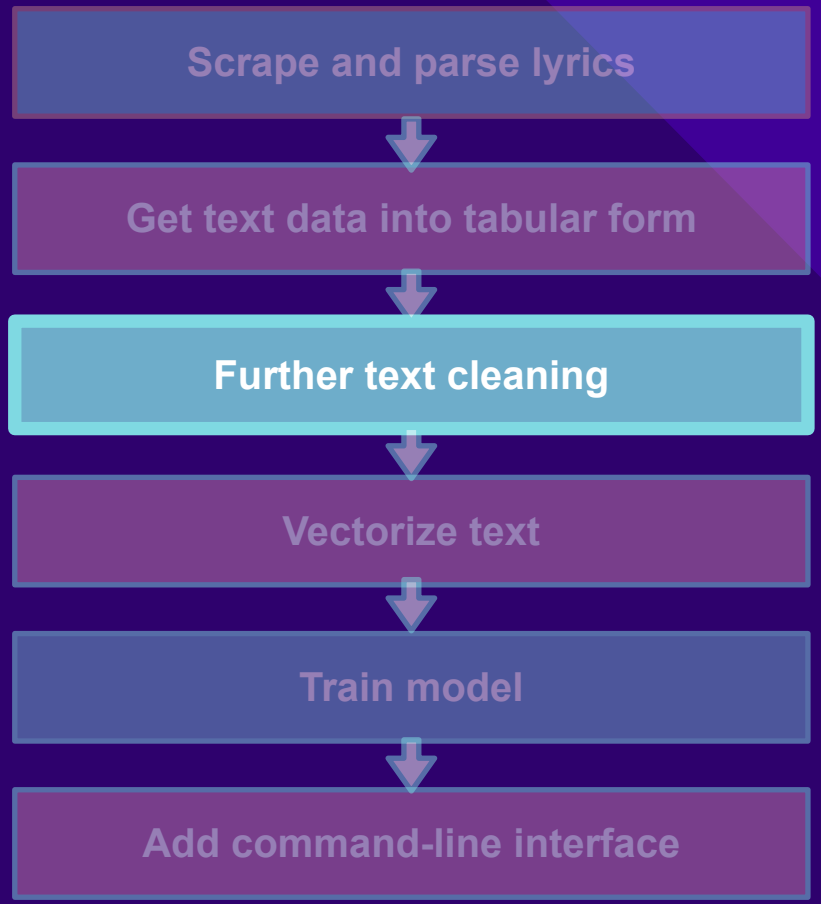
- **requests + Regular Expressions**
- **requests + BeautifulSoup (Parsing HTML)**
- **Scrapy (bonus!)**

More Detailed Roadmap

- Scrape and parse lyrics
- **Get text data into tabular form**
- Further text cleaning
- Vectorize text
- Train model
- Add command-line interface

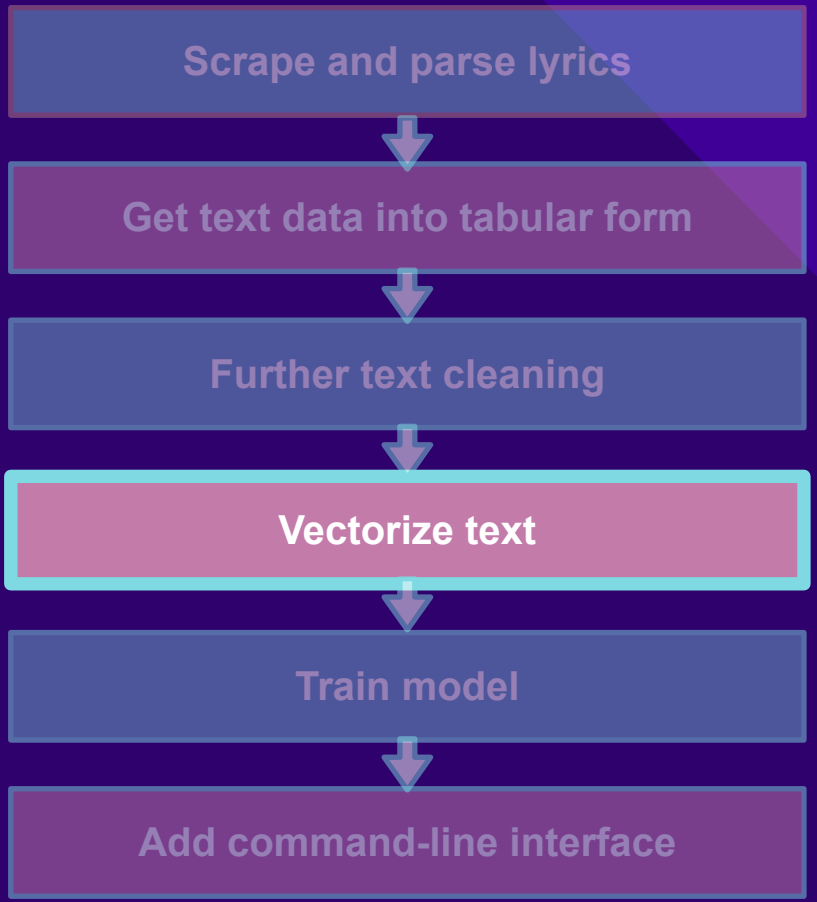
List of Lyrics (X)	List of Artists (y)
[“hello darkness my old friend…”,	[“simon and garfunkel”,
“…we will we will rock you…”,	“queen”,
“…i see a little silhouetto of a man…”,	“queen”,
“you are my fire the one desire…”]	“backstreet boys”]

More Detailed Roadmap



Tools used:
- **Lemmatization**
- **RegEx / lots of string cleaning!!**

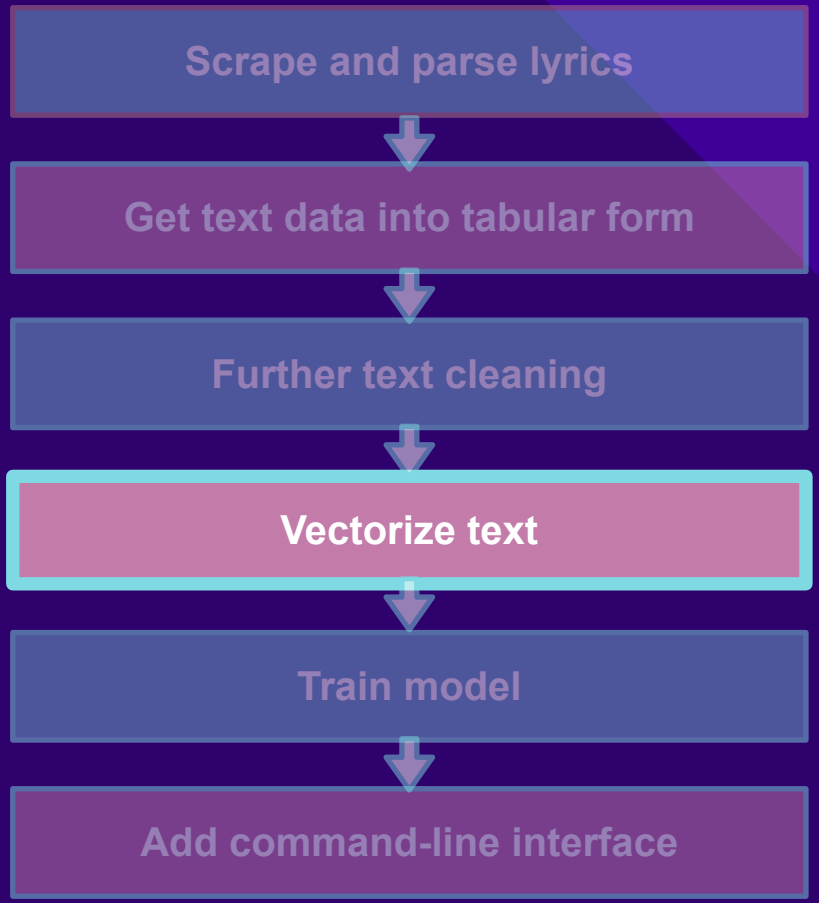
More Detailed Roadmap



Bag of Words (BOW) Lesson:
- **CountVectorizer**
- **TfIdfTransformer**

- *(or simply the TfIdfVectorizer)*

More Detailed Roadmap

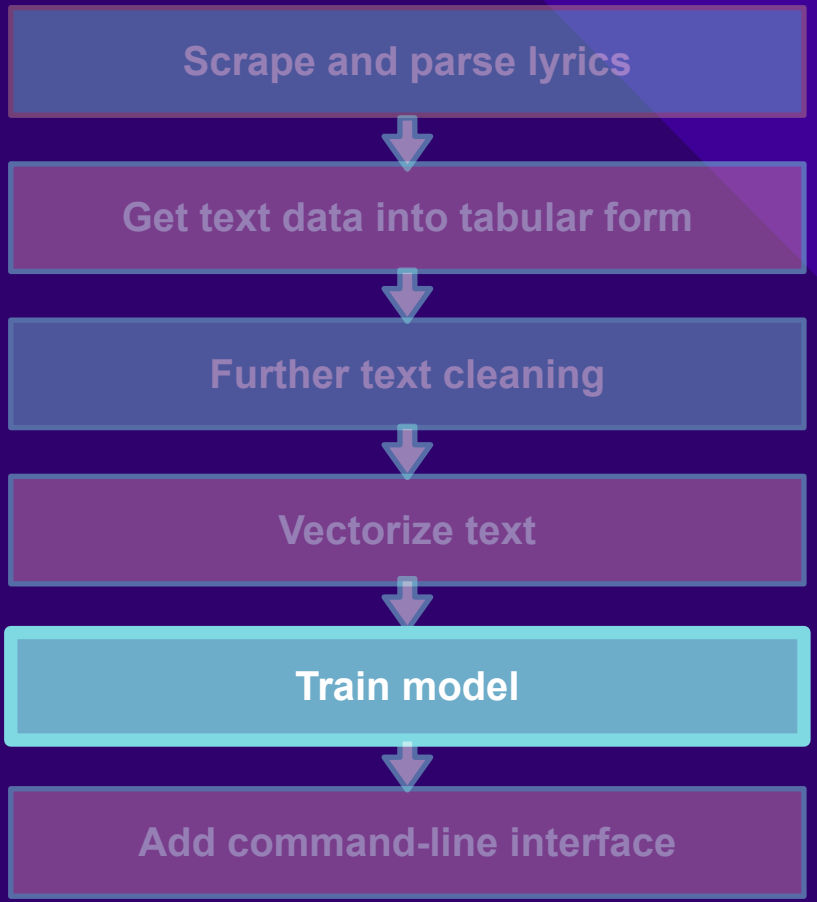


Bag of Words (BOW) Lesson:
- **CountVectorizer**
- **TfIdfTransformer**

- *(or simply the TfIdfVectorizer)*

Class Imbalance techniques (e.g. SMOTE) can then be applied on the vectorized dataframe.

More Detailed Roadmap



Lots of Options:
- LogisticRegression
- RandomForest
- GradientBoostingClassifier

- NEW: Naive Bayes!

More Detailed Roadmap

Important Lessons:

- **Writing Python Functions** (laying a solid foundation)
- **Command-Line-Interface**