# Creating a Machine Learning Pipeline to Support Survey Analysis

FEDCASIC WORKSHOP - 04.06.2022

Brian Francis Sadacca
*Accenture Federal Services*

Joanna Fane Lineback, Elizabeth May Nichols
*US Census Bureau*

*All data presented are from publicly available sources and are not titled*

# Workshop Overview

**Introduction:**

*What is Machine Learning, how can it support survey analysis, and how can researchers find what works?*

**Techniques overview:**

*What does a machine learning pipeline for text and audio analysis look like? What can we do right now?*

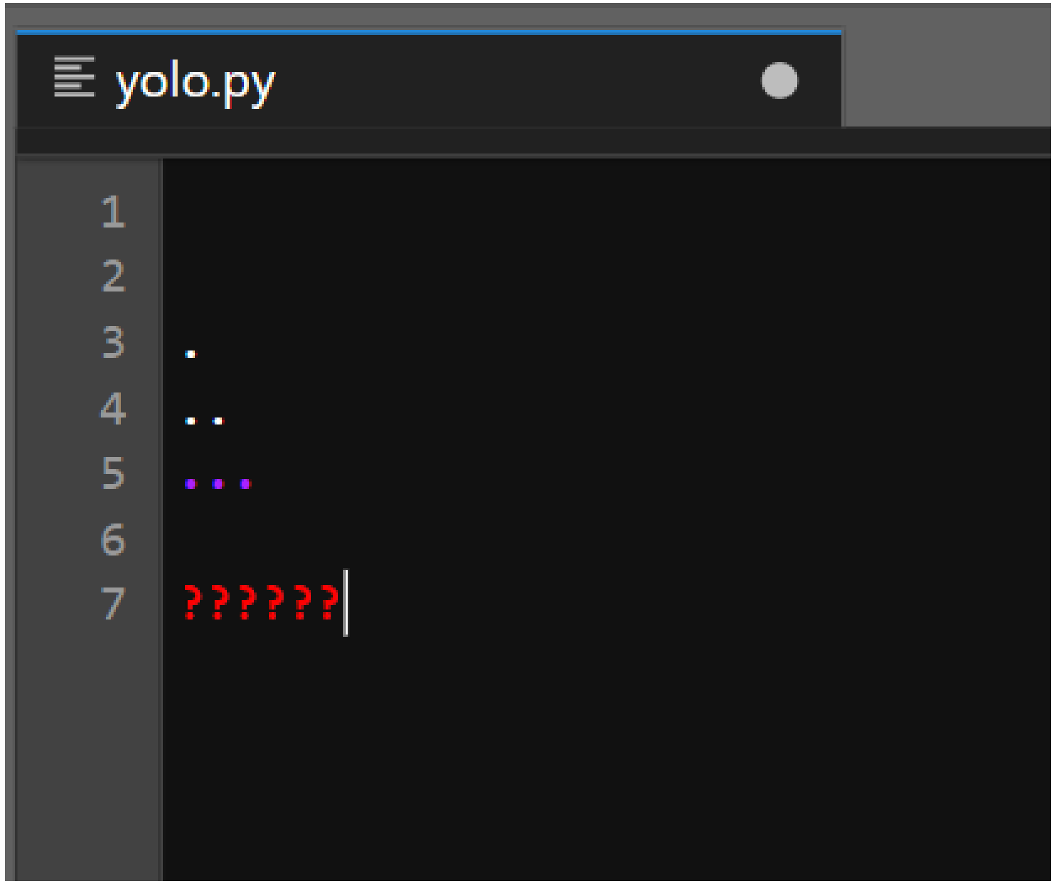
**Practical Example**

*How do we design, build, evaluate, and optimize models to answer research questions?*

# Why is Machine Learning exciting?

**Try to write a program:**

- that takes a picture as an input
- outputs names for all the objects in that picture
- and outputs where those objects are in the picture



# Why should *we* care about machine learning?

**For the research superpowers:** ML can give us insights and quantitative evaluation of datasets difficult to summarize by human review alone!

**It’s everywhere:** ML is here at FEDCASIC (Twitter Research! Survey sentiment! Frame development! Call analytics!) and has been for years!

**It’s gotten good:** ML models can now give insights into ‘unstructured’ text, audio, and images (in addition to ‘structured’ tabular data).

# What is Machine Learning (ML)?

Machine learning models **find predictive patterns** in data

**from the data themselves** (unsupervised machine learning)

or **with the help of labels** (supervised machine learning)

**( How is it different from statistics? )**

*Lots of overlap! Similar approaches, but generally…*

- *Statistics focuses on understanding relationships*
- *Machine learning focuses on making predictions*

# Machine learning to predict based on examples

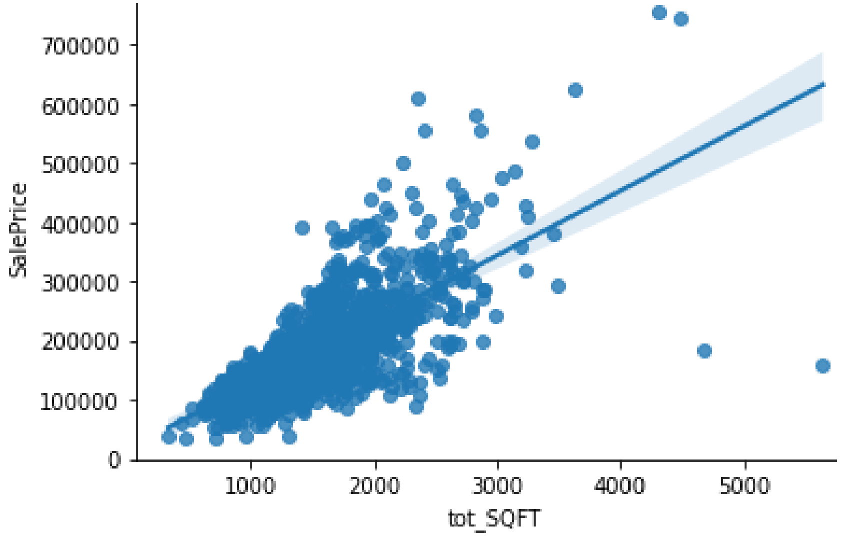
(supervised ML)

**Classification:** *is this safe to eat?*

*NO!*

*Yes.*

**Regression:** *is this house expensive given its size?*



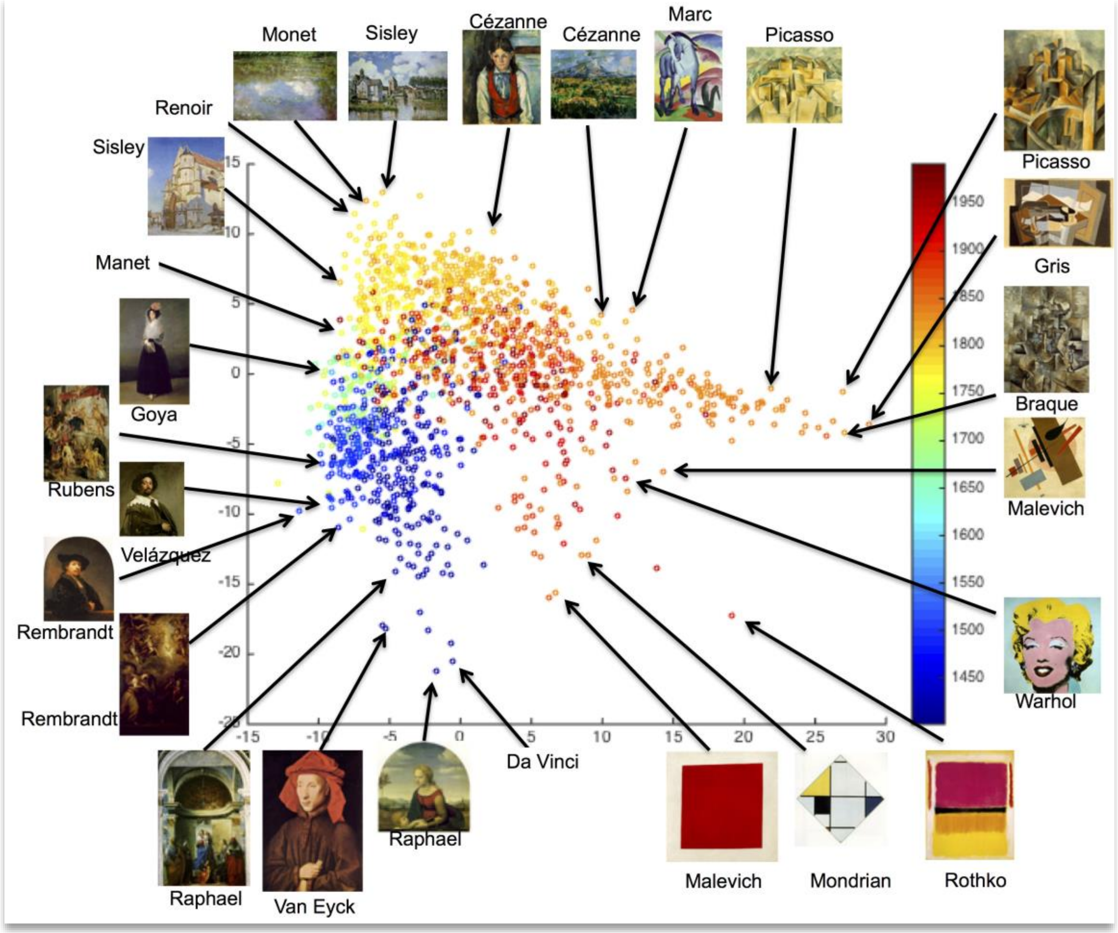

Home sale price data sourced from: De Cock, Dean. "Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project." *Journal of Statistics Education* 19.3 (2011).

# Machine learning to find patterns from the data themselves

(unsupervised ML)

**Dimensionality reduction:** *is there a simple way to describe all the similarities and differences?*

**Clustering:** *how many groups are there?*




Elgammal, Ahmed, et al. "The shape of art history in the eyes of the machine." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. No. 1. 2018.

# What are some ways I could apply these to survey data?

**Free text?**

**Audio/video interviews?**

**Tabular data?**

**Survey metadata?**

**Find common question answers**
(clustering! classification!)

**Transcribe, recognize emotions**
(classification!)

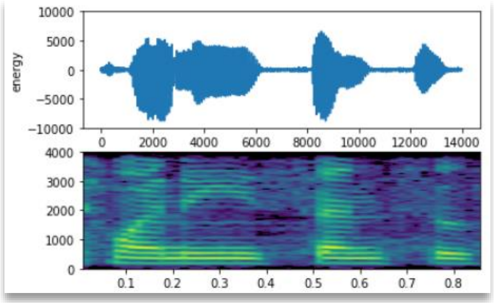
**Predict values based on partial data**
(clustering, classification, and/or regression!)

**Behavioral patterns identification**
(dimensionality reduction! clustering!)

# An example pipeline for survey audio

Raw audio

1



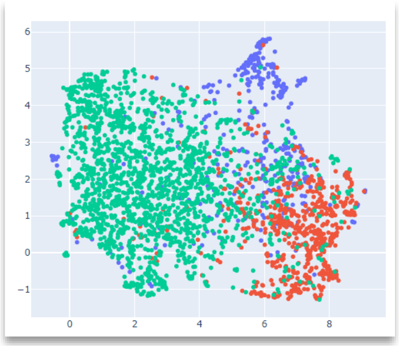
Pre-processed audio

2

“may I record this call for quality assurance purposes?”

Transcript

3



Call similarity, clusters

4

5

Identify positive & negative emotion on calls

6

7

8

Identify call topics, Identify nearest FAQ

**Approaches used in pipeline:**

1. Normalize audio gain and split channels
2. Transcribe audio for each channel
3. Identify similarity among transcripts with ‘word embedding’ models
4. Identify emotion from audio
5. Identify emotion from text
6. Classify calls as negative or positive for operational use
7. Identify common groups of calls, and summarize groups by topic
8. Identify FAQs related to call content for operational use, identify common topics over time

# An example pipeline for open-ended questions

“well, the first time I applied for a job…”

3

free-text answer

text similarity, clusters

5

7

Identify positive & negative emotion

6

8

Identify text topics
Identify ‘entities’ discussed (e.g., people, places)

**Approaches used in pipeline:**

1. ~~Normalize audio gain and split channels~~
2. ~~Transcribe audio for each channel~~
3. ~~Identify similarity among transcripts with ‘word embedding’ models~~
4. ~~Identify emotion from audio~~
5. Identify emotion from text
6. Classify calls as negative or positive for operational use
7. Identify common groups of calls, and summarize groups by topic
8. Identify FAQs related to call content for operational use, identify common topics over time

# How can I find what's out there to use?

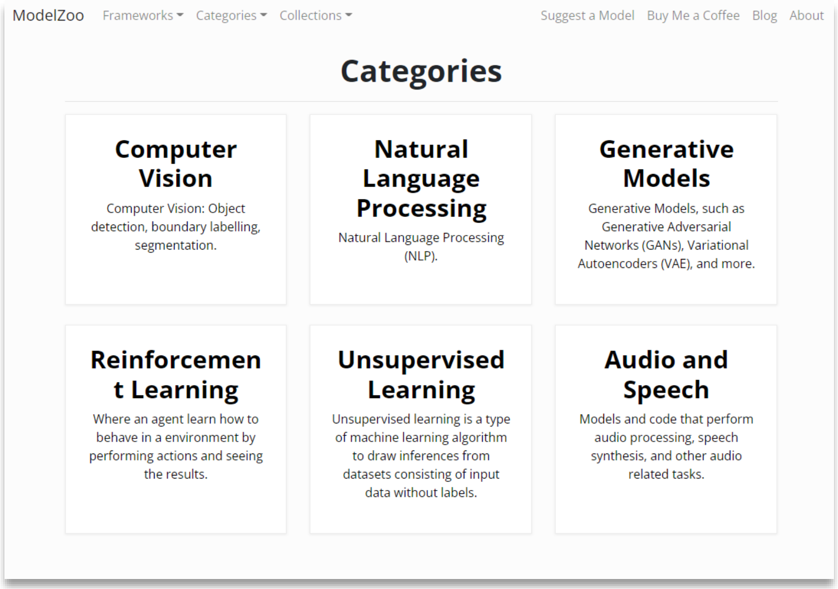
**People have:**

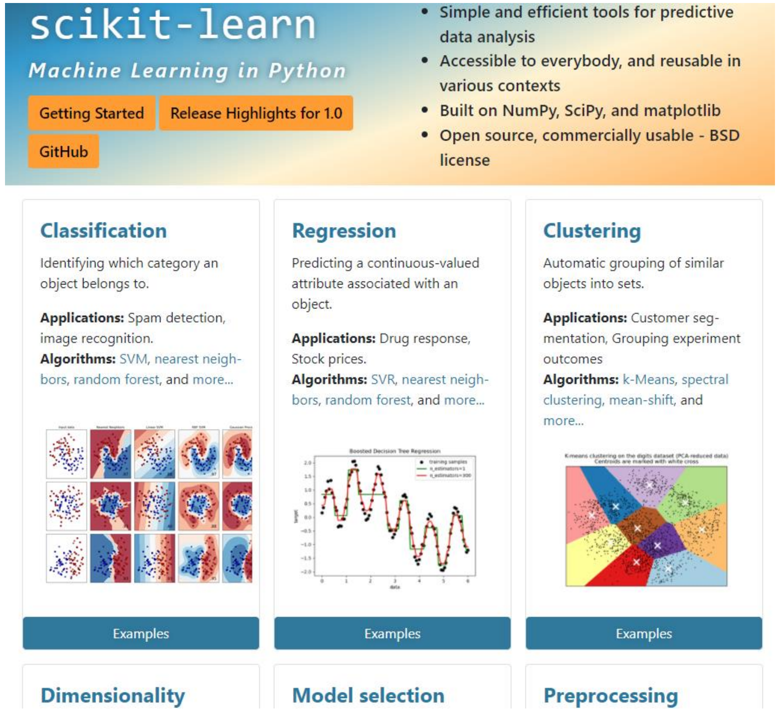
- trained and share models
- built frameworks
- and sell commercial ML solutions

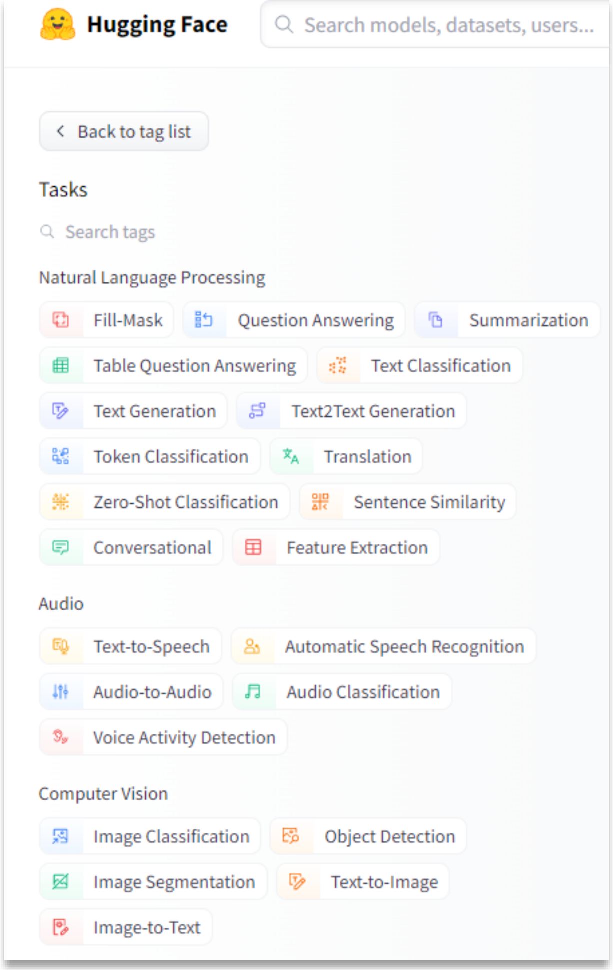
*You do not need to reinvent the wheel (or be a machine-learning engineer) to use advanced ML for research!*

# What's out there?

## There are open-source frameworks and repositories that give a menu of options – play with them!



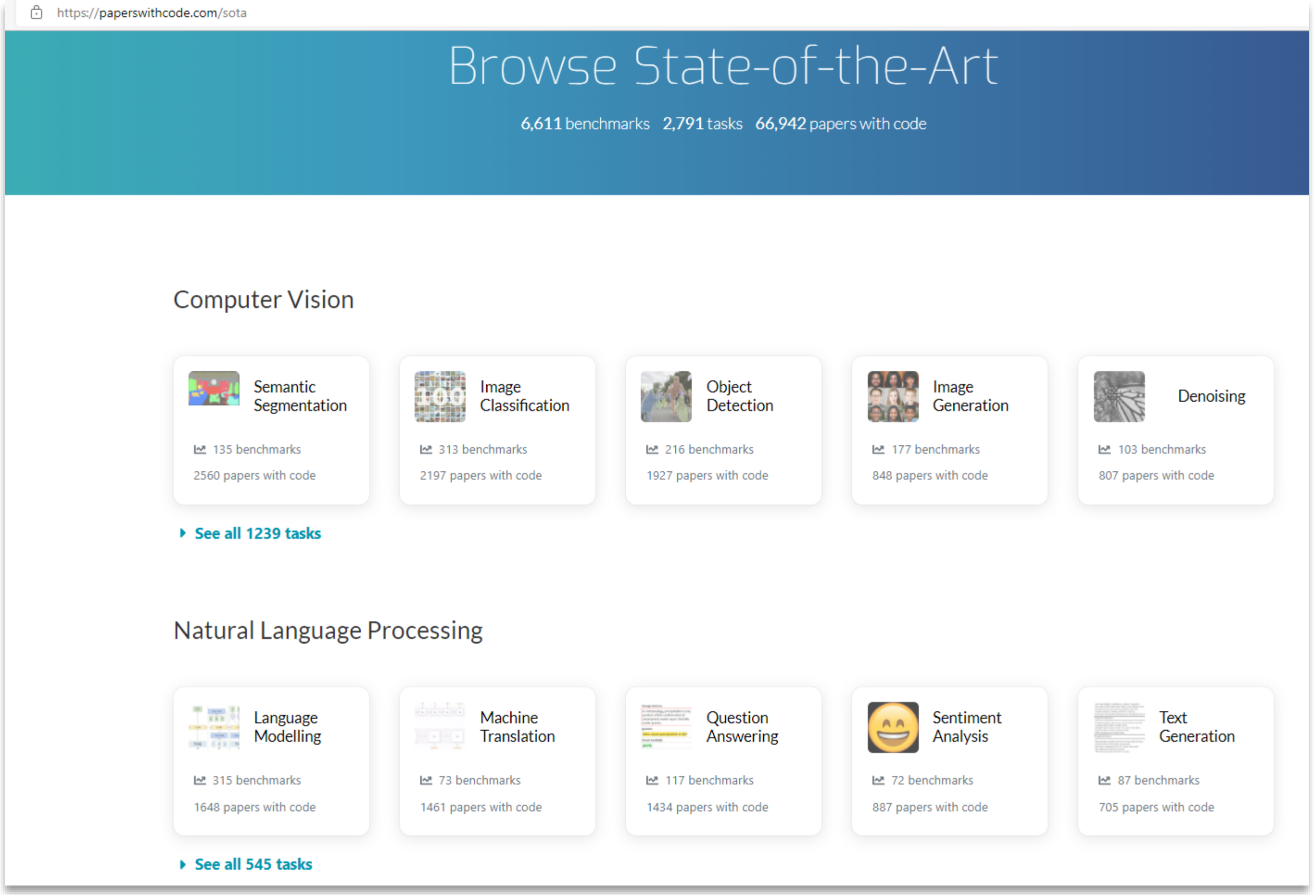

NLTK

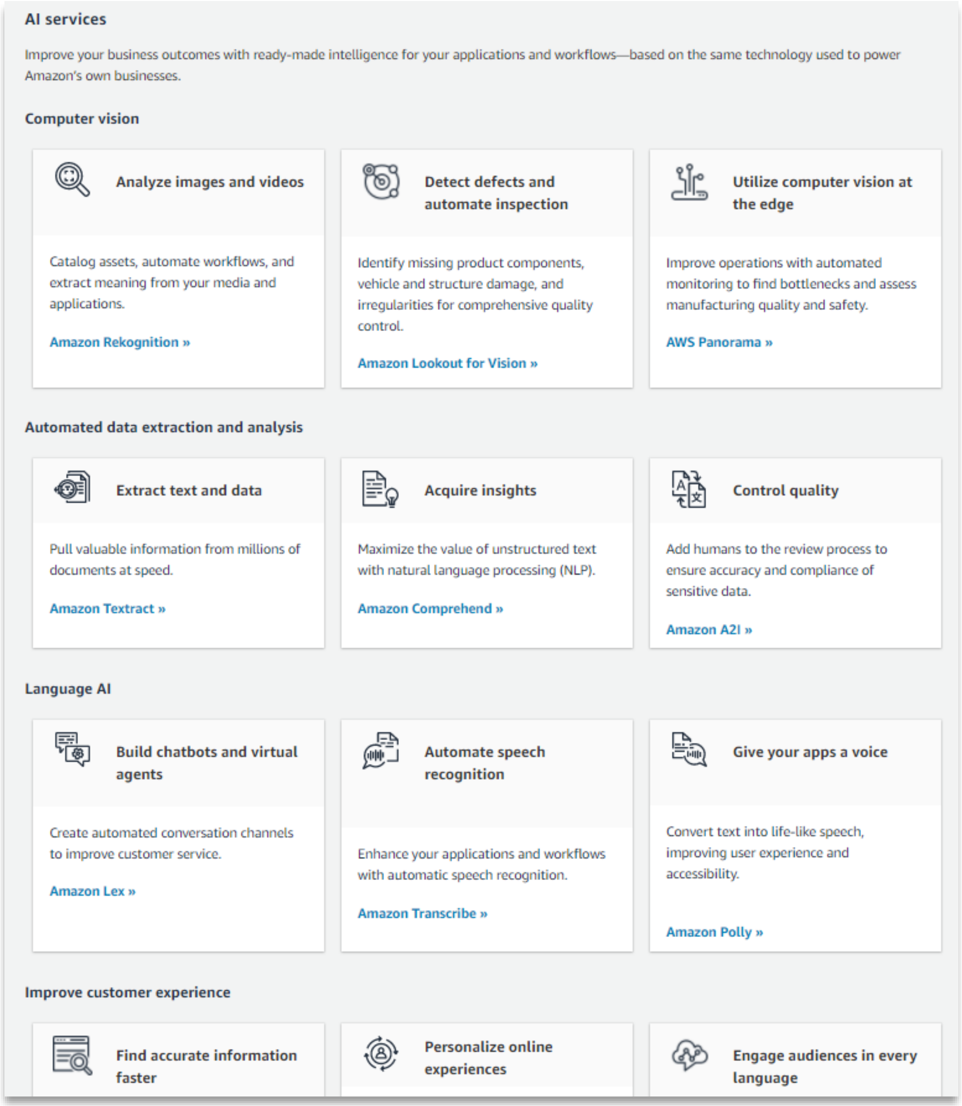
spaCy

# What’s out there?

**Commercial offerings from the big cloud providers (e.g., AWS, below), and a range of big and small tech companies can deploy without code:**

**Aggregators link papers/models across methods (e.g., paperswithcode)**



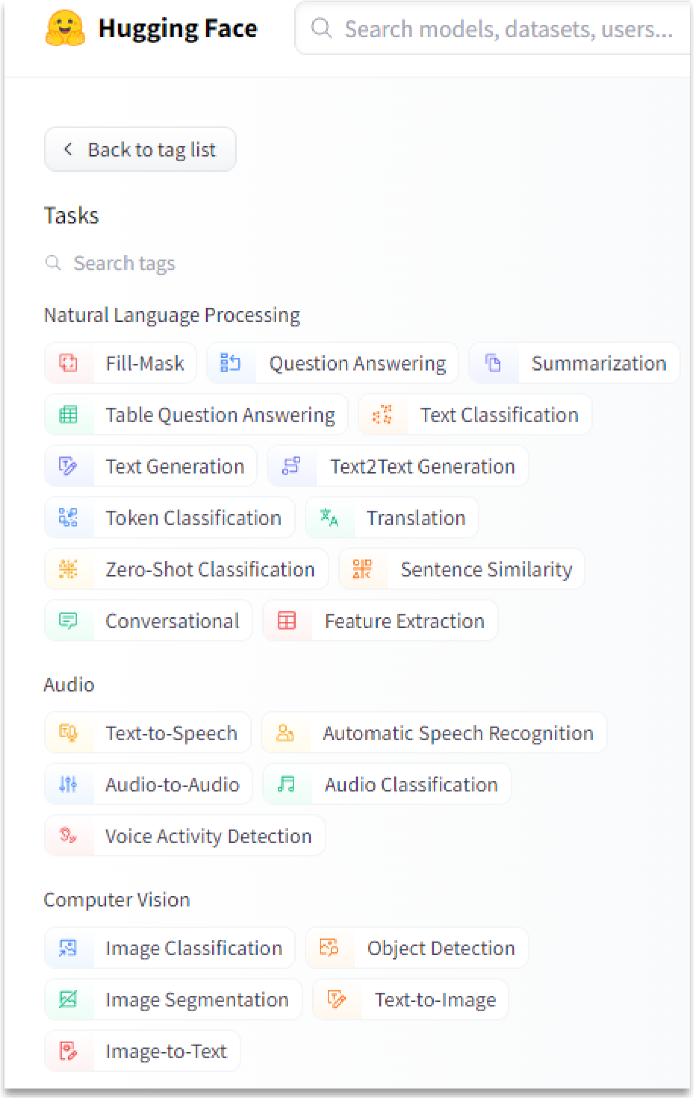
# What’s out there?

**This all seems easy – can I just use something off the shelf, on autopilot?**

***For each model, you still need to:***

1. **Know your data (and its limitations).**
2. **Make sure the model is effective for its purpose.**
3. **Have an intuition on why a model gives a given result for each step in a pipeline.**
4. **Evaluate each model’s performance beyond the KPI: for issues of bias, reliability, security, and transparency.**

# An example framework: HuggingFace




## HuggingFace model-hub API

Transcription Example

Sentiment Example

# Ok, so how can I actually do this, today?

## Worked example: Building an interview processing pipeline

- **Define Problem and design plan:** Clearly define goals of models to ensure you have a plan for checking if/how models meet needs.

- **Develop pipeline with models:** Frameworks make it easier, be aware but your use case may not match, so start small, and make sure to test, plot, and try to understand how/why the models succeed/fail before growing too fast (in amount of data, complexity, and capabilities).

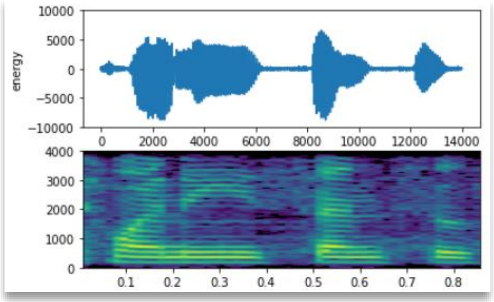
- **Evaluate models:** Curate labels, qualitative and quantitative and check performance at each step. Evaluate if error at an early step impacts later steps (to define when models are 'good enough'). Plot/explore data in as raw a form as possible at each step allows iterative sanity checks. In addition to 'key' indicators (KPIs), also review for issues of bias, reliability, security, and transparency.

- **Productionalize models:** Document everything, including individual models, their integration with other models and systems, data used to train/test.  Evaluate performance (speed vs accuracy, implementation) considerations, deployment methods (automation?), and future iterative testing to plan for model drift if in production for sustained period.

# Defining the plan for analysis of a survey interview or open-ended question

Raw audio

1




Pre-processed audio

2

"the first job I ever had was …"

Interview (or open-ended question) text

3

Interview similarity, clusters

4

5

What topics and entities are discussed in interviews or open-ended questions?

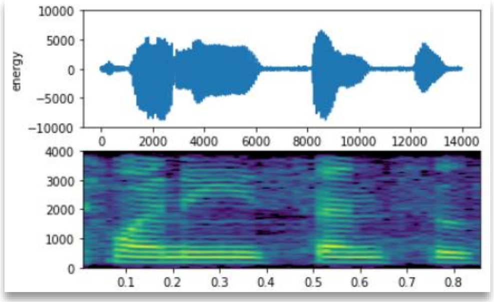
**Approaches needed for 'interview' pipeline:**

1. Resample to standard 16k rate
2. Transcribe audio
3. Identify similarity among transcripts with 'word embedding' models
4. Identify common groups of interviews
5. Identify interview topics from groups; identify named entities from groups

# Defining the plan for analysis of a survey interview or open-ended question

Raw audio

1



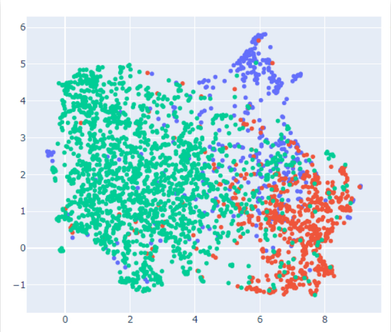

Pre-processed audio

2

“the first job I ever had was …”

Interview (or open-ended question) text

3



Interview similarity, clusters

4

5

What topics and entities are discussed in interviews or open-ended questions?

**Approaches needed for ‘interview’ pipeline:**

1. Resample to standard 16k rate
   **TEST: does the audio sound audible?**
2. Transcribe audio
   **TEST: error rate against known transcripts**
3. Identify similarity among transcripts with ‘word embedding’ models
   **TEST: are similar interviews ‘nearby’? Are probe words grouped?**
4. Identify common groups of interviews
   **TEST: find mutual information between manual and automatic clusters**
5. Identify interview topics from groups; identify named entities from groups
   **TEST: do automatic keywords / entities align with manual labels?**

# Worked example: Building an interview processing pipeline

- **Define Problem and design plan:** Clearly define goals of models to ensure you have a plan for checking if/how models meet needs.

- **Develop pipeline with models:** Frameworks make it easier to prototype, be aware but your use case may not match, so start small, and make sure to test, plot, and try to understand how/why the models succeed/fail before growing too fast (in amount of data, complexity, and capabilities).

- **Evaluate models:** Curate labels, qualitative and quantitative and check performance at each step. Evaluate if error at an early step impacts later steps (to define when models are 'good enough'). Plot/explore data in as raw a form as possible at each step allows iterative sanity checks. In addition to 'key' indicators (KPIs), also review for issues of bias, reliability, security, and transparency.

- **Productionalize models:** Document everything, including individual models, their integration with other models and systems, data used to train/test.  Evaluate performance (speed vs accuracy, implementation) considerations, deployment methods (automation?), and future iterative testing to plan for model drift if in production for sustained period.

# Worked example: Building an interview processing pipeline

## DEMO

### Transcribe Text

This initial segment transcribes text with a single-line call to the pipeline functionality.

Pipelines (and other libraries built-in functionality) simplify the use of models, but lose the ability to customize funtionality and often lead to worse overall performance.

The subsequent call uses more verbose code to integrate multiple models (including vocabulary restrictions) in the ASR pipeline.

```
[ ]  from transformers import pipeline

     #two lines can generate a simple transcript
     speech_recognizer = pipeline("automatic-speech-recognition", model="facebook/wav2vec2-large-960h-lv60-self")
     transcript = speech_recognizer('./dta00306_002.mp3')
```

```
Some weights of Wav2Vec2ForCTC were not initialized from the model checkpoint at facebook/wav2vec2-large-960h-lv60-self and are
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```
[ ]  #display transcript
     print(transcript['text'])
```

```
AND THE WINDES WOULD TAKE IT AND GIVE ITT TO THE SPINNERS FOR THEY ALWAYS HAD A BOY TAKING A BOBB ING TO BRING IT FROM ONE PLACI
```

evaluate transcript versus manual transcription

```
[ ]  import jiwer
     import numpy as np

     #specify the manually transcribed data
     manual_transcript = ["AND THE WINDERS WOULD TAKE IT AND GIVE IT TO TO THE SPINNERS CAUSE THEY ALWAYS HAD A BOY

     #transform the transcripts (remove punctuation, lowercase)
     transformation = jiwer.Compose([
         jiwer.ToLowerCase(),
         jiwer.RemoveWhiteSpace(replace_by_space=True),
         jiwer.RemoveMultipleSpaces()
```

*Demo data analyzed are from interviews from the Occupational Folklife Project of the Library of Congress' American Folklife Center*

# Worked example: Building an interview processing pipeline

- **Define Problem and design plan:** Clearly define goals of models to ensure you have a plan for checking if/how models meet needs.

- **Develop pipeline with models:** Frameworks make it easier, be aware but your use case may not match, so start small, and make sure to test, plot, and try to understand how/why the models succeed/fail before growing too fast (in amount of data, complexity, and capabilities).

- **Evaluate models:** Curate labels, qualitative and quantitative and check performance at each step. Evaluate if error at an early step impacts later steps (to define when models are 'good enough'). Plot/explore data in as raw a form as possible at each step allows iterative sanity checks. In addition to 'key' indicators (KPIs), also review for issues of bias, reliability, security, and transparency.

- **Productionalize models:** Document everything, including individual models, their integration with other models and systems, data used to train/test. Evaluate performance (speed vs accuracy, implementation) considerations, deployment methods (automation?), and future iterative testing to plan for model drift if in production for sustained period.

# How do I know if my models are any good?

1. **Know your data (and its limitations).**
2. **Make sure the model is effective for its purpose.**
3. **Have an intuition on why a model gives a given result for each step in a pipeline.**
4. **Evaluate each model's performance beyond the KPI: for issues of bias, reliability, security, and transparency.** ***This is often called 'Responsible AI'***

# Evaluating Machine Learning with a Responsible AI Focus

## ML use cases in pipeline

**Generating transcripts** to identify interview topics or to review interviewer performance.

**Clustering transcripts** from interviews to identify common interview topics/concerns.

**Identifying emotional content** of responses and interviews to improve questions.

<u>Other survey use cases:</u>

**Predictive modeling** to calculate high-scrutiny metrics; **computer vision** to support digitization; **anomaly detection** to identify collection quality issues

## Opportunities to apply Responsible AI

**Are models/data without bias, or is bias known?** Differences in transcription quality among groups can lead to disparate impacts in discovery or evaluation.

**Are analyses repeatable/reliable/robust?** Clustering/classification of interviews may be sensitive to data used in training and subject to drift over time.

**Are models transparent in how values are calculated?** Auto-generated features need to be human-interpretable for evaluation and for use in survey improvements.

**Is there a reputational risk if the accuracy of models is poor or biased?** If there are issues models focused on organizational ‘core competencies’, there could be reputational impacts to the overall organization.

# Worked example: Building an interview processing pipeline

- **Define Problem and design plan:** Clearly define goals of models to ensure you have a plan for checking if/how models meet needs.

- **Develop pipeline with models:** Frameworks make it easier, be aware but your use case may not match, so start small, and make sure to test, plot, and try to understand how/why the models succeed/fail before growing too fast (in amount of data, complexity, and capabilities).

- **Evaluate models:** Curate labels, qualitative and quantitative and check performance at each step. Evaluate if error at an early step impacts later steps (to define when models are 'good enough'). Plot/explore data in as raw a form as possible at each step allows iterative sanity checks. In addition to 'key' indicators (KPIs), also review for issues of bias, reliability, security, and transparency.

- **Productionalize models:** Document everything, including individual models, their integration with other models and systems, data used to train/test. Evaluate performance (speed vs accuracy, implementation) considerations, deployment methods (automation?), and future iterative testing to plan for model drift if in production for sustained period.

# How do I get my models to production?

## *MLOps:*

1. **Develop a framework for integration with other components (e.g., data sources, other systems, reporting tools)**
2. **Ensure models are *fast enough* for the use-case, and optimize software and hardware implementation as needed**
3. **Automate testing (and retesting)**
4. **Automate re-training (if appropriate)**
5. **Automate deployment approach to reduce manual effort in model updates**
6. **Develop framework for model documentation, sharing, and re-use by others**

# Workshop Review

**Introduction:**

*What is Machine Learning, how can it support survey analysis, and how can researchers find what works?*

**Techniques overview:**

*What does a machine learning pipeline for text and audio analysis look like? What can we do right now?*

**Practical Example**

*How do we design, build, evaluate, and optimize models to answer research questions?*

# Questions?

*For any follow-up questions please reach-out to*
*brian.sadacca@accenturefederal.com*