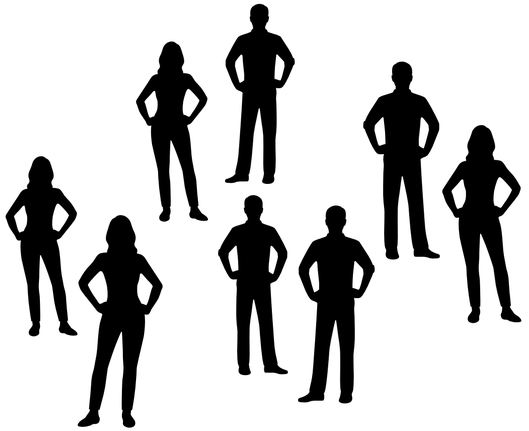
Text Embeddings

And two (practical) use cases

Shaw Talebi

Problem with Text

It isn’t computable!



Feet	Inches
5	7
5	9
5	9
6	2
6	0
5	6
5	10
5	9

Summarize Heights

Avg ~ 5’ 9”

Job Description
Data Analyst in retail, 5 yrs. Specializes in consumer behavior & predictive analytics.
ML Engineer in fintech, 3 yrs. Develops algorithms for stock market predictions.
Data Scientist in healthcare, 10+ yrs. Uses patient data for better care outcomes.
Database Admin for e-commerce, 7 yrs. Focuses on efficient, secure data storage.
BI Analyst in hospitality, early career. Turns data into insights for growth.
Data Architect, 15 yrs, builds big data systems and data lakes for startups.
Freelance Data Visualization Specialist, 4 yrs. Makes complex data engaging.
Senior Data Engineer in automotive, 8 yrs. Works on data pipelines for autonomous vehicles.

Summarize roles

🤷

Text Embeddings

Translate words into numbers

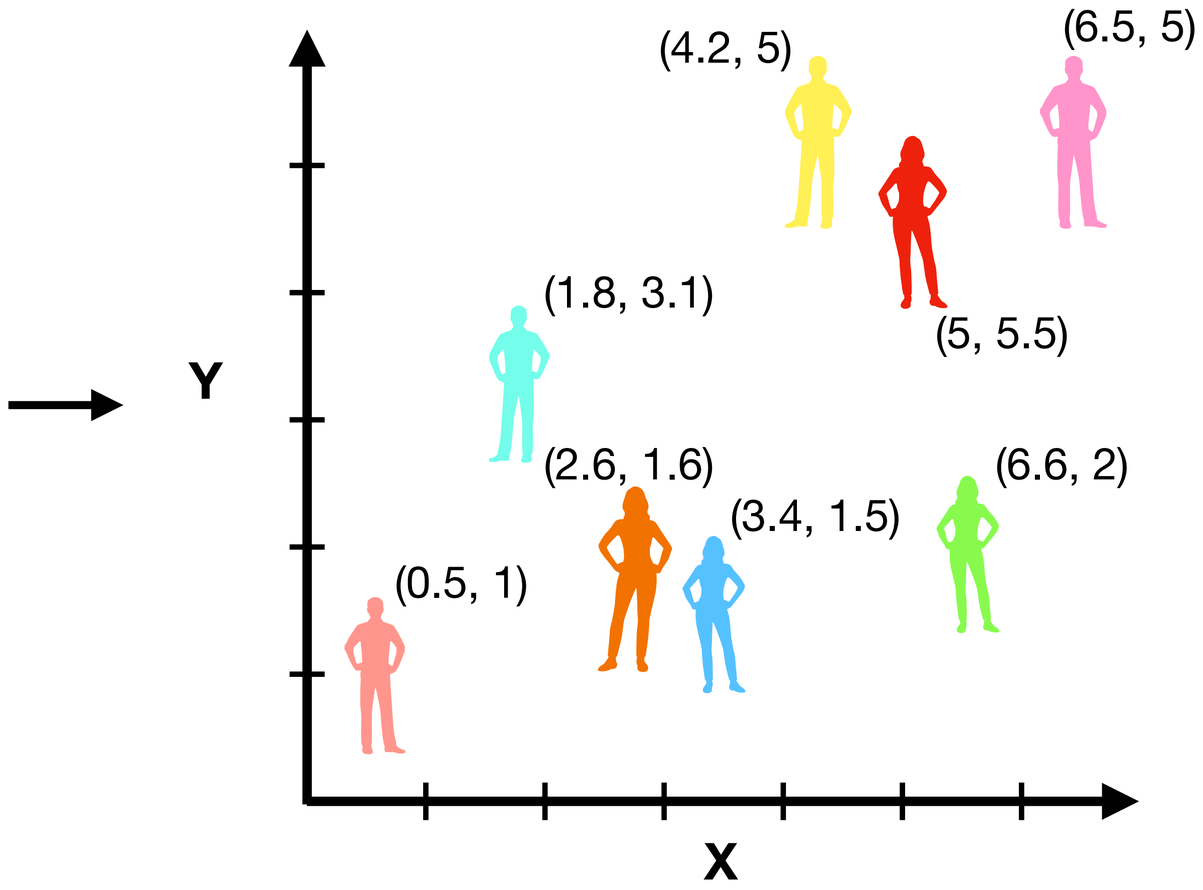
Job Description		Text Embedding
Data Analyst in retail, 5 yrs	→	(3.4, 1.5)
ML Engineer in fintech, 3 yrs	→	(1.8, 3.1)
Data Scientist in healthcare, 10+ yrs.	→	(6.6, 2)
Database Admin for e-commerce, 7 yrs.	→	(4.2, 5)
BI Analyst in hospitality, early career.	→	(0.5, 1)
Data Architect, 15 yrs.	→	(6.5, 5)
Freelance Data Visualization Specialist, 4 yrs.	→	(2.6, 1.6)
Senior Data Engineer in automotive, 8 yrs.	→	(5, 5.5)

[1] [2]

Text Embeddings

Translate words into ***meaningful*** numbers

Job Description
Data Analyst in retail, 5 yrs
ML Engineer in fintech, 3 yrs
Data Scientist in healthcare, 10+ yrs.
Database Admin for e-commerce, 7 yrs.
BI Analyst in hospitality, early career.
Data Architect, 15 yrs.
Freelance Data Visualization Specialist, 4 yrs.
Senior Data Engineer in automotive, 8 yrs.

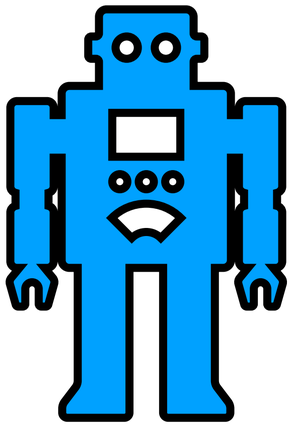


 [1]

Why should I care?

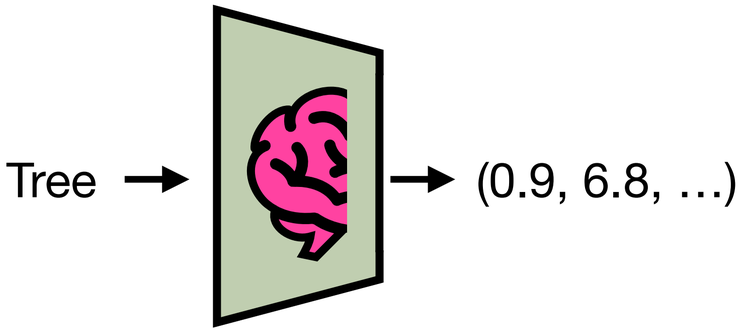
Can’t I just use ChatGPT?

AI Assistant



- Early days
- Major computational cost
- LLM security risks (see [3])
- Unpredictable responses (e.g. hallucinations)

Text Embedding

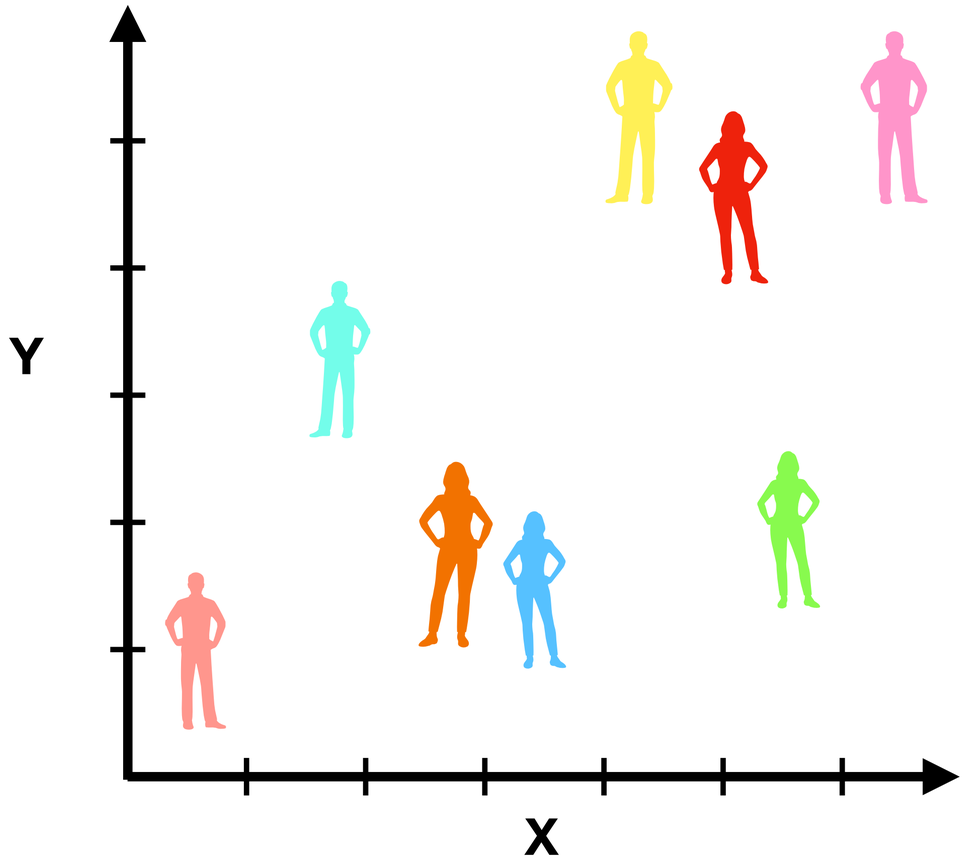


- Been around for decades
- (Much) lower computational cost
- (Far) fewer security risks
- Predictable responses

[2] [3]

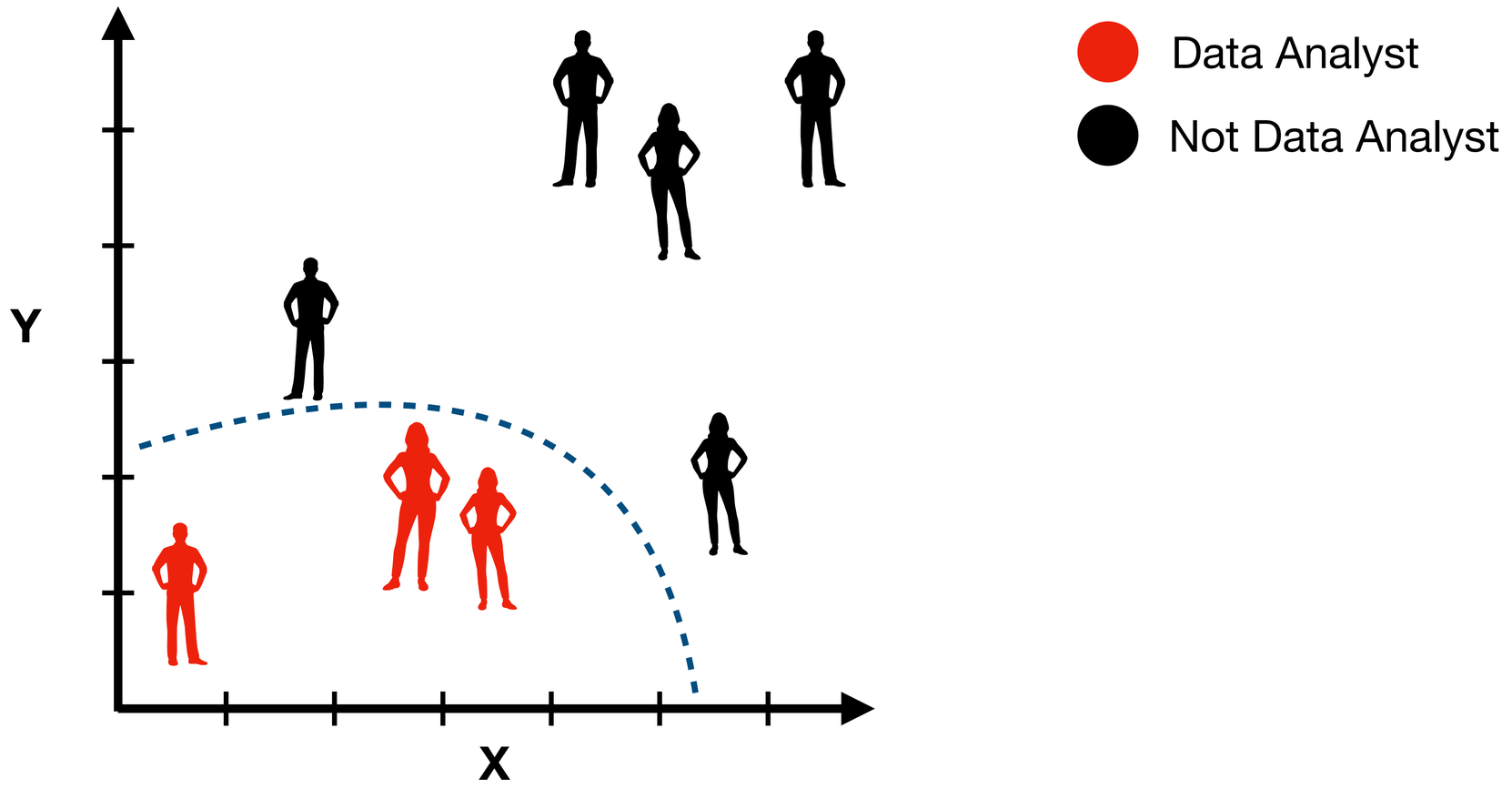
Use Case 1: Text Classification

Text Classification = assigning a label to a piece of text



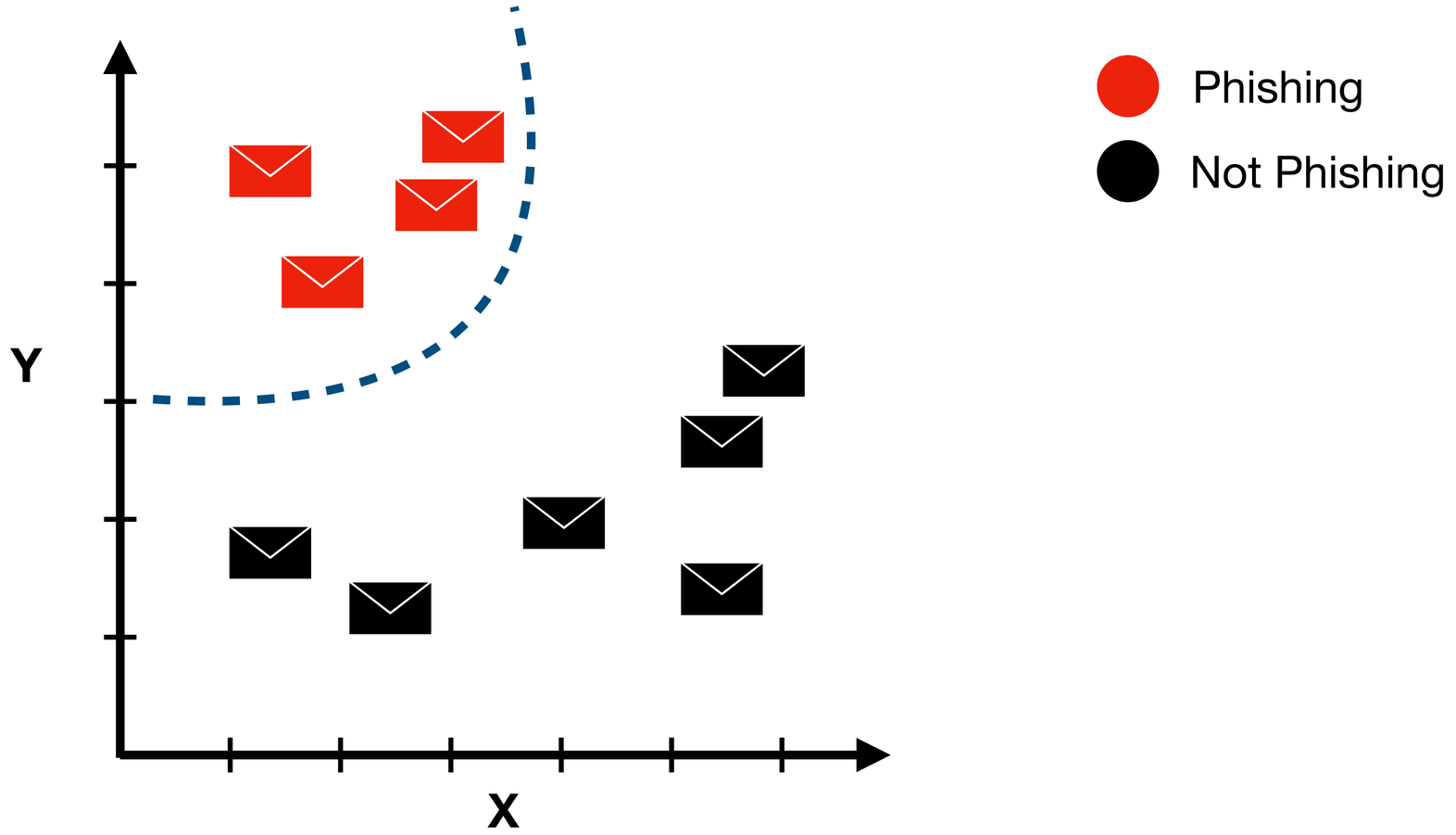
Use Case 1: Text Classification

Text Classification = assigning a label to a piece of text



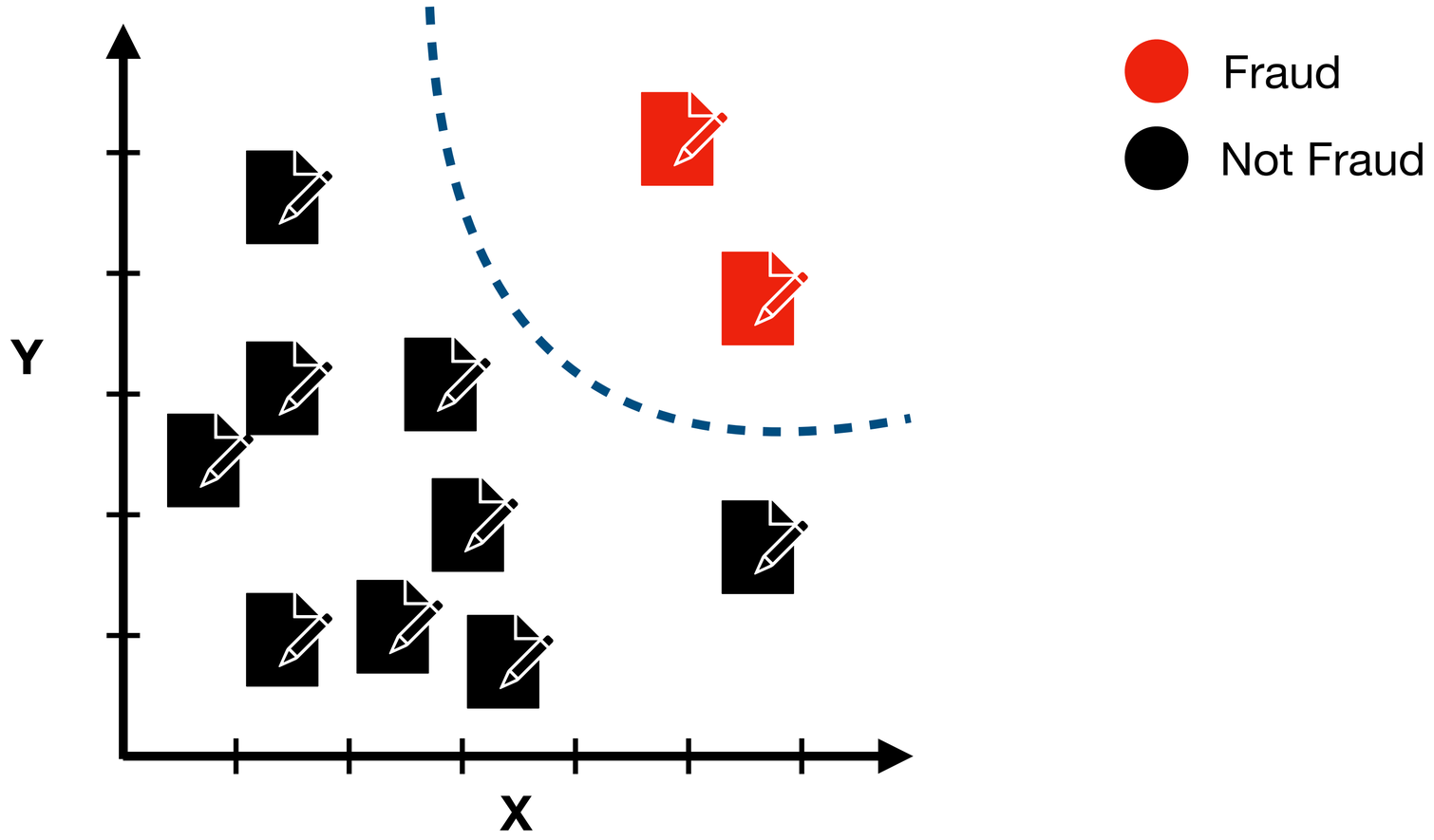
Use Case 1: Text Classification

Text Classification = assigning a label to a piece of text



Use Case 1: Text Classification

Text Classification = assigning a label to a piece of text



Use Case 1: Text Classification

Imports

```
import openai
from sk import my_sk

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import roc_auc_score
```

Read Data

```
df_resume = pd.read_csv('resumes/resumes_train.csv')
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs/text-embeddings

Use Case 1: Text Classification

Generate Embeddings

```
def generate_embeddings(text, my_sk):
    # set credentials
    client = openai.OpenAI(api_key = my_sk)

    # make api call
    response = client.embeddings.create(
        input=text,
        model="text-embedding-3-small"
    )

    # return text embedding
    return response.data
```

```
# generate embeddings
text_embeddings = generate_embeddings(df_resume['resume'], my_sk)

# extract embeddings
text_embedding_list =
  [text_embeddings[i].embedding for i in range(len(text_embeddings))]
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs/text-embeddings

Use Case 1: Text Classification

Store Embeddings in Dataframe

```
# define df column names
column_names =
  ["embedding_" + str(i) for i in range(len(text_embedding_list[0]))]

# store text embeddings in dataframe
df_train = pd.DataFrame(text_embedding_list, columns=column_names)

# create target variable
df_train['is_data_scientist'] = df_resume['role']=="Data Scientist"
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs/text-embeddings

Use Case 1: Text Classification

Model Training

```
# split variables by predictors and target
X = df_train.iloc[:,:-1]
y = df_train.iloc[:,-1]

# train rf model
clf = RandomForestClassifier(max_depth=2, random_state=0).fit(X, y)
```

Evaluation

```
# model accuracy for training data
print(clf.score(X,y))

# AUC value for training data
print(roc_auc_score(y, clf.predict_proba(X)[:,1]))
```

```
# output
# 1
# 1
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs/text-embeddings

Use Case 1: Text Classification

Load Testing Data

```
# import testing data
df_resume = pd.read_csv('resumes/resumes_test.csv')

# generate embeddings
text_embedding_list = generate_embeddings(df_resume['resume'], my_sk)
text_embedding_list =
  [text_embedding_list[i].embedding for i in range(len(text_embedding_list))]

# store text embeddings in dataframe
df_test = pd.DataFrame(text_embedding_list, columns=column_names)

# create target variable
df_test['is_data_scientist'] = df_resume['role']=="Data Scientist"

# define predictors and target
X_test = df_test.iloc[:,:-1]
y_test = df_test.iloc[:,-1]
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs/text-embeddings

Use Case 1: Text Classification

Evaluate Model on Testing Data

```
# model accuracy for testing data
print(clf.score(X_test,y_test))

# AUC value for testing data
print(roc_auc_score(y_test, clf.predict_proba(X_test)[:,1]))

# output
# 0.98
# 0.9983333333333333
```

Notes on overfitting:

1) 1537 predictors, only 100 records

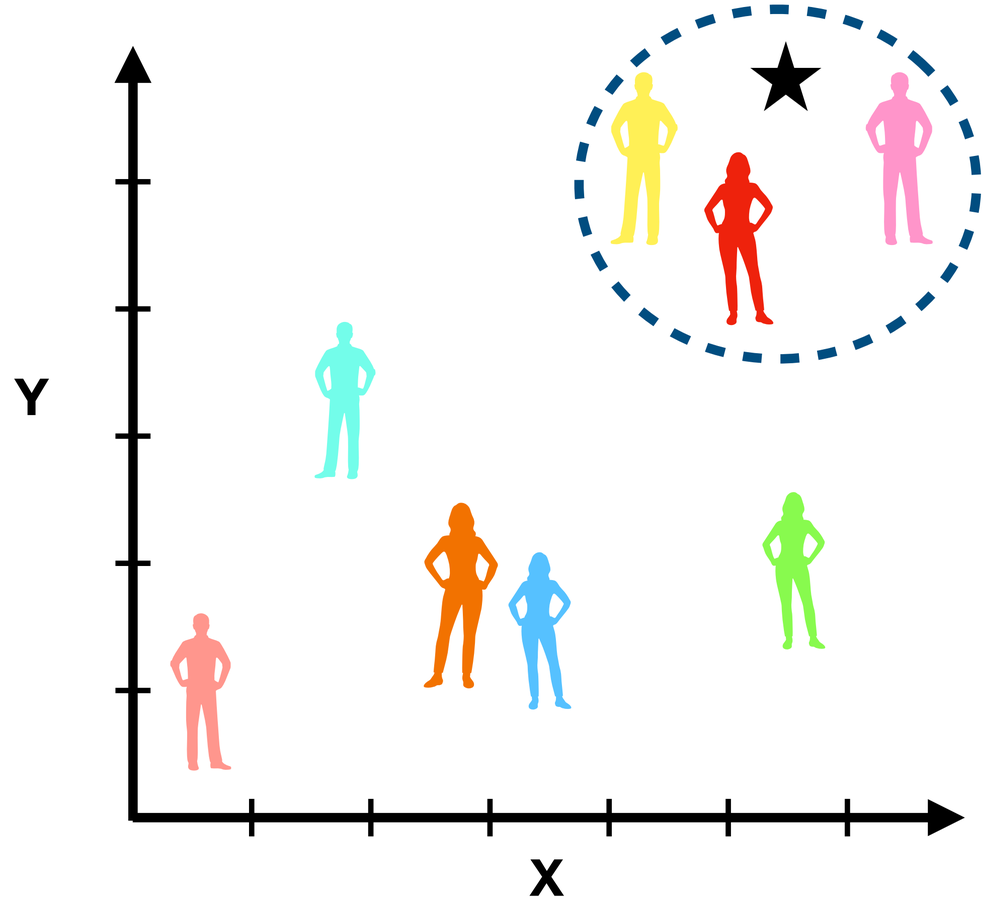
2) Training and testing were generated in identical ways


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs/text-embeddings

Use Case 2: Semantic Search

Semantic Search = return results based on the *meaning* of user's query

"I need someone to build my data infrastructure"



★ Query

Keyword search may return suboptimal results

Use Case 2: Semantic Search

Imports

```
import numpy as np
import pandas as pd

from sentence_transformers import SentenceTransformer

from sklearn.decomposition import PCA
from sklearn.metrics import DistanceMetric

import matplotlib.pyplot as plt
import matplotlib as mpl
```

Read Data

```
df_resume = pd.read_csv('resumes/resumes_train.csv')

# relabel random role as "other"
df_resume['role'][df_resume['role'].iloc[-1] == df_resume['role']] = "Other"
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs/text-embeddings

Use Case 2: Semantic Search

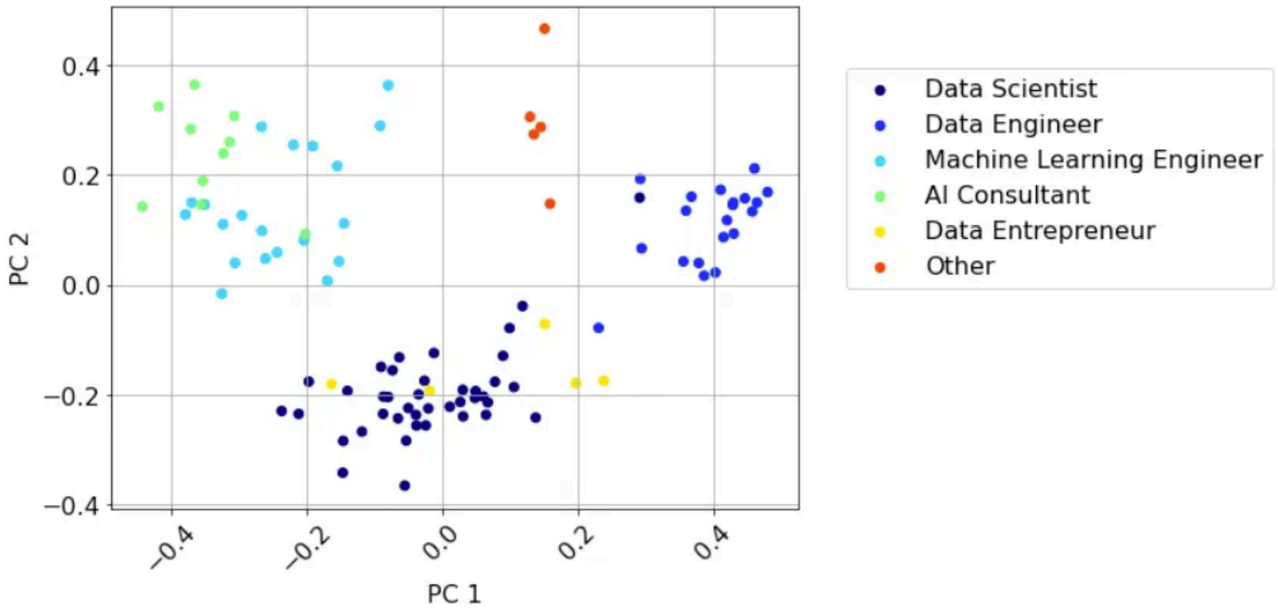
Generate Embeddings

```
# import pre-trained model (full list: https://www.sbert.net/docs/pretrained_mod
model = SentenceTransformer("all-MiniLM-L6-v2")

# encode text
embedding_arr = model.encode(df_resume['resume'])
```

Reduce dimensionality from 384 to 2 with PCA

(code on GitHub)



https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs/text-embeddings

Use Case 2: Semantic Search

Define and Encode Query

```
# define query
query = "I need someone to build out my data infrastructure"

# encode query
query_embedding = model.encode(query)
```

Compute Nearest Neighbors

```
# define distance metric (other options: manhattan, chebyshev)
dist = DistanceMetric.get_metric('euclidean')

# compute pair-wise distances between query embedding and resume embeddings
dist_arr =
  dist.pairwise(embedding_arr, query_embedding.reshape(1, -1)).flatten()
# sort results
idist_arr_sorted = np.argsort(dist_arr)
```


https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs/text-embeddings

Use Case 2: Semantic Search

Print roles of top 10 results

```
# print roles of top 10 closest resumes to query in embedding space
print(df_resume['role'].iloc[idist_arr_sorted[:10]])
```

```
48         Data Engineer
58         Data Engineer
43         Data Engineer
93    Data Entrepreneur
41         Data Engineer
55         Data Engineer
40         Data Engineer
56         Data Engineer
46         Data Engineer
47         Data Engineer
Name: role, dtype: object
```

Look at resume of top result

John Doe

Summary: Highly skilled and experienced Data Engineer with a strong background in designing, implementing, and maintaining data pipelines. Proficient in data modeling, ETL processes, and data warehousing. Adept at working with large datasets and optimizing data workflows to improve efficiency.

Professional Experience:

- **Senior Data Engineer**
  XYZ Tech, Anytown, USA
  June 2018 - Present
  - Designed and developed scalable data pipelines to handle terabytes of data daily.
  - Optimized ETL processes to improve data quality and processing time by 30%.
  - Collaborated with cross-functional teams to implement data architecture best practices.
- **Data Engineer**
  ABC Solutions, Sometown, USA
  January 2015 - May 2018
  - Built and maintained data pipelines for real-time data processing.
  - Developed data models and implemented data governance policies.
  - Worked on data integration projects to streamline data access for business users.

Education:

- **Master of Science in Computer Science**
  University of Technology, Cityville, USA
  Graduated: 2014
- **Bachelor of Science in Computer Engineering**
  State College, Hometown, USA
  Graduated: 2012

Technical Skills:

- Programming: Python, SQL, Java
- Big Data Technologies: Hadoop, Spark, Kafka
- Databases: MySQL, PostgreSQL, MongoDB
- Data Warehousing: Amazon Redshift, Snowflake
- ETL Tools: Apache NiFi, Talend
- Data Visualization: Tableau, Power BI

Certifications:

- Certified Data Management Professional (CDMP)
- AWS Certified Big Data - Specialty

---

https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs/text-embeddings

Use Case 2: Semantic Search

Visualizing Queries in Embedding Space

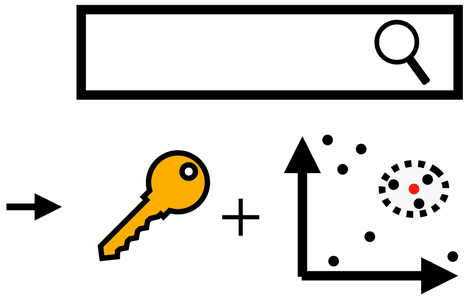
https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs/text-embeddings

Use Case 2: Semantic Search

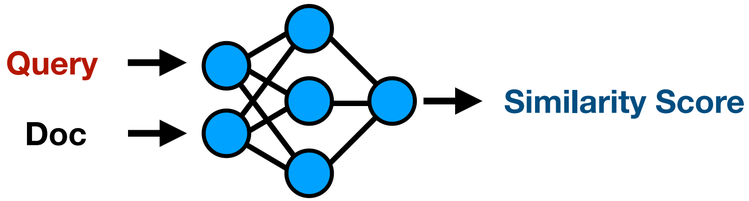
"Data Engineer with Apache Airflow experience"

Only 1 of the top 5 results (3rd) have Airflow experience

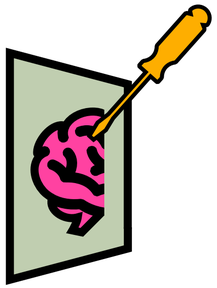
Improving Search



Hybrid Search



Reranker



Fine-tune Embedding

https://github.com/ShawhinT/YouTube-Blog/tree/main/LLMs/text-embeddings

Text Embeddings, Classification, and Semantic Search

An introduction with example Python code

Shaw Talebi

Published in Towards Data Science · 11 min read · 6 hours ago

References

[1] https://youtu.be/A8HEPBdKVMA?si=PA4kCnfgd3nx24LR

[2] R. Patil, S. Boit, V. Gudivada and J. Nandigam, “A Survey of Text Representation an
Embedding Techniques in NLP,” in IEEE Access, vol. 11, pp. 36120–36146, 2023, doi:
10.1109/ACCESS.2023.3266377.

[3] https://owasp.org/www-project-top-10-for-large-language-model-applications/