



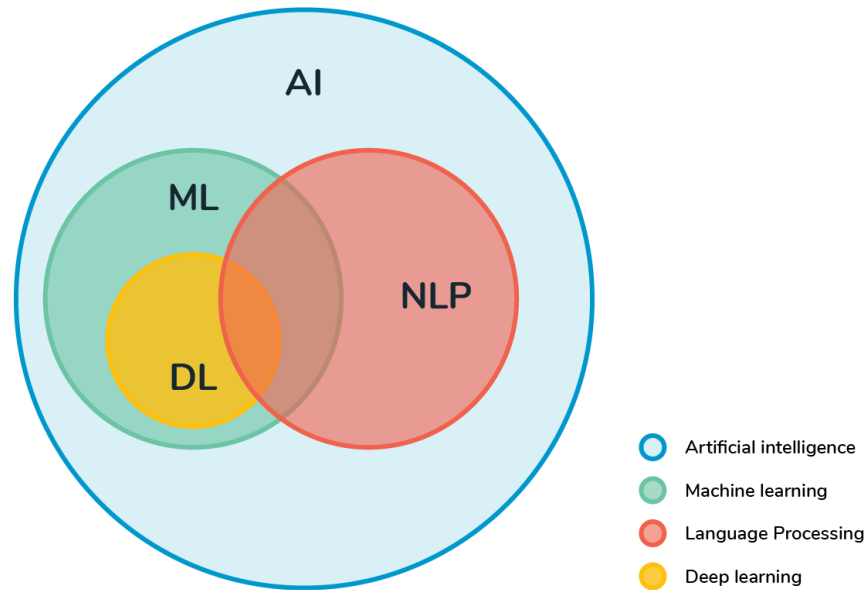
# Language Detection AI

---

CKRAFT-BOT



# AI – NLP



- Spam detection
- Fake new detection
- Language detection
- Chat bots
- Smart Assistance
- Sentiment analysis
- Text summarization
- Q&A
- Translation
- Transcription
- Spell check

# Dataset

- Kaggle
- 22,000 rows
  - 22 Languages
  - 1000 texts/ lang

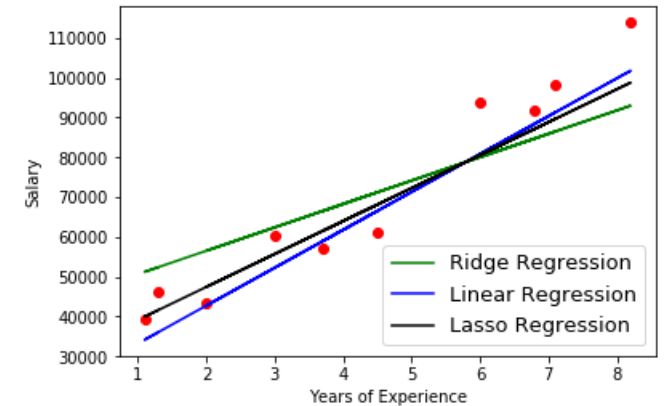
# Project

- Import in dataset
- Quick review of dataset
  - Clean
- Training model
- Model eval (accuracy test)
- Observations

# Training Process

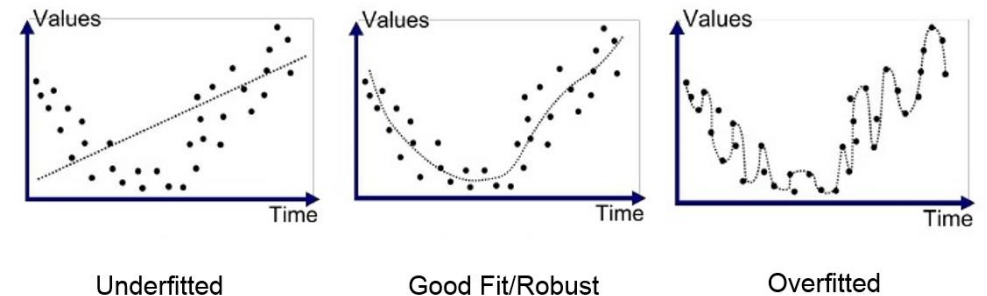
- Algs
  - Training model: Ridge Classifier
  - Model eval: (Multinomial) Naive Bayes classifier
- Model setup
  - Vectorization
    - Parsing
    - Convert text to numeric
    - Convert to matrix
      - Row X column
      - Language X unique vocabs (no dupes)
  - Matrixes → baseline for predication

# Ridge Classifier



- Like linear regression
  - Best fit line
  - Strength of var relationships
    - IV  $\rightarrow$  DV
  - Coefficients
    - Twinkle twinkle little stars
    - P Values
      - $< 0.001$  near perfect, high confidence
      - $< 0.05$  stat sig
      - $> 0.05$  not stat sig, null hyp

- But not linear regression
  - Mitigates under/overfitting by modifying weights through L2 regularization application
    - Takes the weights to near zero while not taking it to zero



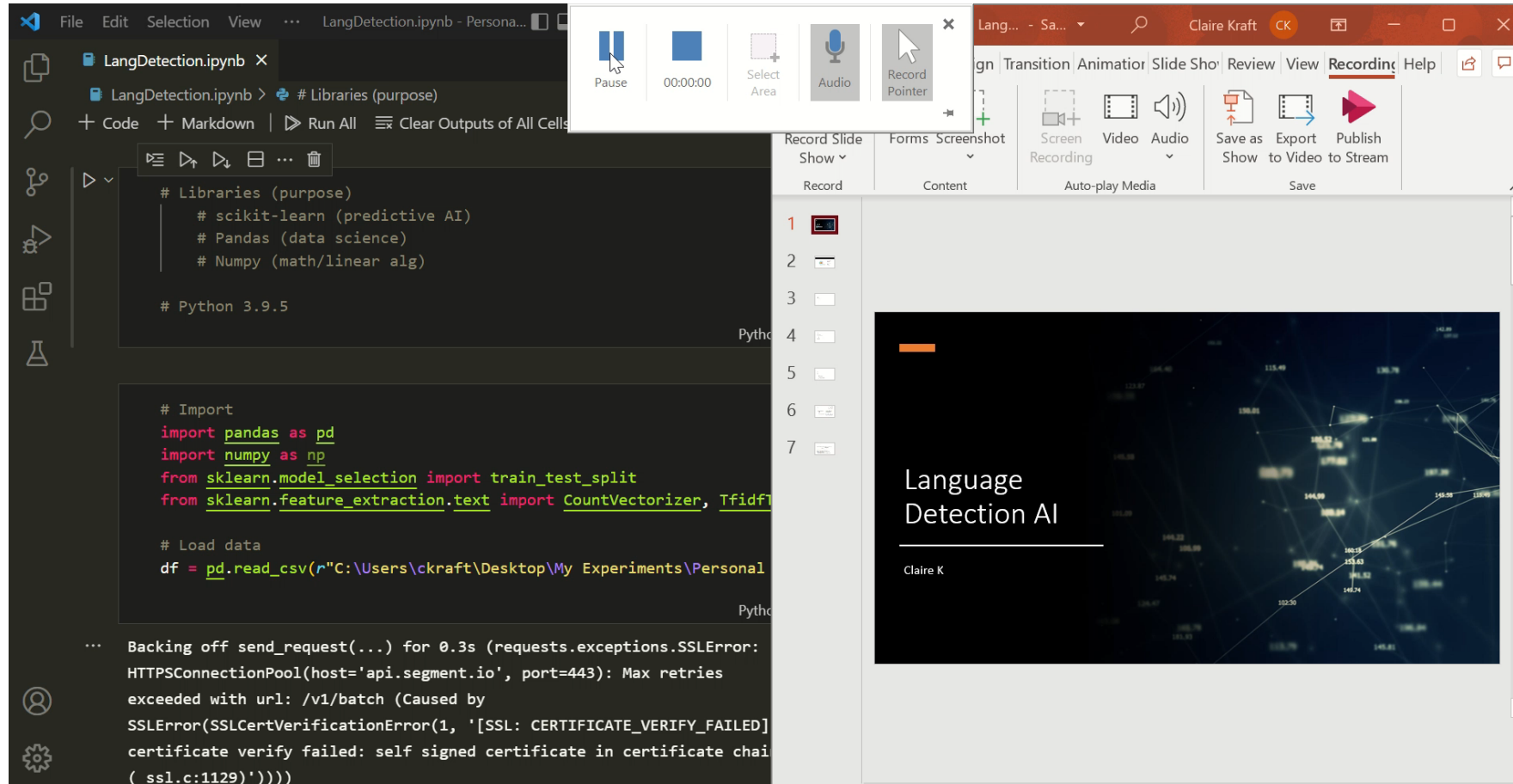
# Model Eval

- (Multinomial) Naive Bayes classifier (algorithm) is based off the Bayes Theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- Naive Bayes classifier is based off the Bayes Theorem
- Pros of using this classifier alg: efficiency and ease
- Assumption: every value (feature) is independent or “unique”
- Calculates conditional probability
  - An event occurring given presumptions or evidence that have already occurred
  - Simply put likelihood of matching existing convention otherwise “naïve”

# Monologue for Vanessa



The image shows a screen recording of a Jupyter Notebook and a presentation slide. The Jupyter Notebook, titled "LangDetection.ipynb", is open in a web browser. It contains two code cells. The first cell is a comment block listing libraries and their purposes: "# Libraries (purpose)", "# scikit-learn (predictive AI)", "# Pandas (data science)", "# Numpy (math/linear alg)", and "# Python 3.9.5". The second cell contains Python code for importing libraries and loading data: "# Import", "import pandas as pd", "import numpy as np", "from sklearn.model\_selection import train\_test\_split", "from sklearn.feature\_extraction.text import CountVectorizer, TfidfVectorizer", "# Load data", and "df = pd.read\_csv(r\"C:\Users\ckraft\Desktop\My Experiments\Personal...\"". The third cell shows an error message: "Backing off send\_request(...) for 0.3s (requests.exceptions.SSLError: HTTPSConnectionPool(host='api.segment.io', port=443): Max retries exceeded with url: /v1/batch (Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE\_VERIFY\_FAILED] certificate verify failed: self signed certificate in certificate chain (\_ssl.c:1129)))))".

Overlaid on the Jupyter Notebook is a presentation slide titled "Language Detection AI" by Claire K. The slide features a dark background with a network graph visualization. The graph consists of numerous nodes, each labeled with a numerical value, connected by lines. The nodes are distributed across the slide, with some clusters and some isolated nodes. The title "Language Detection AI" is prominently displayed in the upper left, and the author's name "Claire K" is in the lower left.

The screen recording interface includes a toolbar at the top with icons for Pause, Select Area, Audio, and Record Pointer. The recording status bar at the bottom shows the recording is in progress, with a timer at 00:00:00. The recording settings are visible on the right, showing the recording is in progress, with a timer at 00:00:00. The recording settings are visible on the right, showing the recording is in progress, with a timer at 00:00:00.