Resume Classification

Using Machine Learning

Project by:

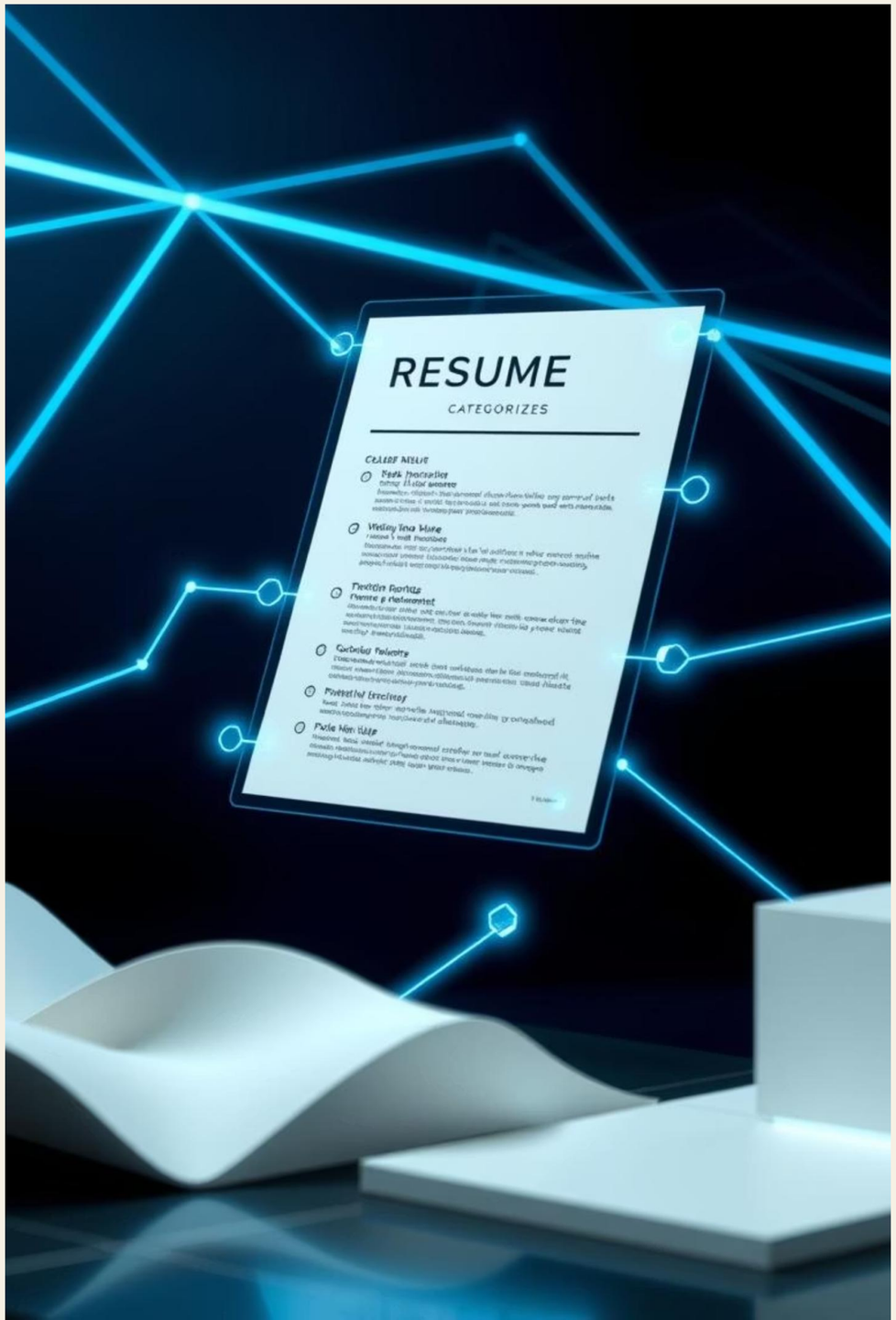
Farah Fatima Azmeali Rashid

Vinisha Sahoo

Shebaz Sheru Shaikh

Ghanshyam Kamlakr Patil

Alwyna William Chandanshiv



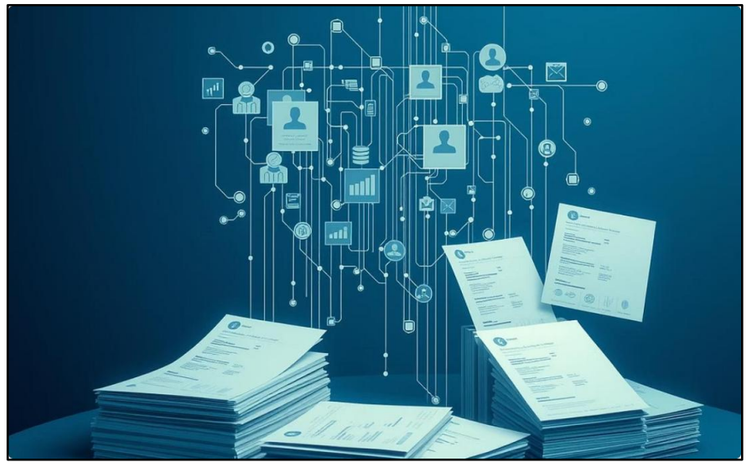
Introduction to Resume Classification

The Challenge:

*Recruiters spend an average of **23 hours** screening resumes for a single hire.*

Manual resume screening is time-consuming and prone to human bias, impacting hiring efficiency and candidate experience.

Our Solution:



Automated Categorization

Leveraging machine learning to instantly classify resumes into predefined job roles, saving valuable time.

Enhanced Efficiency

Enabling recruiters to focus on qualified candidates, reducing time-to-hire and improving overall productivity.

Leveraging Machine Learning

Utilizes advanced ML models for accurate and scalable resume screening , minimizing human error.

Dataset Overview

*Our prototype was developed using a carefully curated dataset comprising **79 resumes** to train and validate our classification models.*

*These resumes are distributed across **four distinct job categories**, reflecting common roles in the tech industry:*

- *React Developer*
- *SQL Developer*
- *PeopleSoft Consultant*
- *Workday Consultant*

Exploratory Data Analysis (EDA)

Class Distribution

We analyzed the distribution of resumes across the four categories to understand dataset balance.

Keyword Frequency

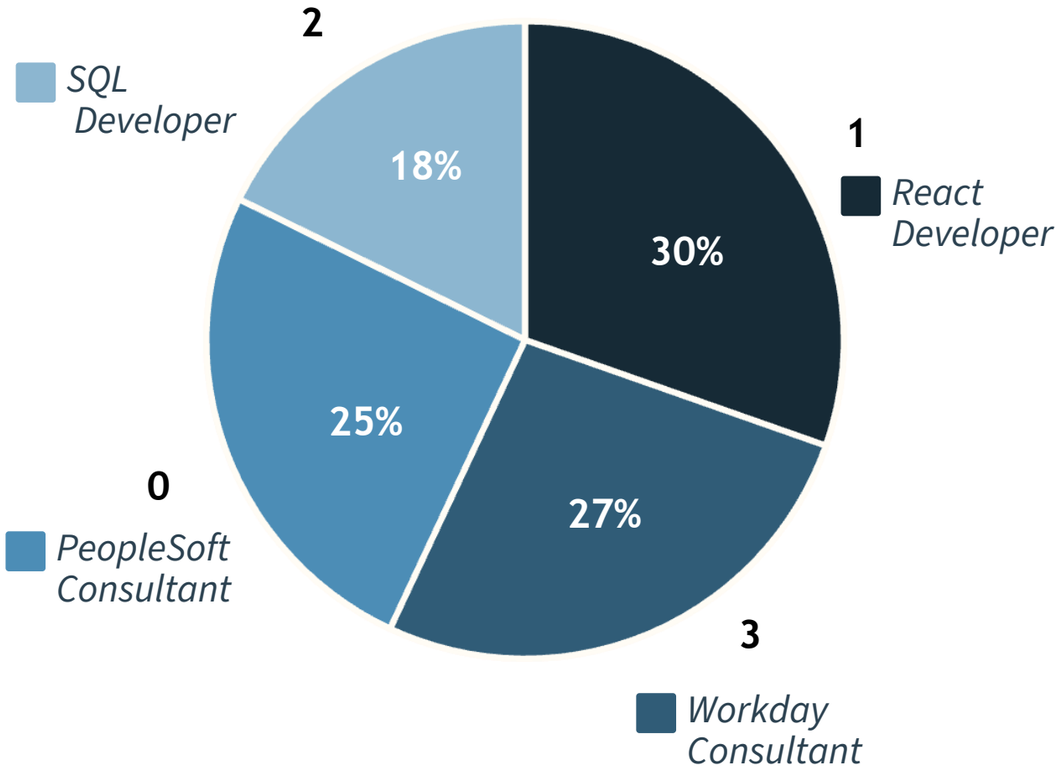
Examined the most frequent keywords associated with each job role, revealing distinctive patterns.

Clear Distinctions

Our analysis observed clear distinctions between classes based on these keywords, indicating strong signal for classification.

The EDA phase was crucial for understanding the inherent structure and characteristics of our resume dataset, guiding subsequent feature engineering and model selection.

Class Distribution: A Balanced View



The target variable 'role' is a multiclass feature comprising four distinct classes. The distribution is not significantly imbalanced; the largest class contains 24 samples, which is only 1.7 times the smallest class with 14 samples.

In most NLP classification tasks, such a minor imbalance is considered acceptable and typically does not require oversampling or class balancing techniques, ensuring our model learns effectively from all categories.

Feature Engineering: Crafting Custom Resume Metrics

To enhance our model's predictive power, we engineered custom features directly from the raw resume content. This involved extracting quantitative metrics that reflect the structural characteristics of each resume:

Character Count

The total number of characters in a resume, providing a measure of its overall length and verbosity.

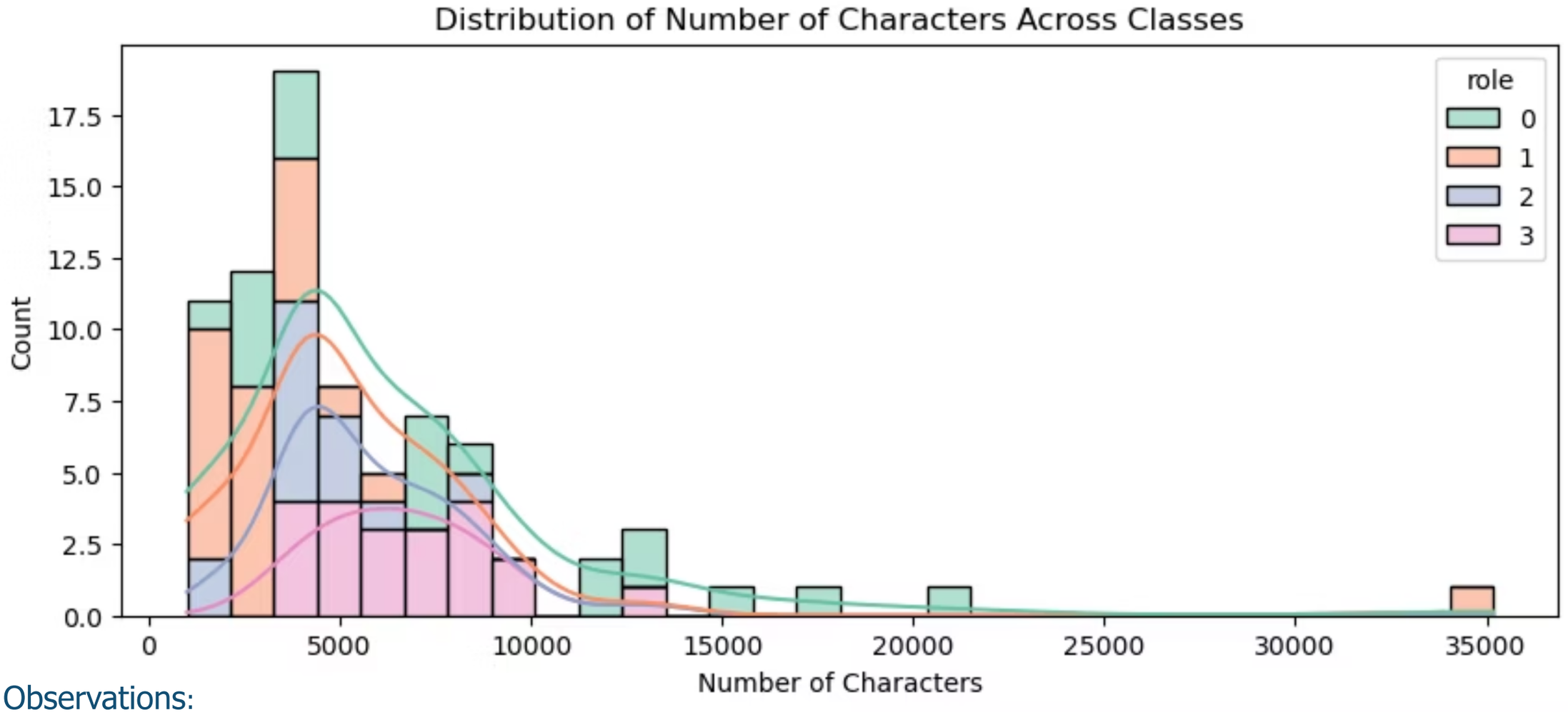
Word Count

The total number of words, offering insights into content density and detail.

Sentence Count

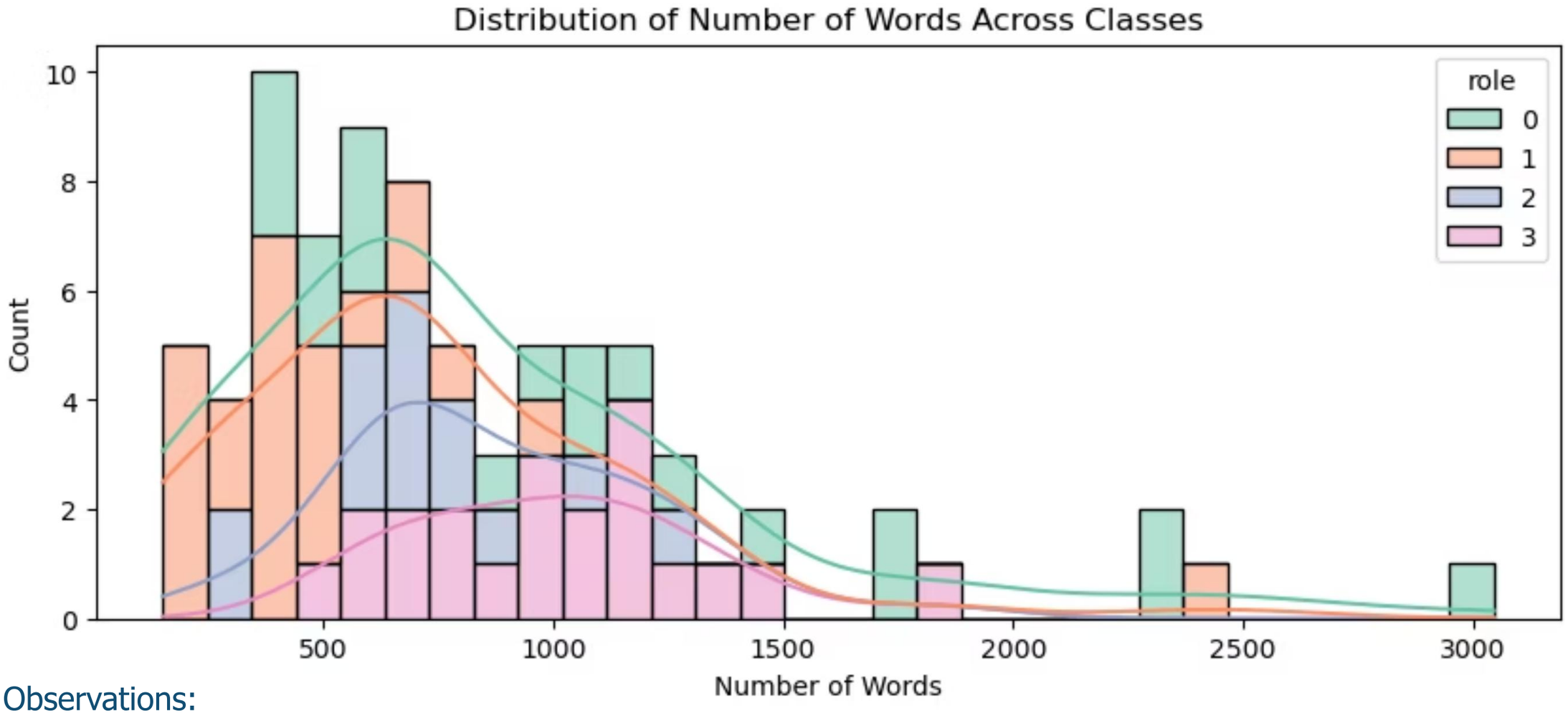
The number of sentences, indicating narrative structure and flow.

These engineered features allow our machine learning models to leverage beyond keyword-based information, capturing nuanced differences between resume types.



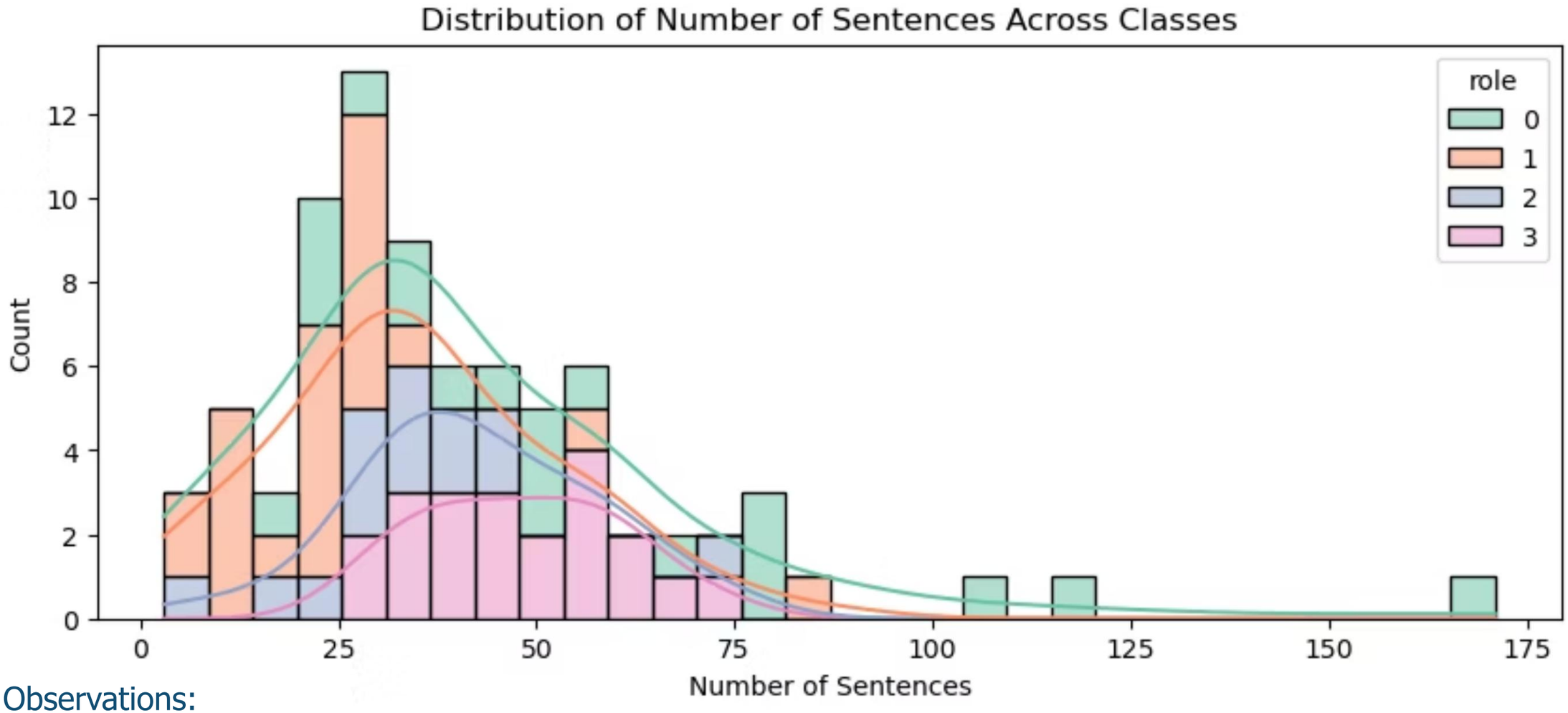
Key Observations:

- **Most Resumes:** Cluster between 3,000 and 8,000 characters.
- **React Developers (1):** Generally shorter, around 3,000–4,000 characters.
- **PeopleSoft Consultants (0):** Exhibit the widest range, with many resumes exceeding 10,000 characters, suggesting more extensive detailed skill sets.
- **SQL Developers (2):** Typically fall within the 3,000–6,000 character range, with fewer exceptionally long documents.
- **Workday Consultants (3):** Tend to be slightly longer, averaging 5,000–8,000 characters.
- **Outliers:** A few resumes across all categories extend beyond 15,000 characters.



Key Observations:

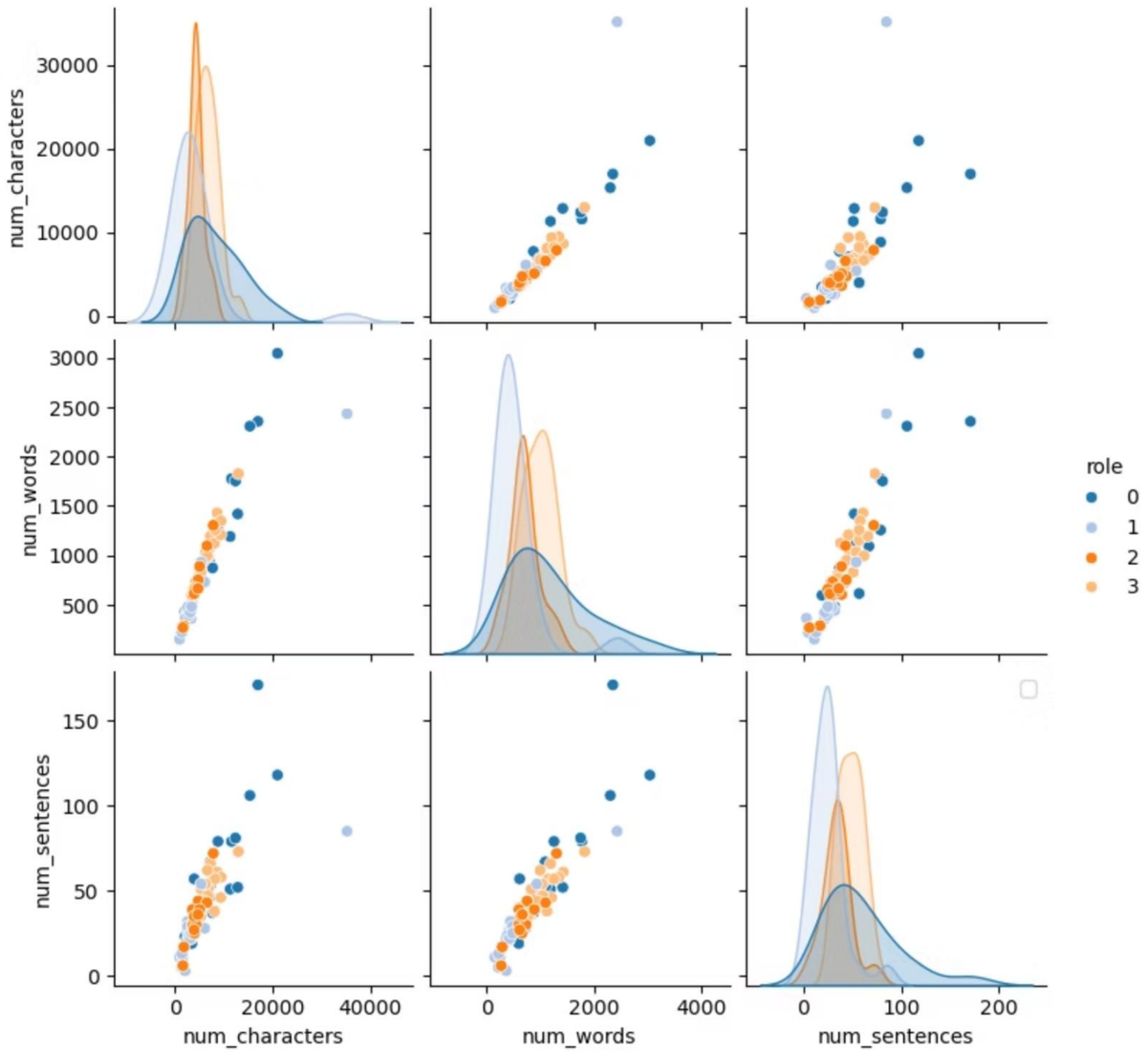
- **Most Resumes:** Fall within the 500–1200 word range.
- **React Developers (1):** Exhibit shorter lengths, typically around 500–800 words.
- **PeopleSoft Consultants (0):** Show a broad range, with many exceeding 1500 words, indicating more detailed content.
- **SQL Developers (2):** Primarily concentrated between 500–900 words.
- **Workday Consultants (3):** Tend to be slightly longer, generally ranging from 800–1200 words.
- **Outliers:** A small number of resumes extend beyond 2000 words.



Key Observations:

- **Most Resumes:** Contain between 20–60 sentences.
- **React Developers (1):** Appear denser in the 20–40 sentence range, suggesting concise bullet points.
- **PeopleSoft Consultants (0):** Display a wider spread, with many exceeding 60 sentences, highlighting comprehensive experience descriptions.
- **SQL Developers (2):** Mostly fall between 25–50 sentences.
- **Workday Consultants (3):** Are slightly longer, typically ranging from 30–60 sentences.
- **Outliers:** Resumes with over 100 sentences are primarily from PeopleSoft Consultants, reinforcing their detailed nature.

Inter-Feature Correlations: Pairplot of Custom Variables



Key Insights:

Strong Positive Correlation: A clear linear relationship is observed between characters, words, and sentences, indicating that resumes longer in one metric are consistently longer in others.

PeopleSoft (0) Uniqueness: These resumes show greater spread and higher outliers across all metrics, reinforcing their detailed nature.

React (1) & SQL Dev (2) Consistency: These categories are tightly clustered, suggesting more uniform resume lengths.

Workday (3) Moderate Spread: These resumes show a moderate distribution but without the extreme outliers seen in PeopleSoft.

This strong positive correlation confirms that these custom features collectively capture a consistent dimension of resume verbosity, which can be valuable for classification.

Inter-Feature Correlations: Heatmap of Custom Variables

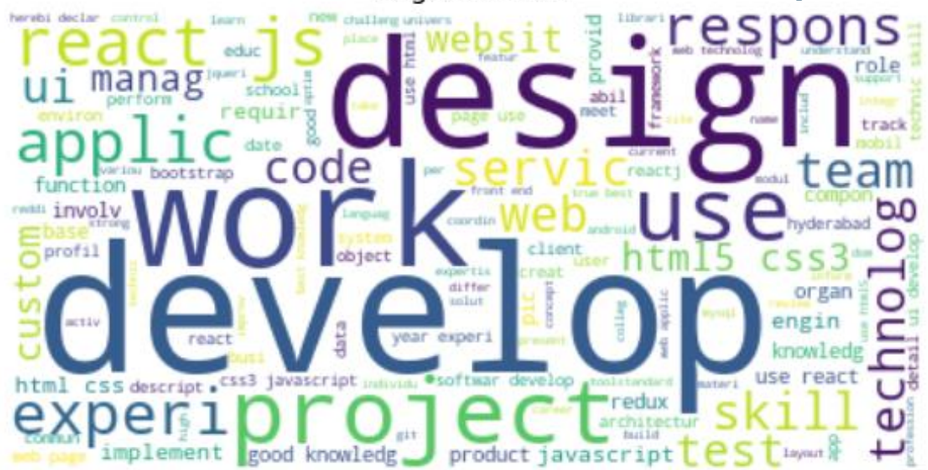
Observation:

The custom features num_characters, num_words, and num_sentences show high correlation with each other (above 0.9), meaning they carry similar information. While this multicollinearity can affect linear models, it's usually not a major issue in classification tasks, especially when strong features like TF-IDF or Bag of features have a very weak correlation (less than 0.1) with the target variable role. So, on their own, they are not likely to improve model accuracy much, but they may still be helpful as additional features in ensemble models or for basic data analysis and readability insights.

Word Cloud: Most Frequent Terms

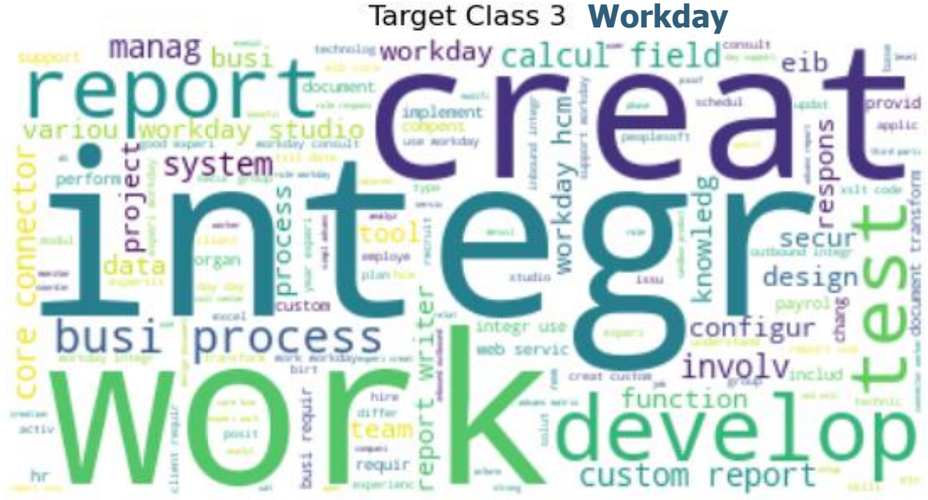
Target Class 0 **Peoplesoft**

Target Class 1 **React Developer**



Target Class 2 **SQL Developer**

Target Class 3 **Workday**



Preprocessing and Transformation for Optimal Accuracy

Effective preprocessing—turning raw resume text into a machine-readable form—drives better model results.

Text Cleaning

Lowercasing, removing punctuation, numbers, and special characters to standardize the text.

Tokenization & Stopword Removal

Breaking text into individual words (tokens) and eliminating common, uninformative words using NLTK.

Vectorization

Converting text into numerical features using TF-IDF (Term Frequency-Inverse Document Frequency) and Bag of Words.

Data Splitting

Implementing train-test splits and stratified cross-validation to prepare the data for robust model training and evaluation.

This systematic approach ensures that our models learn from clean, relevant, and well-structured data.

Diverse Models, Focused Goal

We explored a spectrum of machine learning algorithms to identify the optimal performer for resume classification.

Linear Models

- ***Logistic Regression:*** *A strong baseline for classification.*
- ***LinearSVC:*** *Effective for high-dimensional data, often outperforming others on text.*
- ***SGDClassifier:*** *Efficient for large-scale datasets, adaptable to various loss functions.*

Probabilistic & Ensemble

- ***Multinomial Naive Bayes:*** *A classic for text classification, robust to noise.*
- ***RandomForest:*** *Ensemble method, good for feature importance and reducing overfitting.*
- ***XGBoost:*** *Powerful gradient boosting, known for high performance and speed.*

Each model was rigorously evaluated across key metrics: accuracy, precision, recall, and F1-score to ensure a holistic understanding of their capabilities.

Unpacking the Performance: A Closer Look at Results

LinearSVC + TF-IDF	***100%***	***1.000***	***1.0 ± 0.00***
Logistic Regression + TF-IDF	*98%*	*0.9875*	*0.98 ± 0.02*
Multinomial Naive Bayes + TF-IDF	*97%*	*0.9750*	*0.97 ± 0.03*
RandomForest + TF-IDF	*97%*	*0.9750*	*0.92 ± 0.03*
SGDClassifier+ TF-IDF	*98%*	*0.9875*	*0.98 ± 0.02*
XGBoost+ TF-IDF	*97%*	*0.9750*	*0.92 ± 0.03*

The Champion: LinearSVC with TF-IDF

*Our analysis clearly demonstrated **LinearSVC combined with TF-IDF vectorization** as the top-performing model, achieving a flawless 100% accuracy on both test and cross-validation sets. This exceptional consistency, evidenced by a 0.0 standard deviation in cross-validation, speaks to its stability and reliability in classifying resumes.*

Understanding 100% Accuracy: Beyond Overfitting

A perfect score often raises questions about overfitting. However, our rigorous methodology and data characteristics validate the model's performance on this specific dataset.

No Data Leakage

Strict separation of training and testing data ensured the model was evaluated on truly unseen resumes.

Unseen Resume Performance

Consistent high accuracy on fresh, unclassified resumes outside the training set confirmed generalization.

Stable Cross-Validation

A 0.0 standard deviation across validation folds indicates highly stable and reliable performance.

Highly Separable Classes

The chosen resume categories featured distinct and unique keyword sets, leading to clear classification boundaries.

⚠ **Note:** While impressive, the relatively small dataset size (**79 resumes**) suggests the need for future validation with a larger, more diverse corpus to confirm scalability.

Challenges & Optimizations

Challenges Faced

Input Handling Issues

Single text transformation function failed when processing multiple inputs (Series) during tuning.

Serialization Failures

Pickle file couldn’t be created because the custom preprocessing functions couldn’t be saved.

Prediction Discrepancy

Model trained on preprocessed data couldn't directly predict on raw text in the application.

Optimizations Made

Unified Input Handling

Updated `text_transform` to accept both single and multiple (Series) inputs, enhancing flexibility.

Integrated Preprocessing

Modified training to accept raw text preprocessing and vectorization directly into the pipeline for seamless operation.

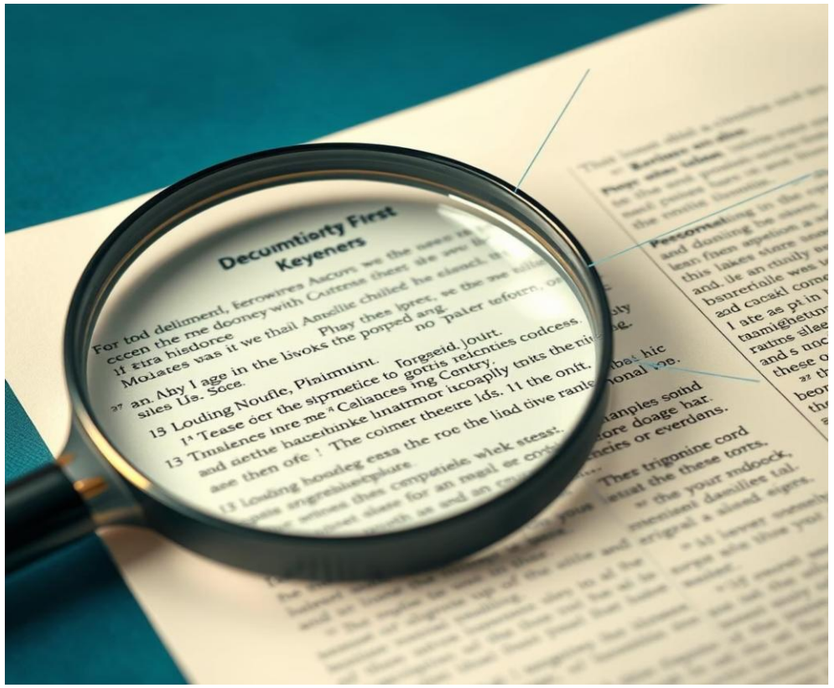
Streamlined Deployment

Created a .pkl file that processes raw resume content internally, enabling smooth, direct prediction.

Conclusion & Roadmap: The Path Forward

Key Achievement:

- **LinearSVC + TF-IDF:** Proven as the most effective approach for this dataset, demonstrating robust classification capabilities.
- **Feasibility Confirmed:** This project successfully validates the potential of machine learning to automate and enhance resume classification for recruiters.



Future Directions:

01

Expand Dataset Diversity

Incorporate a larger volume of resumes with more nuanced and potentially ambiguous skill sets.

02

Multi-Label Classification

Develop capabilities to handle resumes that might fit multiple job roles simultaneously.

03

Advanced NLP Techniques

Explore Word Embeddings and Transformer-based models (e.g., BERT) for richer semantic understanding.