PROJECT BETTER TALENT
USING DEEP LEARNING AND DJANGO FRAMEWORK

PROJECT REPORT

SUBMITTED TO

SVKM'S NMIMS (Deemed to be) UNIVERSITY

IN PARTIAL FULFILLMENT FOR THE DEGREE OF

MASTER OF SCIENCE

IN

STATISTICS & DATA SCIENCE

BY

ANIRUDDHA KASHINATH SHELKE

NMIMS NILKAMAL SCHOOL OF MATHEMATICS, APPLIED STATISTICS & ANALYTICS

V.L. Mehta Road, Vile-Parle (West), Mumbai – 400056

APRIL 2025

PROJECT BETTER TALENT
USING DEEP LEARNING AND DJANGO FRAMEWORK

PROJECT REPORT

SUBMITTED TO

SVKM'S NMIMS (Deemed to be) UNIVERSITY

IN PARTIAL FULFILLMENT FOR THE DEGREE OF

MASTER OF SCIENCE

IN

STATISTICS & DATA SCIENCE

BY

ANIRUDDHA KASHINATH SHELKE

SVKM'S NMIMS Deemed to be UNIVERSITY | NILKAMAL SCHOOL OF MATHEMATICS, APPLIED STATISTICS & ANALYTICS

Program Director (Statistics)	**Dean NSOMASA**
NSOMASA (Prof. Sunil S. Shirvaikar)	**Dr. Sushil Kulkarni**

NMIMS NILKAMAL SCHOOL OF MATHEMATICS, APPLIED STATISTICS & ANALYTICS

V.L. Mehta Road, Vile-Parle (West), Mumbai – 400056

APRIL 2025

CERTIFICATE

This is to certify that work described in this thesis entitled "PROJECT BETTER TALENT USING DEEP LEARNING AND DJANGO FRAMEWORK" has been carried out by (Aniruddha Kashinath Shelke) under my supervision. I certify that this is his bonafide work. The work described is original and has not been submitted for any degree to this or any other University.

Date: 4th April 2025
Place: Mumbai

Internal Mentor	**Company Mentor**
(Mrs. Shraddha Sarode)	**(Mr. Sundar Balasubramaniam)**
Date:	**Date:**

axis MY INDIA | People Empowerment Platform

631/632, Building no-6 3rd Floor, Solitaire Corporate Park

Andheri East, Mumbai, Maharashtra 400093

ACKNOWLEDGEMENT

I would like to extend my heartfelt gratitude to both NMIMS and Axis My India for providing me with the opportunity to work on this project. This journey has been incredibly enriching, allowing me to delve into the realms of Deep learning, Natural language processing (NLP), and database management.

I am especially thankful to Mrs. Shraddha Sarode for her invaluable guidance and support throughout this project. My sincere thanks to Mr. Sundar Balasubramaniam. for helping me acclimate to the new environment at Axis My India and introducing me to the fascinating field of workflow scheduling.

A special note of appreciation goes to my Manager, Mr. Sundar Balasubramaniam, whose unwavering support made this journey smoother. I am also grateful to my colleagues and seniors at both Axis My India and NMIMS for their constant encouragement and assistance.

Through this project, I have learned how to build a model using Deep learning and NLP techniques, manage databases, and deploy the model using Django to create a functional website.

Sincere Thanks,

Aniruddha Kashinath Shelke

INDEX

ABSTRACT

In modern recruitment, assessing a candidate’s problem-solving skills remains a critical aspect of the hiring process. Traditional hiring methods primarily rely on resumes and structured interviews, which often fail to provide an accurate representation of a candidate’s analytical and decision-making capabilities. To address these limitations, organizations implement a case study driven recruitment process, where candidates are required to analyze real world business problems and propose data driven solutions.
However, a significant challenge arises as candidates increasingly turn to AI generated responses, paraphrased content, or copied answers from experienced professionals. The growing accessibility of generative AI models such as ChatGPT and other large language models makes it difficult to determine whether a response truly reflects a candidate’s independent thought process. This leads to concerns regarding the integrity of candidate evaluations and the overall effectiveness of the hiring process.

To mitigate this issue, an AI Response Detection System is developed to classify candidate responses as human written or AI generated using Natural Language Processing (NLP), machine learning, and deep learning techniques. The system employs advanced text analysis methodologies, including linguistic feature extraction, BERT and GPT2 embeddings, K Means clustering, and an autoencoder based anomaly detection model to identify AI generated patterns in responses. By leveraging unsupervised learning, anomaly detection, and text representation models, the system enhances the accuracy of candidate assessments, ensuring that hiring decisions are based on genuine analytical capabilities rather than AI assisted responses.

This report provides an in-depth explanation of the feature extraction techniques, deep learning architectures, and clustering methodologies implemented in the system. It presents a comprehensive discussion on the working principles of BERT, GPT2, K Means clustering, and Autoencoders, highlighting their specific roles in detecting AI generated text. Additionally, the report discusses the rationale behind model selection and optimization strategies to improve detection accuracy.

INTRODUCTION

2.1 Problem Statement

In modern recruitment, evaluating candidates based on their problem-solving abilities, analytical acumen, and domain expertise is a fundamental objective. To enhance the rigor and effectiveness of candidate assessment, a case study-driven hiring framework is implemented. This approach requires candidates to visit a recruitment platform, select a specific job role, and engage with a carefully curated case study designed by subject matter experts. Candidates must then analyze the scenario and submit a structured response, showcasing their ability to reason critically, derive insights, and propose effective solutions. However, the implementation of this framework reveals a critical challenge. A growing number of candidates submit AI-generated responses using advanced language models such as ChatGPT, GPT-4, and other paraphrasing tools. Others leverage pre-existing solutions sourced from professional networks or external repositories, making it difficult to ascertain whether the response authentically represents the candidate's intellectual capabilities.

This challenge undermines the integrity of the recruitment process, as AI-generated content does not inherently reflect original thought, independent problem-solving skills, or domain-specific knowledge. Consequently, hiring managers face significant obstacles in differentiating between genuine, merit-based responses and those artificially enhanced through AI, leading to potential inaccuracies in candidate evaluation and selection.

2.2 Risks of AI-Generated Responses in Hiring

The infiltration of AI-generated content in recruitment poses multiple strategic and operational risks, each of which compromises the efficacy, fairness, and reliability of the hiring process:

1. Skill Mismatch and Performance Disparities: Candidates who submit AI-generated responses may create a false perception of competency, leading recruiters to believe they possess the requisite skills. However, once hired, these individuals may struggle to perform in real-world scenarios, resulting in discrepancies between perceived potential and actual job performance.
2. Erosion of Fairness in Candidate Evaluation: The use of AI-assisted responses introduces an unlevel playing field, where candidates who rely on generative AI gain an undue advantage over those who submit authentic, independently formulated

answers. This disrupts the principle of meritocratic selection, leading to biased hiring outcomes.

3. Operational Inefficiencies and Workplace Disruptions: Recruiting AI-reliant candidates can translate into lower productivity, increased onboarding difficulties, and higher attrition rates, as these individuals may lack the critical thinking abilities necessary to navigate complex, real-world business challenges. This necessitates additional investments in training and supervision, thereby escalating operational costs.
4. Diminished Originality and Innovation: AI-generated responses typically adhere to predictable linguistic patterns and formulaic reasoning, reducing the diversity of thought and limiting the organization's capacity for creative problem-solving. Over time, an overreliance on AI-assisted responses could lead to homogenized decision-making, diminishing the organization's ability to innovate and adapt to evolving challenges.

LITERATURE REVIEW

The advancement of Natural Language Processing (NLP) and machine learning has significantly transformed how textual data is analyzed, classified, and understood. Several studies have explored different methodologies to optimize text classification, authorship verification, and AI-generated content detection. Our approach integrates deep learning models like BERT and GPT-2, unsupervised clustering via K-Means, and latent space representation using autoencoders. The following literature provides a foundation for our methodology.

Devlin et al. (2019) introduced BERT (Bidirectional Encoder Representations from Transformers), a transformer-based NLP model that significantly improved context understanding in textual data. Unlike traditional word embedding methods, BERT processes words bidirectionally, capturing both preceding and succeeding word contexts. This advancement enhanced various NLP applications, including text classification and sentiment analysis.

Radford et al. (2019) presented GPT-2 (Generative Pretrained Transformer-2), an autoregressive language model capable of generating human-like text. Unlike BERT, GPT-2 is trained for next-word prediction, making it highly effective for text generation and coherence evaluation. Researchers have explored its potential in AI-generated text detection, comparing its embeddings with human-written text to identify linguistic discrepancies.

Mikolov et al. (2013) pioneered the word2vec model, which introduced word embeddings based on distributional semantics. This laid the groundwork for advanced embedding techniques, such as BERT and GPT-2, that leverage deep learning for contextualized word representations. These embeddings have been widely used for text classification, clustering, and sentiment analysis.

Hinton and Salakhutdinov (2006) explored the concept of autoencoders, a neural network-based dimensionality reduction technique that learns compact feature representations. Autoencoders have been instrumental in capturing hidden patterns within data, making them useful for anomaly detection and text clustering. In our study, autoencoders extract meaningful latent representations from textual responses to aid in classification.

MacQueen (1967) introduced the K-Means clustering algorithm, which groups data points based on similarity. Researchers have applied K-Means in NLP for topic modeling and document clustering, demonstrating its efficiency in identifying inherent patterns within unstructured text. In our methodology, we utilize K-Means to group latent representations of responses, allowing us to assess text similarity and classify new responses accordingly.

Reimers and Gurevych (2019) proposed Sentence-BERT (SBERT), an extension of BERT optimized for sentence similarity tasks. SBERT reduces computational complexity while maintaining high semantic accuracy, making it ideal for evaluating text similarity. This research aligns with our approach, where we compute cosine similarity between new responses and pre-existing BERT/GPT-2 embeddings to determine authenticity.

Recent advancements in authorship verification and AI-generated text detection have leveraged deep learning models, clustering techniques, and latent space representations. Our approach builds upon these studies, integrating multiple methodologies to enhance the accuracy and robustness of response classification. The combination of handcrafted linguistic features, transformer-based embeddings, unsupervised clustering, and deep learning-based dimensionality reduction provides a comprehensive framework for analyzing textual responses effectively.

RATIONALE

The rationale for this project stems from the increasing need to evaluate and authenticate textual responses, particularly in academic and recruitment settings. As AI-generated text becomes more sophisticated, distinguishing between human-written and machine-generated content poses a significant challenge. Our framework aims to bridge this gap by developing a robust classification model that leverages deep learning, clustering techniques, and latent space representations to ensure accurate text analysis.

By integrating **BERT and GPT-2 embeddings**, we capture the contextual and semantic depth of responses, enabling us to differentiate between AI-generated and human-authored content effectively. The incorporation of **autoencoders** helps in dimensionality reduction while preserving key linguistic patterns, ensuring the extraction of meaningful latent features. Furthermore, **K-Means clustering** allows us to group similar textual responses, providing an additional layer of classification based on learned patterns.

Our approach ensures fairness and transparency in candidate evaluations by automating and standardizing the classification process. This is particularly relevant in the context of recruitment and academic assessments, where AI-generated responses may pose challenges in credibility and originality verification. By creating a **Django-based framework** and a structured database, we streamline the process of collecting, storing, and analyzing student responses, ensuring a seamless experience for both applicants and evaluators.

Ultimately, this project contributes to enhancing the reliability of text classification methods, promoting integrity in assessments, and enabling organizations to make informed decisions based on data-driven insights. As AI-generated content continues to evolve, adopting advanced classification mechanisms will play a pivotal role in maintaining authenticity, credibility, and fairness in various domains.

AIMS & OBJECTIVES

Aims:

1. To develop an AI-driven text classification framework that accurately differentiates between human-written and AI-generated responses.
2. To enhance the recruitment process by creating a system where candidates can submit their details, respond to case studies, and have their responses evaluated in a structured, automated, and unbiased manner.
3. To integrate Natural Language Processing (NLP) techniques such as BERT, GPT-2, K-Means clustering, and Autoencoders to effectively analyze text responses and extract meaningful patterns.
4. To streamline data storage and analysis by implementing a Django-based web framework and a centralized database where applicant information and responses are securely stored and processed.
5. To contribute to the academic and recruitment sectors by ensuring that textual evaluations remain fair, transparent, and resistant to manipulation by AI-generated content.

Objectives:

1. Collect and preprocess a diverse dataset of human-written and AI-generated responses by performing tokenization, stop word removal, and vectorization techniques.
2. Extract features from text using BERT embeddings for contextual meaning, GPT-2 embeddings for fluency analysis, and handcrafted features such as sentence length, lexical diversity, and readability scores.
3. Reduce dimensionality using Autoencoders while preserving key textual patterns and apply K-Means clustering to group responses into meaningful categories.
4. Develop a probability-based classification system by comparing new responses against previously classified data using cosine similarity and latent space analysis.
5. Design and implement a Django-based web framework where students can submit their details and responses, which are then stored securely in a centralized database.
6. Regularly update and fine-tune the model to improve accuracy and adaptability while ensuring robustness against evolving AI-generated content.

DATA PREPARATION

6.1 ABOUT THE DATA:

The dataset for this project was meticulously curated to ensure contextual relevance, linguistic diversity, and analytical depth. The data collection process was structured around multiple business case studies, each designed to simulate real-world corporate challenges across diverse industry domains. These case studies served as the foundation for generating a robust dataset that reflects professional decision-making, strategic thinking, and analytical reasoning.

6.1.1 Key Themes Across Case Studies:

1. **Process Optimization & Efficiency** – Streamlining operations, improving quality control, and enhancing financial performance through structured analysis.
2. **Data-Driven Decision Making** – Using survey analytics, statistical modeling, and market research to extract actionable insights.
3. **Strategic Talent Management** – Overcoming recruitment challenges, improving employee retention, and strengthening employer branding.
4. **Project Execution & Risk Mitigation** – Managing resource allocation, addressing survey response drop-offs, and optimizing workflow timelines.

6.2 RESPONSE GENERATION & AI VARIABILITY:

Each response was generated using advanced AI models capable of creating contextually relevant and coherent text. To ensure variability and prevent redundancy, the following techniques were applied:

- **Lexical Variation & Paraphrasing**: AI-generated responses were systematically paraphrased to introduce lexical diversity while retaining core meaning. This helped train models to recognize similar ideas expressed in different ways.
- **Varying Sentence Structures**: The dataset included responses with different grammatical structures short, direct responses as well as longer, more detailed explanations to account for natural variations in human communication.
- **Complexity Spectrum**: AI responses were tailored to include a mix of simple, easy-to-read explanations and sophisticated, high-vocabulary responses to simulate the variety found in real-world interactions.

To create a balanced and comprehensive dataset, responses were categorized based on multiple linguistic attributes:

- **Length Distribution**: Responses varied in length from single sentence replies to multi-paragraph explanations.
- **Formality Spectrum**: Some responses were written in a highly formal tone, suitable for professional or business communication, while others had a more conversational or informal approach.
- **Domain-Specific Vocabulary**: Responses incorporated industry-specific terminology relevant to different business functions, ensuring the model could recognize specialized language.

METHODOLOGY

7.1 NLP TECHNIQUES FOR FEATURE CREATION:

In the field of AI-generated text detection, Natural Language Processing (NLP) techniques play a crucial role in distinguishing between human-written and AI-generated responses. A key approach involves **feature engineering** and **linguistic analysis**, which focus on the structural and lexical properties of text. Below is an elaboration of the core techniques utilized for AI response detection:

7.1.1 Readability & Complexity Metrics:

AI-generated responses are typically more structured, grammatically precise, and easier to read compared to human-written text, which often exhibits greater variability in sentence structure and complexity. To quantify these differences, the following metrics were employed:

1. **Flesch Reading Ease Score** – Measures text readability based on sentence length and syllable count. AI-generated text generally scores higher due to its structured and coherent nature, whereas human-written text tends to include a mix of complex and simple sentences.
2. **Gunning Fog Index** – Assesses text complexity by evaluating sentence length and the proportion of polysyllabic words. AI-generated responses often exhibit lower complexity scores compared to human responses, which may include more intricate sentence structures.
3. **Type-Token Ratio (TTR)** – Examines vocabulary diversity by comparing the number of unique words (types) to the total number of words (tokens). AI-generated responses often use a more uniform vocabulary, whereas human-written text displays greater word variation, leading to a higher TTR.

7.1.2 Sentence & Word Structure Analysis:

Another critical aspect of AI text detection is the examination of sentence and word patterns. This includes:

4. **Average Sentence Length –** AI-generated responses tend to maintain a consistent sentence length, contributing to a uniform and structured flow of text. In contrast, human writing naturally varies in sentence length, incorporating a mix of short and long sentences.

5. **Word Frequency Distribution –** AI models often reuse certain phrases and stylistic patterns, making their responses more predictable. By analyzing word frequency distributions, AI-generated text can be identified based on repetitive language patterns, whereas human writing typically shows greater lexical diversity.
6. **Perplexity Score Computation**– Perplexity is a statistical measure used to evaluate how predictable a text is. Lower perplexity scores indicate a more structured and formulaic text, which is characteristic of AI-generated responses. Human-written text, by contrast, tends to have a higher perplexity score due to the natural variability in sentence construction, word choice, and writing style.

7.2 Deep Learning-Based Feature Extraction Using BERT:

To further enhance AI-generated text detection, deep learning-based feature extraction was employed using BERT embeddings. Unlike traditional NLP techniques, embeddings provide a dense numerical representation of text, allowing the model to capture semantic relationships and writing styles more effectively.

7.2.1 BERT: Architecture, Working Mechanism, and Its Role in AI Detection:

1. Overview of BERT (Bidirectional Encoder Representations from Transformers):

- BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based deep learning model designed to enhance natural language understanding (NLU). Developed by Google AI in 2018, BERT significantly improved the ability of models to process and understand human language by considering both left and right context in a sentence. Unlike previous NLP models, which processed text either left-to-right or right-to-left, BERT reads an entire sequence of words simultaneously (bidirectionally), leading to superior contextual understanding.
- This innovation made BERT a state-of-the-art model in various NLP tasks such as text classification, sentiment analysis, named entity recognition (NER), question answering, and AI-generated text detection.

2. Key Innovations in BERT:

- BERT introduced several key innovations that distinguished it from traditional NLP models:

    1. Bidirectional Contextual Understanding:

1. Traditional models like LSTMs and GRUs process text sequentially (either left-to-right or right-to-left). However, this limits their ability to understand how words relate **to both previous and future words** in a sentence.
2. BERT overcomes this limitation by processing an **entire sequence of words at once**, using **bidirectional self-attention**, ensuring **deeper semantic representation**.
3. This enables **better understanding of ambiguous words, context-dependent meanings, and relationships between words**.

3. Pretraining & Fine-Tuning Paradigm:

1. BERT follows a two-step training process:
    1. **Pretraining**: The model is trained on large-scale datasets using unsupervised learning to understand general language patterns.
    2. **Fine-Tuning**: The pretrained model is further trained on domain-specific or task-specific datasets using supervised learning.

This transfer learning approach reduces the need for extensive labeled data and allows BERT to generalize well across different NLP tasks.

4. Transformer-Based Architecture:

BERT is built on the Transformer architecture, first introduced by Vaswani et al. (2017).
It uses self-attention mechanisms to determine how different words relate to each other, making it highly effective for text comprehension.

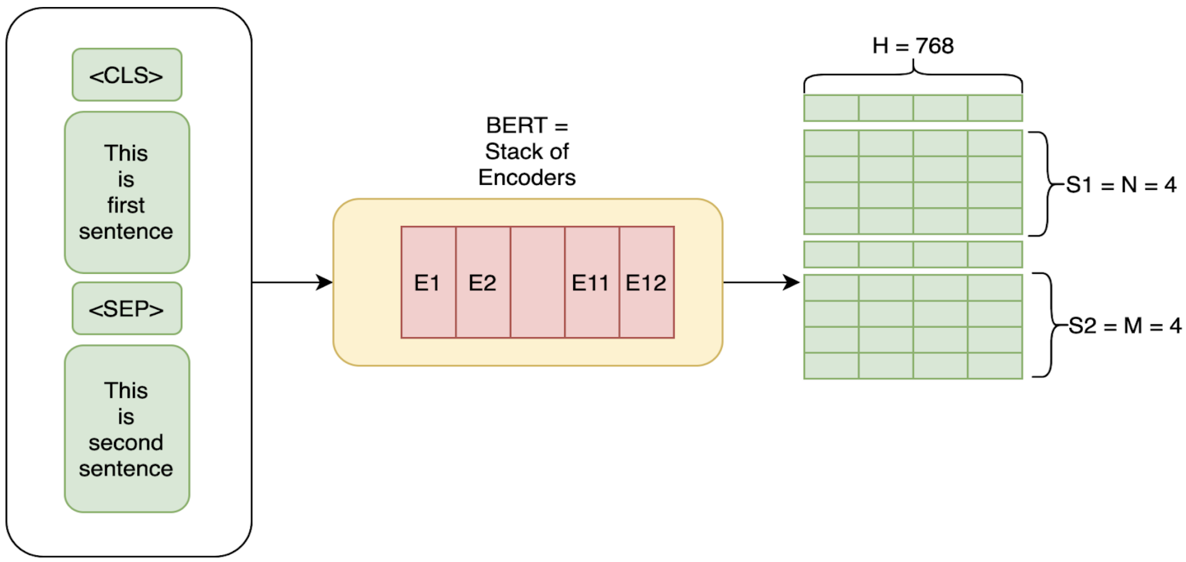


Fig. 7.a

7.2.2 Working Mechanism of BERT

BERT processes text in multiple stages, which include **tokenization, embedding generation, transformer layers, and output processing**.

1. **Tokenization & Input Representation**

Before feeding text into BERT, it undergoes **tokenization** to convert words into a format understandable by the model. BERT uses the **WordPiece Tokenizer**, which breaks words into subwords to handle **out-of-vocabulary (OOV) words** efficiently.

Key Input Components in BERT:

- **[CLS] Token**: A classification token placed at the beginning of every input sentence. The embedding of this token is later used for classification tasks.
- **[SEP] Token**: A separator token used to distinguish between different sentences in input.
- **Token Embeddings**: Each word (or subword) is converted into a numerical vector representation.
- **Segment Embeddings**: If multiple sentences are provided, segment embeddings help the model differentiate between them.
- **Positional Embeddings**: Since transformers do not process text sequentially, positional embeddings help retain the **word order information**.

2. **Embedding Layer**

After tokenization, the input tokens are passed through the **embedding layer**, where they are converted into dense numerical vectors. These embeddings represent the **semantic meaning** of words and phrases.

1. **Transformer Layers & Self-Attention Mechanism**

The core strength of BERT lies in its **multi-layer transformer architecture**, which enables deep contextual understanding.

Self-Attention Mechanism

BERT uses **multi-head self-attention**, where each word in a sentence **attends to all other words**. This mechanism helps capture:

- **Long-range dependencies** (e.g., relationships between words far apart in a sentence).
- **Contextual relationships** (e.g., differentiating meanings of the same word in different contexts).

Each transformer layer consists of:

1. **Multi-Head Self-Attention**: Determines how much focus should be given to different words in the sequence.
2. **Feedforward Neural Network**: Further processes contextual embeddings.
3. **Layer Normalization & Dropout**: Improves stability and prevents overfitting.

Output Processing

After passing through transformer layers, the output consists of **contextual word embeddings** that encode the meaning of words **in relation to their surrounding context**.

For **classification tasks** (such as AI-generated text detection), the embedding of the **[CLS] token** is extracted as the final representation of the sentence.

7.3 DEEP LEARNING-BASED FEATURE EXTRACTION USING GPT-2:

GPT-2 (Generative Pre-trained Transformer 2) is a **large-scale generative language model** developed by **OpenAI**. Released in 2019, GPT-2 marked a significant advancement in natural language generation by demonstrating the ability to generate **coherent, contextually relevant text** based on a given input. Unlike **BERT**, which is a bidirectional model used for understanding text, GPT-2 is **unidirectional** (left-to-right), meaning it predicts the next word in a sequence based on previous words.

GPT-2's success in generating human-like text made it one of the most widely adopted models for various NLP tasks, including **text generation, language modeling, and AI-generated text detection**. Its ability to predict the most probable word in a sequence based on a preceding context allows it to create highly fluent and structured sentences.

7.3.1 Key Innovations in GPT-2

GPT-2 introduced several innovations that distinguished it from traditional language models:

Unidirectional Contextual Understanding

- GPT-2 processes text in a **left-to-right sequence**, where each word is predicted based on the previous words in the sequence.
- This contrasts with **BERT's bidirectional approach**, which reads the entire sequence at once, capturing relationships from both directions.
- GPT-2's **unidirectional nature** makes it especially effective for **text generation tasks** where generating coherent and grammatically correct sentences is essential.

Large-Scale Pretraining

- GPT-2 is pretrained on a massive corpus of text data, enabling it to learn general language patterns, syntax, and grammar.
- This pretraining allows GPT-2 to perform a wide range of **zero-shot** tasks without needing task-specific fine-tuning, making it highly adaptable across various NLP domains.

Transformer Decoder Architecture

- GPT-2 follows the **Transformer architecture**, specifically the **decoder-only** variant, as opposed to BERT's encoder-decoder architecture.
- The **self-attention mechanism** in GPT-2 allows it to model the relationships between words across long-range sequences, despite the unidirectional processin

7.3.2 Working Mechanism of GPT-2

GPT-2 processes text in several stages, which include **tokenization, embedding generation, transformer decoder layers, and output generation**.

1. Tokenization & Input Representation

The first step in GPT-2 is **tokenization**, where the input text is converted into **subword tokens** using **Byte-Pair Encoding (BPE)**.

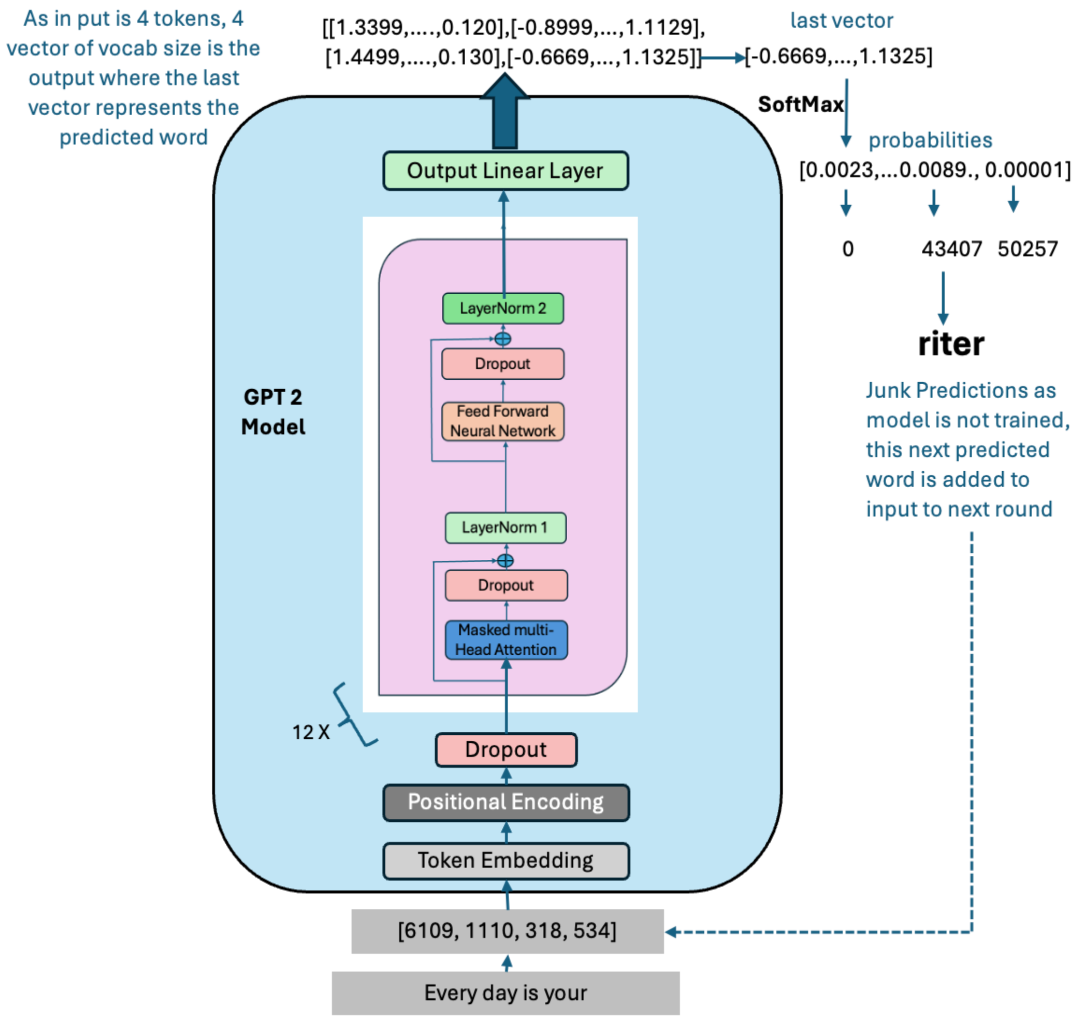


Fig. 7.b

2. **Key Input Components in GPT-2:**
   - **Tokens**: The input text is broken down into subwords or words, depending on the vocabulary learned during pretraining.
   - **Embeddings**: Each token is represented by a **dense vector** (embedding) that encodes the token's semantic meaning. These embeddings are fed into the model for further processing.
3. **Transformer Decoder Layers & Self-Attention Mechanism**

GPT-2 consists of **multiple layers of transformers** with the following components:

a. Masked Self-Attention Mechanism

- GPT-2 employs **masked self-attention**, meaning each word in the sequence only attends to previous words (left-to-right), preventing the model from peeking ahead during text generation.

- This helps ensure that the model generates text sequentially, making predictions based on the preceding context.

b. Feedforward Neural Network

- Each attention output is processed by a feedforward neural network to learn patterns and relationships between words.

c. Layer Normalization & Dropout

- These techniques are used to stabilize training and reduce the likelihood of overfitting, ensuring that GPT-2 can generalize well to unseen data.

d. Output Generation

- After processing through multiple transformer layers, GPT-2 generates a **probability distribution** over the vocabulary for the next word in the sequence.
- GPT-2 generates text by **sampling words** from the probability distribution, iterating this process to produce coherent, contextually relevant sentences.
- The model generates output one token at a time, using the **previously generated tokens** as context to predict the next one.

7.3.3 Why We Used GPT-2 in AI-Generated Text Detection

GPT-2 was chosen for **AI response detection** due to its **ability to generate text that follows predictable and structured patterns**, which is often a distinguishing feature of AI-generated content.

1. AI-Generated Text Has Lower Perplexity

- **Perplexity** is a measure of how well a language model predicts a sample. AI-generated text tends to have lower perplexity scores, meaning it is more predictable compared to human-written text.
- Since GPT-2 generates text sequentially, its responses exhibit **predictable structures**, making it easier to distinguish from human responses, which are more diverse and complex.

2. Semantic Pattern Recognition

- GPT-2 embeddings can capture the **semantic patterns** in text, allowing us to detect similarities between AI-generated content and known AI responses.
- These embeddings help identify recurring **predictable patterns** and **structural regularities** that are often absent in human writing.

3. **Effective Feature Extraction for AI Detection**

- By extracting the **embeddings** from GPT-2, we can analyze the **statistical properties** of generated text, making it easier to **cluster AI-generated responses**.
- This clustering approach allows us to classify new responses based on their similarity to known AI patterns, improving the accuracy of **AI detection**.

7.4 K-MEANS CLUSTERING IN AI DETECTION: A DETAILED OVERVIEW:

7.4.1 Introduction to K-Means Clustering

K-Means is one of the most popular unsupervised learning algorithms used for clustering, which means grouping similar data points into clusters. It is a type of **distance-based clustering** method where the goal is to partition a dataset into **K distinct, non-overlapping subsets** based on similarity. K-Means has found extensive applications in various domains, including text classification, anomaly detection, image compression, and, as in your project, distinguishing between AI-generated and human-generated content.

7.4.2 Working of K-Means Clustering

The algorithm works iteratively to refine clusters, optimizing the position of centroids based on the data points assigned to them. Let's break down the key steps:

1. **Choosing the Number of Clusters (K)**

The first step in the K-Means algorithm is selecting the number of clusters, **K**, which determines the number of distinct groups the data will be split into.

- **Selection of K** is typically done experimentally by evaluating multiple values and measuring the **within-cluster sum of squares (WCSS)**, silhouette scores, or using techniques like the **elbow method**.
- In the context of AI detection, K represents the number of categories we expect in the data, such as **AI-generated responses** and **human-generated responses**. The goal is to identify the most meaningful separation between these two groups.

2. **Assigning Data Points to Clusters**

Once **K** clusters are determined, the algorithm proceeds by assigning each data point to the nearest cluster center.

- The **distance metric** commonly used is **Euclidean distance**, which measures the straight-line distance between points. However, other distance measures like **cosine similarity** can also be used when dealing with textual data (such as embeddings).

- Each **AI-generated or human-written response** is assigned to the cluster whose centroid (center of the cluster) is closest in terms of similarity. In your project, this would involve assigning **textual data** (e.g., embeddings of AI and human responses) to the correct cluster based on features like sentence structure and word choice.

3. **Updating Cluster Centers**

Once all data points are assigned to clusters, the cluster centers (centroids) are recalculated.

- The new centroids are determined by averaging the positions of all data points within each cluster.
- The centroid represents the **mean position** of the points in that cluster. For AI-generated and human-written responses, centroids will represent the common traits of each group, such as **predictability** and **structured language** in AI responses versus the **creative flow** and **variability** in human responses.

4. **Iterative Optimization**

After updating the centroids, the algorithm **reassigns data points** to the nearest centroid based on the new positions. This process is repeated until there is no significant change in the centroids' positions—i.e., the algorithm converges.

- **Convergence** means that the clusters are stable, and the classification of each data point is no longer changing.
- In the case of AI detection, once the algorithm has converged, we can be confident that AI and human responses are distinctively clustered based on their underlying features.

7.4.3 Why K-Means is Effective in AI Detection

K-Means clustering can significantly enhance the process of detecting AI-generated text by leveraging its ability to distinguish underlying patterns in data. Below are the reasons why K-Means is a powerful tool for this purpose:

1. Separating AI vs. Human Responses

- AI-generated text tends to follow predictable patterns and structures due to the way machine learning models generate text.
- **AI responses** often have lower variability, and they might exhibit **repetitive structures**, **formal tone**, and **consistent phrasing**.
- In contrast, **human responses** are more diverse in style, tone, and complexity.
- K-Means can exploit these differences by clustering responses into distinct groups, making it easier to identify AI-generated content by its similarity to other AI responses.

2. Identifying Paraphrased AI Responses

Even when AI-generated content is paraphrased or slightly modified, it often retains characteristics that are typical of AI-generated text, such as repetitive phrasing or unnatural sentence structures.

- These subtle similarities make AI-generated responses **cluster together**, even when not identical to original versions.
- K-Means is effective in grouping paraphrased AI content into the same cluster, ensuring that slight modifications of AI text still get classified correctly.

3. Enhancing Overall Classification Accuracy

By combining **K-Means clustering** with **deep learning embeddings**, we create more robust classification models for AI detection.

- **Embeddings**, such as those generated by **GPT-2** or **BERT**, capture the deeper semantic meaning of words and phrases. These embeddings allow K-Means to cluster responses based on their **contextual similarity** rather than just surface-level features (like word frequency).
- This leads to more accurate detection of **AI-generated text**, as it can capture subtle differences between AI and human responses, improving the overall classification accuracy.

4. Scalability and Efficiency

K-Means is computationally efficient, making it well-suited for large datasets. This scalability is crucial when dealing with substantial volumes of text data, such as responses from **AI and human users**.

- The algorithm's ability to handle large datasets ensures that even when the corpus grows, K-Means can quickly cluster new responses and maintain its accuracy without extensive retraining.

K-Means is adaptable to new data. As more AI-generated content becomes available or the writing styles of humans evolve, K-Means can be used to re-cluster the dataset, ensuring that the AI detection model remains effective over time.

Clustering Results for each dataset:

1. Data Analyst (DA)

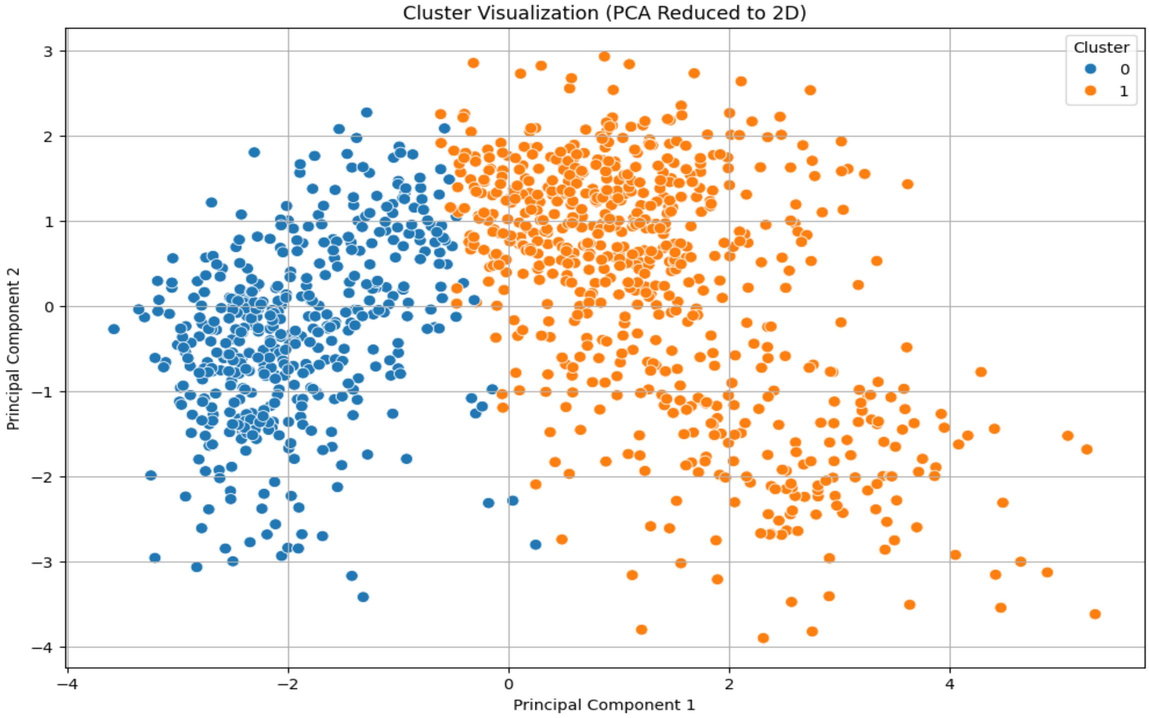


Fig. 7.c

2. Graduate Engineer Trainee (GET):

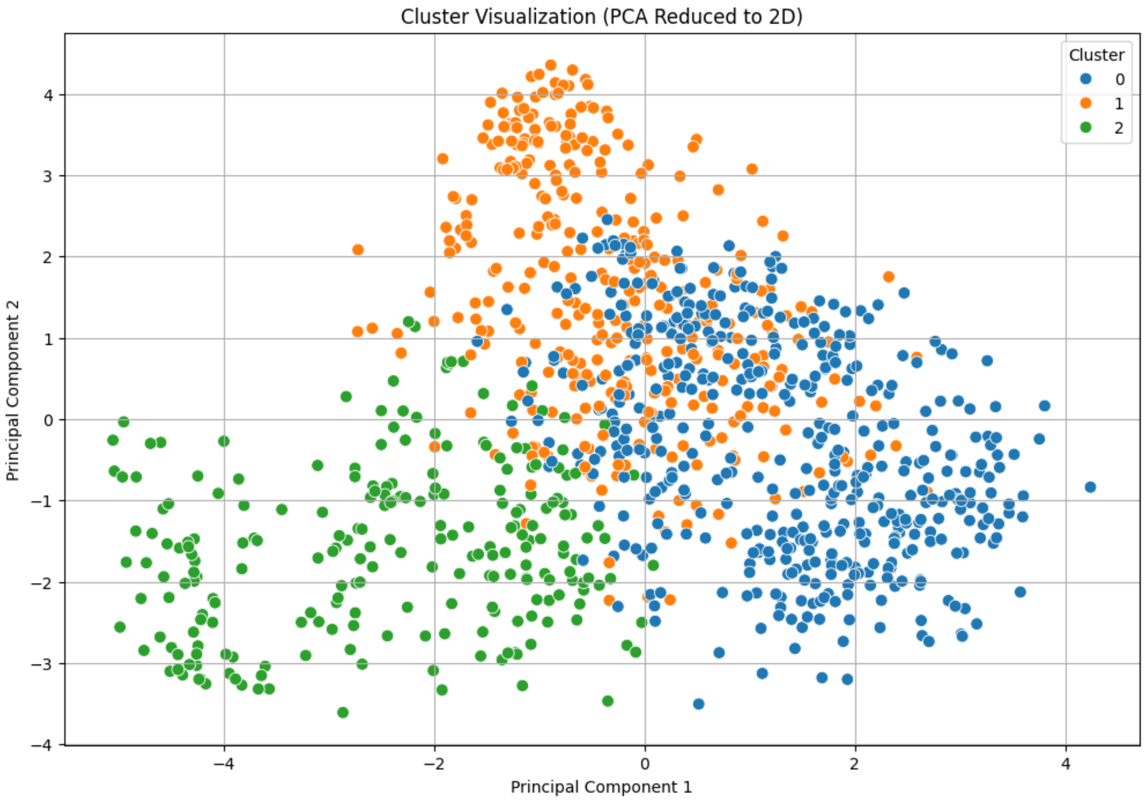


Fig. 7.d

3. Project Manager (PM):

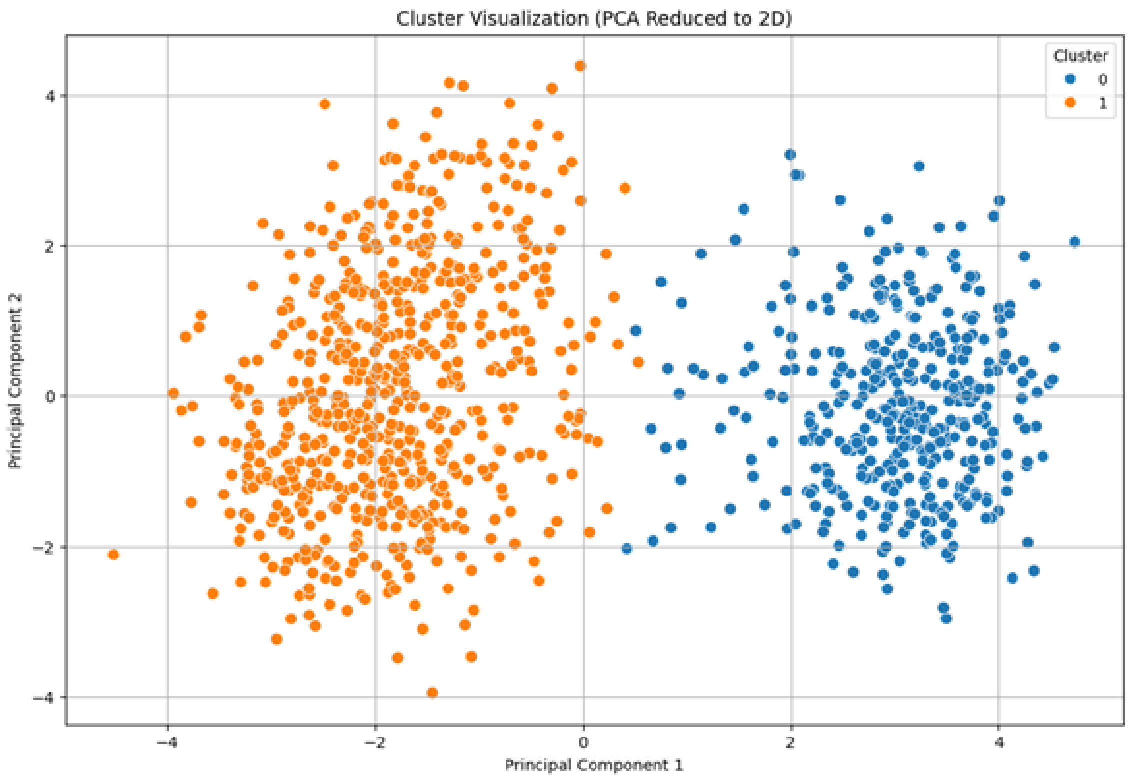


Fig. 7.e

4. Talent Acquisition (TA):

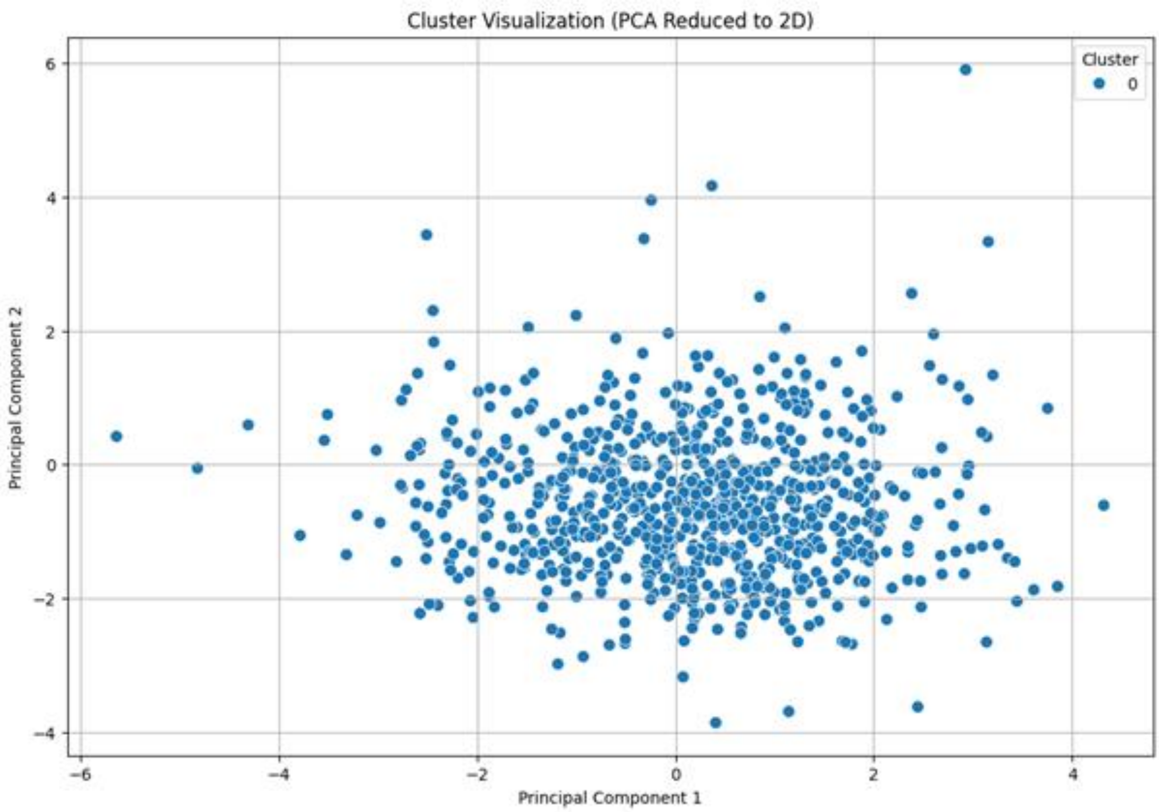


Fig. 7.f

7.5 AUTOENCODERS IN AI DETECTION: A DETAILED OVERVIEW

7.5.1 INTRODUCTION TO AUTOENCODERS

Autoencoders are a type of **unsupervised neural network** that excel in detecting anomalies by learning compact representations of input data. They are widely used in tasks like **dimensionality reduction, noise removal, and anomaly detection**.

In the context of AI detection, autoencoders can differentiate **AI-generated responses from human-written responses** by analyzing their ability to **reconstruct text data**. Since AI-generated text follows distinct patterns compared to human responses, it exhibits **higher reconstruction errors**, making it easier to detect.

7.5.2 HOW AUTOENCODERS WORK

Autoencoders consist of three main components:

1. **Encoder Layer**

- The encoder compresses input data into a **lower-dimensional representation**, also known as **latent space**.
- This layer **extracts essential features** while **removing redundant information** or noise.
- For text data, embeddings (such as BERT or Word2Vec representations) can be passed through an encoder to extract meaningful patterns.
- In AI detection, the encoder captures differences between AI-generated and human-generated text by identifying unique linguistic characteristics.

2. **Bottleneck Layer**

- The bottleneck layer is the **smallest, most compact representation** of the input data.
- This layer helps the autoencoder focus on **patterns and structures** that distinguish AI-generated responses from human-written ones.
- In AI detection, human responses might exhibit **greater variability**, whereas AI-generated responses tend to have **more uniform structure**, making them easier to separate.

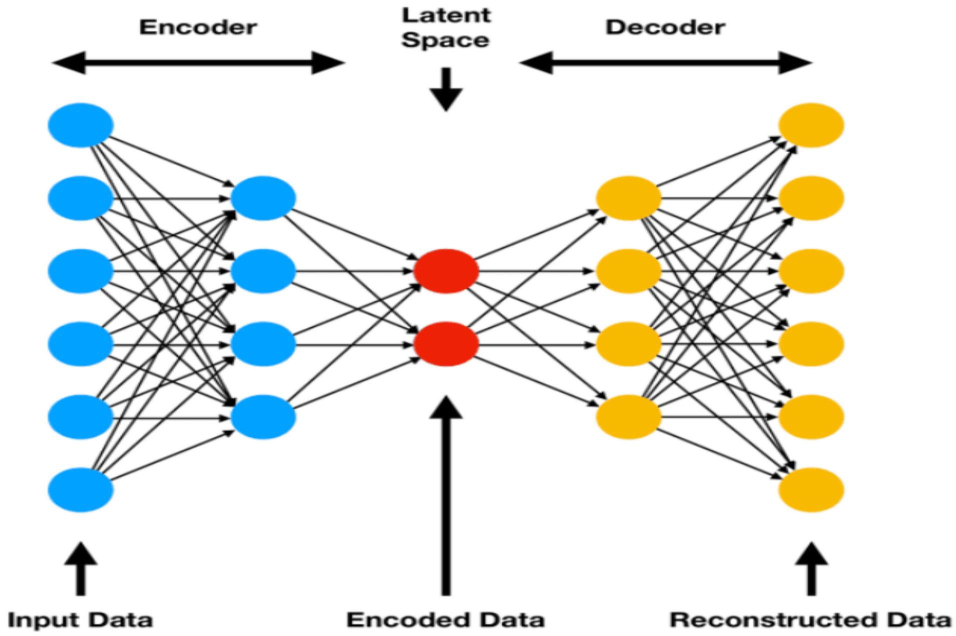
3. **Decoder Layer**

- The decoder attempts to **reconstruct the original input** from its compressed latent representation.
- If the reconstruction is **accurate**, it means that the input follows the learned patterns.

- However, **AI-generated responses often fail to reconstruct accurately**, leading to **higher reconstruction errors** because their linguistic structures are different from the learned human-writing patterns.

7.5.3 WHY AUTOENCODERS ARE EFFECTIVE IN AI DETECTION

1. **Identifying AI Responses as Anomalies**
   - Since autoencoders learn from **human-written responses**, AI-generated responses appear as **outliers** when passed through the model.
   - AI responses typically have **higher reconstruction loss**, meaning that their sentence structures and patterns **do not align well** with the learned human-writing distribution.
   - This makes autoencoders highly effective for **detecting AI-generated responses as**



anomalies.

Fig. 7.g

2. **Capturing Subtle Text Differences**
   - AI-generated content may seem human-like but still retains **specific patterns**, such as:
     - Repetitive phrasing.
     - Overuse of formal structures.
     - Unnatural flow or coherence.

- Even when AI responses are **paraphrased**, their underlying structures remain similar, allowing autoencoders to detect them **through small but significant differences in reconstruction error**.

3. **Improving Robustness of AI Classification**
   - By combining **autoencoder-based anomaly detection** with other techniques such as **deep learning embeddings and clustering (e.g., K-Means)**, the system becomes more **robust**.
   - **Hybrid Approach:**
     - Step **1:** Use **autoencoders** to filter out potential AI responses based on **reconstruction loss**.
     - Step **2:** Apply **K-Means clustering** to further separate AI-generated responses from human-written ones.
     - **Step 3:** Use **deep learning classifiers** (e.g., BERT, LSTMs) to improve classification accuracy.
   - This combination ensures **higher precision** in detecting AI responses, making the system **more effective and scalable**.

7.6 EXPLANATION OF AI RESPONSE CLASSIFICATION FUNCTION

The **AI Response Classification Function** is designed to determine whether a candidate's response is **AI-generated or human-written** by analyzing various linguistic and semantic features. The function utilizes **Natural Language Processing (NLP) techniques, deep learning embeddings, clustering algorithms, and anomaly detection models** to make an informed decision.

To ensure a fair and skill-based hiring process, this function evaluates responses based on their **structural patterns, contextual meaning, and similarity to known AI-generated responses**. The final classification is made using a **weighted probability score** that takes into account multiple similarity metrics.

7.6.1 FEATURE EXTRACTION FROM CANDIDATE RESPONSE

Before making a classification, the system extracts three types of features from the candidate's response:

1. Handcrafted Linguistic Features

These features analyze how the response is structured and how complex or predictable it is. AI-generated responses tend to have a highly structured and grammatically sound format, while human responses show natural variability.

Key linguistic features include:

- **Sentence Length and Word Count:** AI-generated responses tend to have consistent sentence lengths, whereas human responses vary naturally.
- **Readability Scores (Flesch-Kincaid, Gunning Fog Index):** AI responses are often highly readable and structured, while human responses can range in complexity.
- **Type-Token Ratio (TTR):** This metric measures vocabulary diversity. AI-generated responses tend to repeat similar phrases, leading to a lower TTR score.
- **Perplexity Score:** This measures how predictable a piece of text is. AI-generated responses typically have lower perplexity, as they follow a structured and repetitive format, while human responses exhibit higher unpredictability.

2. BERT Embeddings for Contextual Representation

BERT (Bidirectional Encoder Representations from Transformers) is a deep learning model that understands the context of words in a sentence. Unlike traditional NLP models that process words sequentially, BERT reads the entire text bidirectionally, meaning it considers both the left and right context of a word.

BERT embeddings are used in AI detection because they:

- Capture contextual meaning of words and phrases.
- Help distinguish AI-generated responses, which often have unnatural contextual flow.
- Provide a numerical representation of the response, allowing comparison with previously classified human and AI responses.

BERT processes text through multiple self-attention layers, where each word is assigned a weight based on its importance within the sentence. This allows the model to detect unusual phrasing patterns that may indicate AI-generated content.

3. GPT-2 Embeddings for AI-Generated Text Patterns

GPT-2 (Generative Pre-trained Transformer 2) is a generative language model designed to predict the next word in a sentence. It is highly effective at generating text that appears human-like but follows predictable structures.

GPT-2 embeddings are used because they:

- Identify AI-like sentence structures, as GPT-2 responses tend to be grammatically correct but lack human creativity.
- Compare the candidate's response with known AI-generated responses, helping detect similarities.
- Analyze text fluency and coherence, as AI-generated responses often have a highly structured yet somewhat unnatural flow.

Unlike BERT, GPT-2 is a unidirectional model, meaning it processes text in a left-to-right sequence. This makes it highly effective for detecting responses that follow an AI-generated prediction pattern

7.6.2 CLUSTERING-BASED SIMILARITY COMPUTATION

To further distinguish between human and AI-generated responses, K-Means Clustering is applied. Clustering is useful in AI detection because:

1. AI-generated responses tend to form distinct clusters, as they follow structured and repetitive patterns.

2. Human responses are more diverse, leading to a broader distribution in the dataset.

3. Paraphrased AI responses still retain some AI-like characteristics, making clustering effective in identifying them.

K-Means works by grouping similar responses together based on their numerical representation in a high-dimensional space. Each response is assigned to a cluster based on its similarity to previously identified human or AI responses. The distance between the candidate's response and the center of the AI cluster is measured. If the response is close to the AI cluster, it is likely AI-generated.

7.6.3 Cosine Similarity with Pre-Trained Embeddings

In addition to clustering, cosine similarity is used to compare the candidate's response with a database of previously classified AI and human responses. Cosine similarity is a mathematical measure that calculates the angle between two vectors, with values ranging from 0 (completely different) to 1 (identical).

Cosine similarity is applied to:

● **BERT embeddings:** This determines how similar the response is to known human-written responses.

● **GPT-2 embeddings:** This measures how similar the response is to previously detected AI-generated responses.

A high similarity to GPT-2 embeddings suggests that the response is AI-generated, while a high similarity to BERT embeddings indicates a human-written response.

7.6.4 Final Probability Calculation and Classification Decision

Once all similarity metrics are calculated, the system combines them into a weighted probability score to determine whether the response is AI-generated.

The final classification is based on the following:

- Clustering Similarity: How close the response is to known AI clusters.
- BERT Cosine Similarity: How much the response resembles human-written text.
- GPT-2 Cosine Similarity: How much the response resembles AI-generated text.

Since clustering is the most reliable indicator, it is given the highest weight in the probability score.

- If the probability score is 50% or higher, the response is classified as AI-generated.
- If the probability score is below 50%, the response is classified as human-written.

7.7 DJANGO-BASED FRAMEWORK & DATABASE IMPLEMENTATION

To facilitate the case study-driven hiring process, we developed a Django-based web framework that serves as the front-end and back-end system for candidate application, response submission, and AI-generated response detection. This framework ensures a structured, automated, and scalable recruitment process.

The platform operates as follows:

1. Candidate Registration & Application

- Candidates visit the web portal and fill in their personal details, including their name, email, contact information, and educational background.
- They select the job role they wish to apply for.

2. Case Study Assignment & Response Submission

- After selecting a job role, the candidate is redirected to a case study curated by the respective department.
- The candidate analyzes the case study and submits their response through the platform.
- Database Integration & Storage
- Candidate details and their responses are stored in a structured database using MySQL.
- The database maintains separate records for candidate profiles and submitted responses.

3. AI Detection Model Integration

- Once a response is submitted, it is automatically processed by the AI Detection System, which generates an AI Percentage.
- The classification results are stored in the database and made available for recruiters to review.

These are the snippets of the website that we have built using Django framework:

You can access the webpage by clicking on this link: http://amiportal.in:8002/

1. Home Page.

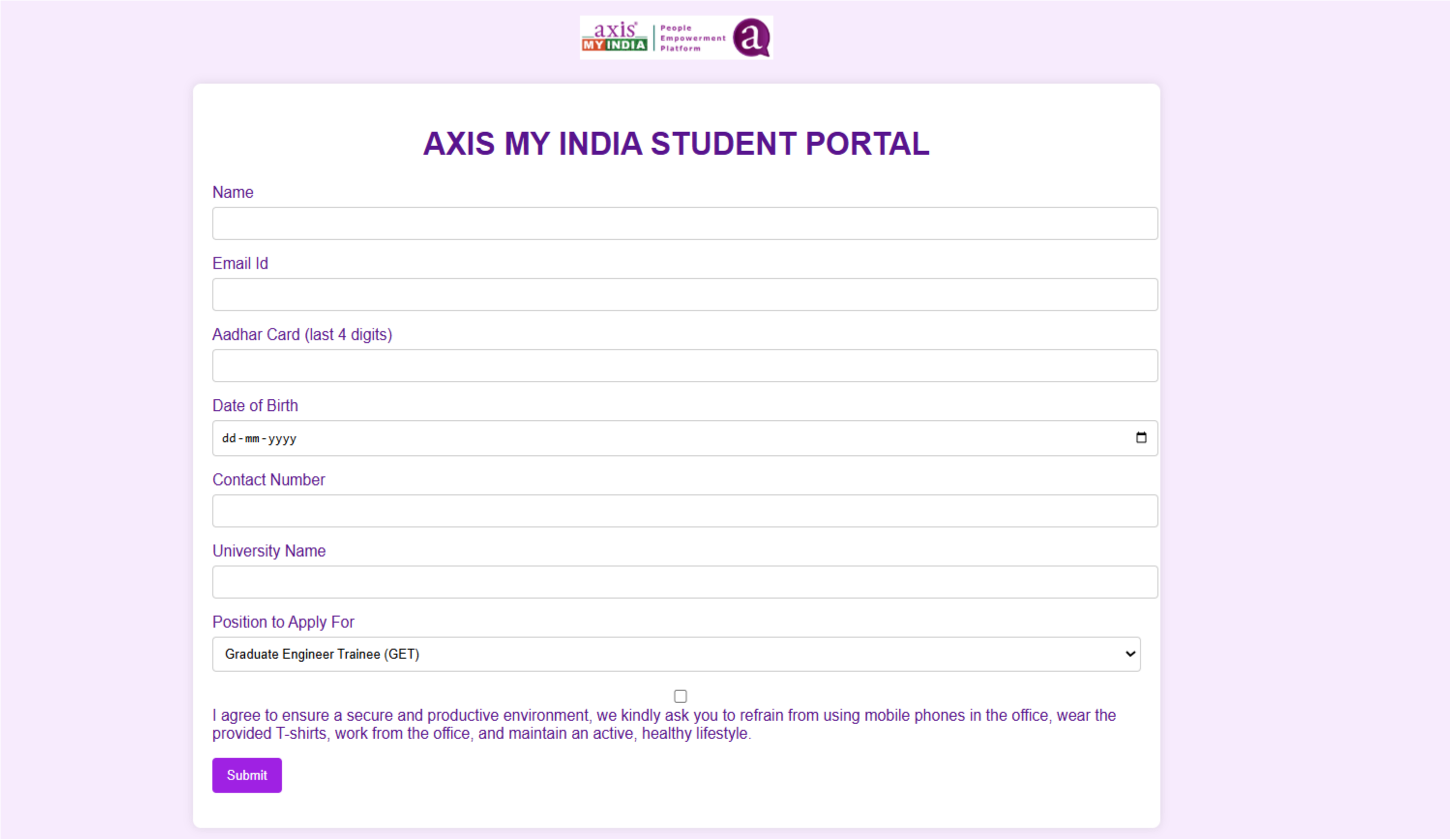


Fig. 7.h

2. Graduate Engineer Trainee (GET) case study section.

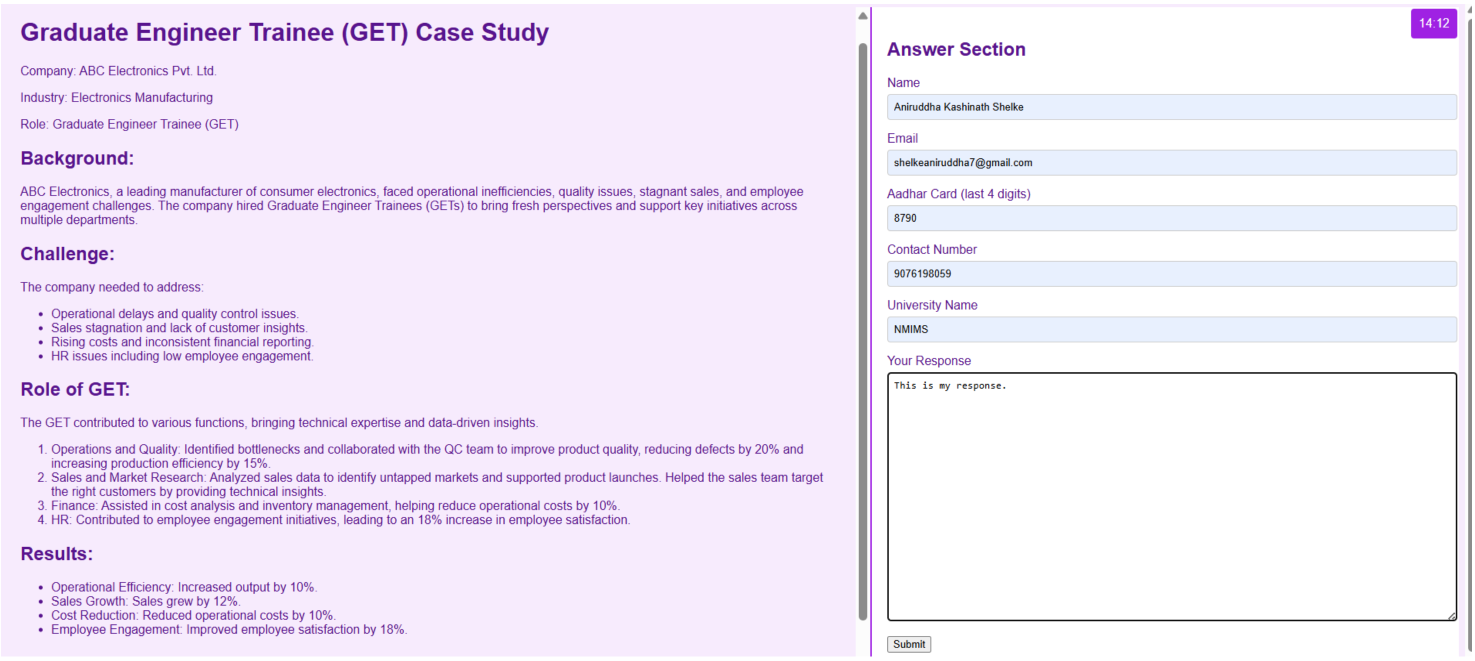


Fig. 7.i

3. Data Analyst (DA) case study section.

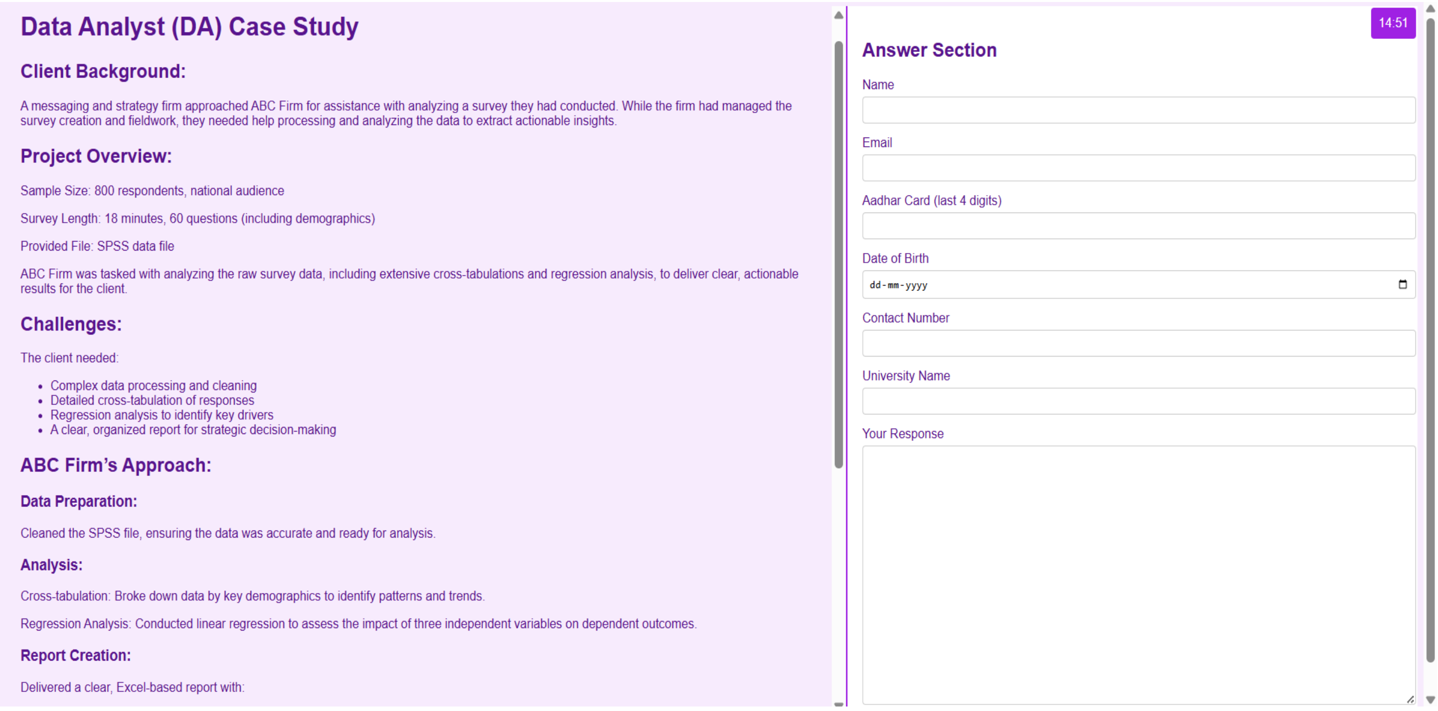


Fig. 7.j

4. Talent Acquisition (TA) case study section.

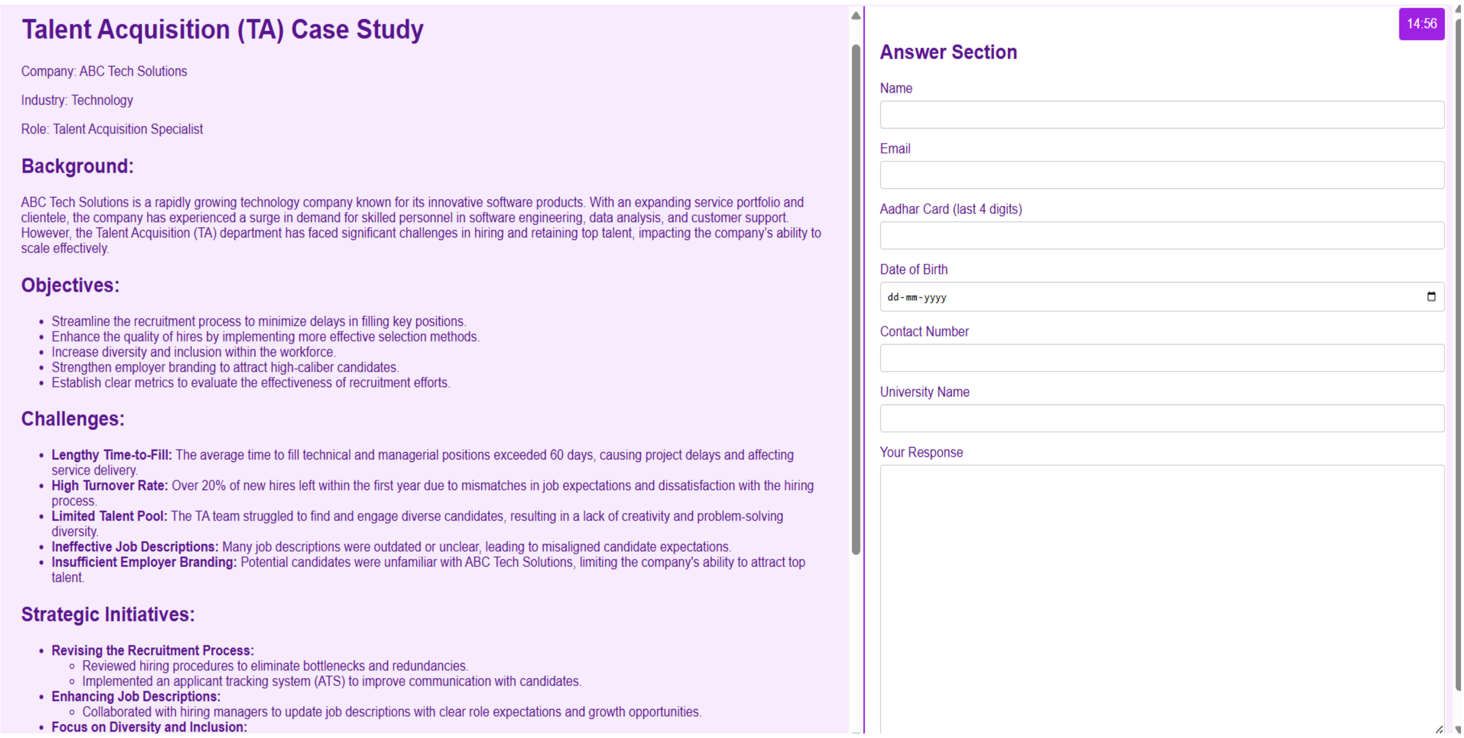


Fig. 7.k

5. Project Manager (PM) case study section.

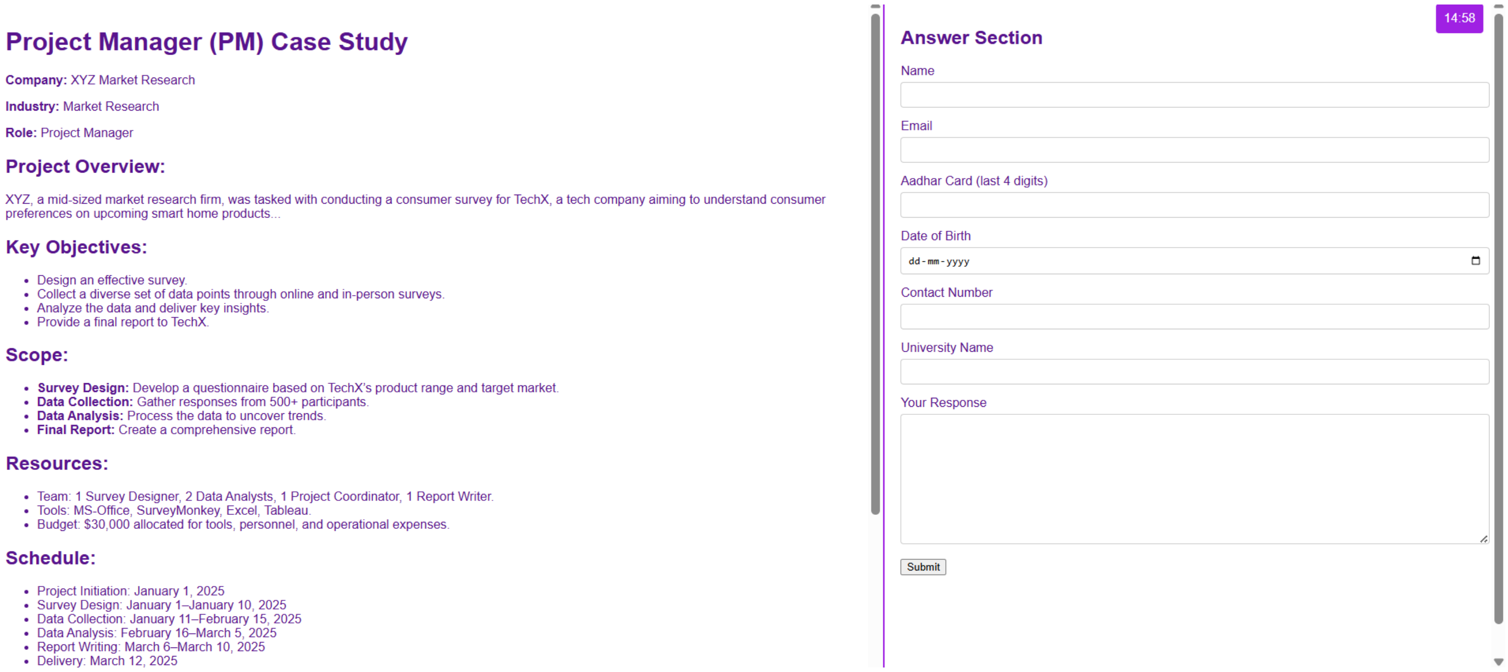


Fig. 7.l

6. Tab Switch Warning

Fig. 7.m

7. **Thankyou Page.**

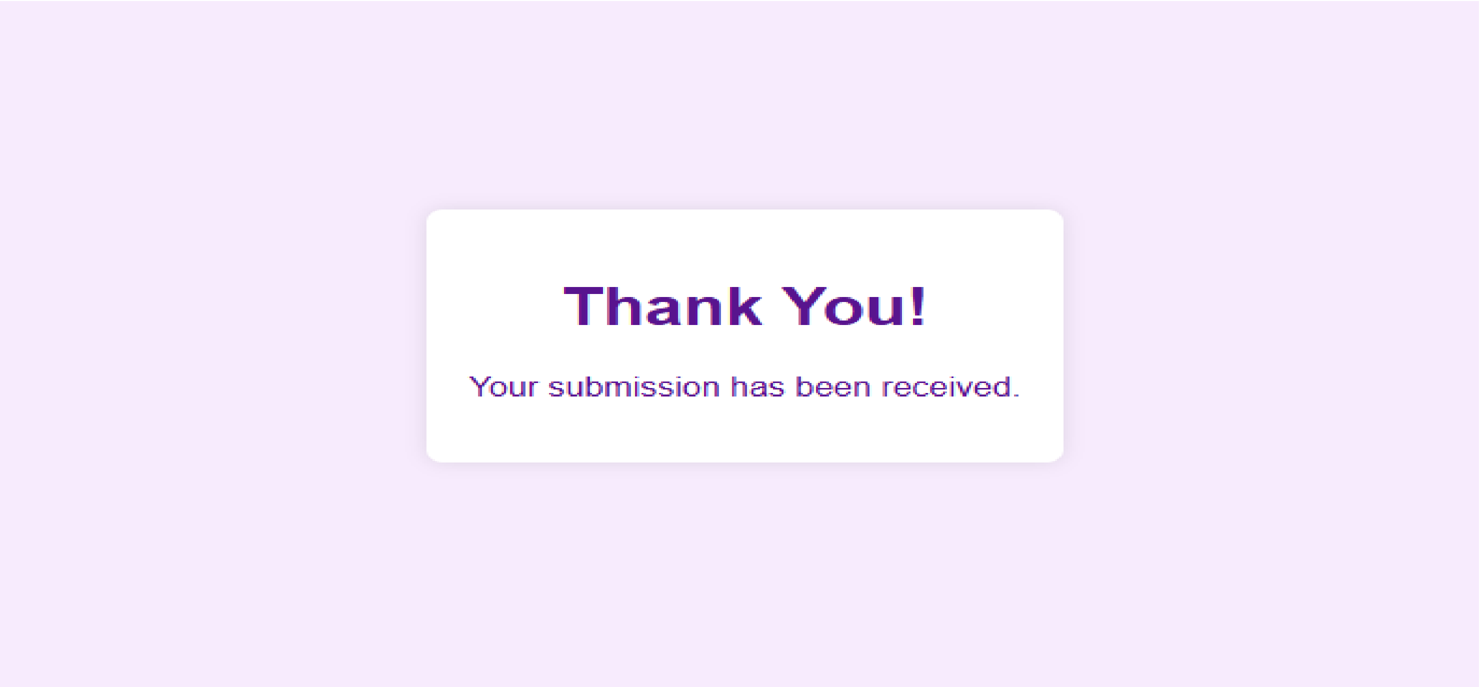


Fig. 7.n

RESULTS & DISCUSSION

After visiting the website and submitting the required details, the candidate is allowed to select a relevant case study and submit their response. This response is then processed through our proprietary AI-detection algorithm, which evaluates various linguistic and semantic features to determine the likelihood that the response was generated by an AI model. The algorithm outputs an AI Probability Score, representing the percentage confidence that the response is AI-generated.

Additionally, **3 candidates were shortlisted from Lady Shri Ram College, 13 from SRCC College, and 1 from Hindu College**.

As per the company's data protection policy, we are not permitted to display any actual candidate information. Hence, we are presenting a sample result below for demonstration purposes only.

Name	Email _Id	Aadhar_Card_last_4_digits	Date_of_Birth	Contact_Number	University_Name	Position_to_Apply_For	Response	AI_Percentage
Rahul Phatak	rahul04@gmail.com	4xx7	04-02-2002	945xxxxx70	NMIMS	Data Analyst (DA)	Survey data analysis experts can quickly process large volumes of survey responses, identify trends, and run sophisticated analyses like regression models. Their experience ensures the data is accurately interpreted, providing richer insights that lead to more informed strategic decisions,	59.8385036
Swetanshu Sahoo	swetanshukumar@gmail.com	6xx9	27-08-2002	789xxxxx61	NIT	Project Manager (PM)	Survey analysts expedite the process of data analysis, allowing businesses to gain insights quickly. Their expertise in identifying patterns and trends from survey responses enables businesses to make data-driven decisions in real-time. This leads to faster adjustments in marketing strategies, product development, and customer engagement, boosting business agility.	50.68101883
Aniruddha Shelke	shelkeaniruddha7@gmail.com	7xx1	21-01-2003	906xxxxx59	NMIMS	Graduate Engineer Trainee (GET)	Survey data professionals assist businesses by providing timely insights derived from survey data analysis. Their ability to process and interpret large datasets quickly enables businesses to make fast, informed decisions, improving overall responsiveness to market trends, customer	44.40704584

Fig. 7.o

SUMMARY & CONCLUSION

This report presents a detailed AI detection framework designed to identify AI-generated responses in hiring assessments. By combining linguistic feature analysis, deep learning embeddings, clustering techniques, and autoencoders, we effectively differentiate human-written vs. AI-generated responses. The AI Response Classification Function is a powerful tool for ensuring fair and unbiased hiring by detecting AI-generated responses. By using linguistic analysis, deep learning models, and machine learning clustering, the system effectively differentiates between human-written and AI-assisted content. This prevents candidates from using AI to bypass skill assessments, ensuring that only genuine, qualified individuals are selected.

This model can be continuously improved by training on new AI-generated text patterns, allowing it to stay ahead of evolving AI models like GPT-4 and beyond. Future enhancements may include real-time detection, adaptive learning mechanisms, and expanded feature engineering for even higher accuracy.

REFERENCES

1. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Arxiv preprint arxiv:1810.04805.
2. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504-507.
3. Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, 281-297.
4. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 26.
5. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. Openai Blog.
6. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Arxiv preprint arxiv:1908.10084.