
**MSc program in Business Administration and Data Science**

Department of Digitalization

# PII Masking and Risk Assessment in Unstructured Text: An NLP-Based Approach

**Natural Language Processing and Text Analytics (CDSCO1002U)**

Examiner:
Rajani Singh

Students:
Eduard Aguado (176199)
Federico De Marinis(176194)
Noe Juarez (176695)
Marco Sburlino (176186)

Number of pages: 15
Word count: 4,231
Character count: 27,861
Submission date: 16-05-2025

# Abstract

This project explores the application of Natural Language Processing and Machine Learning models to classify privacy risk levels in text based on the presence and type of personally identifiable information (PII). The core problem addressed is the need for automated, scalable risk identification in textual data, supporting privacy compliance and data handling decisions. The research question centers on how accurately risk levels can be predicted from unstructured text using both traditional and transformer-based models. The methodology applies supervised classification, regularization, token-level annotation, and model evaluation. The analysis is based on a dataset of 37,853 English documents labeled with BIO-format PII tags, extracted from the *PII Masking 300K* dataset introduced by AI4Privacy (2023). After preprocessing and risk scoring, four models were trained and evaluated: Logistic Regression and Multinomial Naive Bayes with TF-IDF features, DistilBERT for contextual token classification, and GPT-3.5 via API for PII extraction. All models demonstrated stable generalization, with DistilBERT achieving the best overall performance. These results suggest that transformer-based models offer superior contextual understanding for PII detection and risk classification in text, particularly in high-dimensional or nuanced scenarios.



**Keywords:** Natural Language Processing; Privacy Risk Classification; PII Detection; TF-IDF; BERT; GPT -3.5


## List of Abbreviations

| Abbreviation | Definition |
|---|---|
| PII | Personally Identifiable Information |
| BIO | Beginning-Inside-Outside tagging scheme |
| NER | Named Entity Recognition |
| GDPR | General Data Protection Regulation |
| TF-IDF | Term Frequency–Inverse Document Frequency |
| BERT | Bidirectional Encoder Representations from Transofrmers |
| GPT | Generative Pre-training Transformer |
| API | Application Programming Interface |

# Contents

**1 Introduction** **1**

**2 Context and Related Work** **1**

**3 Use Case and Task Definition** **3**

3.1 Task Definition . . . . . 3

3.2 Use Case: Privacy Risk Mitigation in Customer Support Systems . . . . . 3

**4 Methodology** **4**

4.1 Dataset Description and Cleaning . . . . . 4

4.2 Exploratory Data Analysis (EDA) . . . . . 5

4.3 Preprocessing . . . . . 6

4.4 Model Selection . . . . . 7

4.4.1 TF-IDF . . . . . 7

4.4.2 BERT . . . . . 8

4.4.3 GPT -3.5 API . . . . . 9

4.5 Evaluation Techniques . . . . . 9

**5 Results** **10**

5.1 Model Performance . . . . . 10

5.2 Best Model . . . . . 11

**6 Discussion** **13**

**7 Conclusions and Future Work** **13**

# 1 Introduction

Cybersecurity is a growing concern as organizations increasingly rely on digital systems to store and process sensitive information. Cybercrime is projected to cause global damages exceeding $9 trillion in 2024, making it one of the most pressing global risks (World Economic Forum, 2023). The growing use of unstructured text such as emails, support tickets, and chat logs has further intensified the risk of Personally Identifiable Information (PII) exposure, raising serious privacy concerns, including risks of identity theft, financial loss, discrimination, and reputational harm. One example is the cyberattack on the UK's Legal Aid Agency, which compromised over two million sensitive records and highlighted the urgent need for more effective protection measures ("Fraud and Extortion Risk After Cyberattack on Legal Aid Agency", 2025).

To address this, automated PII masking techniques have gained importance. These methods aim to detect and redact sensitive information before it is stored or shared. Traditional rule-based approaches offer limited flexibility and often fail in multilingual or ambiguous contexts. Machine learning and Natural Language Processing methods, particularly Named Entity Recognition (NER), offer more adaptive and scalable solutions by learning to identify PII from labeled data.

This project investigates the use of machine learning and natural language processing models to classify documents by privacy risk based on the PII found in multilingual text. The main objectives are to train a sequence labeling model using BIO (Beginning-Inside-Outside) tagging for PII detection and to develop a document-level risk classifier using features derived from the detected entities. This study examines how accurately models can detect and assess the privacy risk of unstructured text containing PII.

# 2 Context and Related Work

Personally Identifiable Information (PII) refers to any data that can be used, alone or in combination with other information, to identify, contact, or locate an individual (U.S. Department of Labor, 2025). Common examples include names, social security numbers, email addresses, phone numbers, and biometric data. To mitigate risks associated with PII, the European Union has established legal frameworks for its collection and processing through the General Data Protection Regulation (GDPR). Such data may be processed only under specific conditions, including explicit consent or the necessity to protect the data subject's vital interests (European Parliament and Council, 2016). Transparent and lawful data handling is essential for accountability and for fostering user trust.

Despite these regulatory safeguards, PII masking techniques have become increasingly important. The work of Kulkarni Poornima and N. K. Cauvery addresses the challenges of detecting PII in large volumes of unstructured text, proposing a hybrid unsupervised model called C-PIIM to enhance PII

protection (Kulkarni & K, 2021). Their findings indicate that personal emails contain the highest concentration of PII, followed by work emails, with email headers containing more PII than message bodies due to metadata such as sender and recipient information. The proposed model outperforms traditional hierarchical clustering methods in clustering quality but is limited to detecting direct identifiers, excluding indirect ones.

Additionally, applications such as Morpheus developed by NVIDIA presented the potential in using NLP to detect different categories of sensitive information and assess real-time threat detection (credit card numbers, passwords, and user ID). Morpheus uses AI and GPU acceleration to inspect network traffic with minimal delay. Moreover, Mohammedi, A. explores several NLP-based text anonymization methods, including suppression, pseudonymization, noising, and generalization, with the purpose to protect PII and be GDPR-compliant. It emphasizes the use of Microsoft's open-source NLP tool, Presidio, for automated text redaction and anonymization. The research shows that combining these methods actually improves the PII protection, while preserving data usefulness (Mohammedi, 2023).

BIO tagging is a widely adopted scheme in natural language processing for identifying both the type and span of entities within text sequences. In the context of privacy protection, it is particularly effective for detecting PII in unstructured text, enabling accurate masking or redaction. The scheme assigns each token a label indicating its position within an entity: `B` (beginning) for the first token, `I` (inside) for subsequent tokens, and `O` (outside) for tokens not part of any entity.

| **Token** | **BIO Label** | **Masked Output** |
|---|---|---|
| My | O | My |
| name | O | name |
| is | O | is |
| John | B-PER | [NAME] |
| Doe | I-PER | [NAME] |
| and | O | and |
| my | O | my |
| email | O | email |
| is | O | is |
| john.doe | B-EMAIL | [EMAIL] |
| @ | I-EMAIL | [EMAIL] |
| gmail.com | I-EMAIL | [EMAIL] |
| . | O | . |

Table 1: Example of a BIO-tagged sentence and corresponding masked output.

Table 1 presents an example of a BIO-tagged sentence and its corresponding masked output. Each PII sub-token is labeled based on its position within the entity (beginning or inside) followed by the entity type. After labeling, a masked version of the sentence is generated, in which detected PII tokens are replaced with bracketed entity types, while non-PII tokens are preserved. This masked output is useful for producing anonymized text where PII is systematically redacted.

# 3 Use Case and Task Definition

## 3.1 Task Definition

This project aims to address two tasks. The first involves assigning a risk level, from 1 (low) to 3 (high), to each observation based on the frequency and sensitivity of the detected PII entities on the masked text. Weights were defined for each entity type according to its practical privacy relevance, with more sensitive entities such as passwords receiving higher weights than less critical ones like usernames. These weighted frequencies were aggregated to compute a cumulative score for each document, which was then mapped to a discrete risk category using defined thresholds.

The second task focuses on predicting the assigned risk level using the original version of the text. The objective is to train a model to infer privacy risk from anonymized content, simulating a real-world deployment scenario where raw data is inaccessible. Since the test set was labeled during the first task, it was used to evaluate the model's ability to predict risk levels accurately.

## 3.2 Use Case: Privacy Risk Mitigation in Customer Support Systems

Customer support platforms routinely handle vast volumes of unstructured textual data in the form of support tickets, chat transcripts, and complaint emails. These documents often contain sensitive user details such as names, addresses, phone numbers, IDs, or payment data, that must be handled with strict privacy safeguards to comply with data protection laws such as the General Data Protection Regulation (GDPR).

In this context, our system provides a practical solution: it automatically detects Personally Identifiable Information (PII) and classifies the associated document's privacy risk level as Low-Risk or High-Risk. This allows organizations to:

- Automatically redact or mask sensitive information before storage or sharing.
- Assign higher-risk cases to specialized privacy-aware workflows.
- Reduce legal exposure and improve customer trust by safeguarding data in real time.

This use case exemplifies how NLP can be applied to automated compliance, data minimization, and secure document handling in live enterprise settings.

# 4 Methodology

## 4.1 Dataset Description and Cleaning

The dataset used in this project is the *PII Masking 300K* dataset introduced by AI4Privacy (2023). It is designed for training and evaluating models in the task of detecting and masking Personally Identifiable Information (PII) in text. The dataset comprises 225,405 annotated text samples spanning six languages (English, French, German, Italian, Dutch, and Spanish), with localized content across eight jurisdictions. Each entry consists of synthetic or semi-synthetic text generated using proprietary algorithms, ensuring no privacy violations.

The dataset is divided into two sub-corpora: *OpenPII-220K*, which contains 27 general PII types such as names, emails, phone numbers, IDs, and passwords, and *FinPII-80K*, which includes approximately 20 additional types specific to financial and insurance domains. It comprises over 30 million tokens, with around 7.6 million labeled as PII. The annotations were validated through a human-in-the-loop process, achieving a token-level accuracy of 98.3% on a manually reviewed sample. A predefined training/test split of 78.8% and 21.2% is provided.

As seen in Figure 1, each instance contains the following fields: `source_text`, which represents the original unmasked input; `target_text`, which contains the PII-masked version of the text; and `privacy_mask` and `span_labels`, which specify the locations and categories of the identified PII spans. The field `mbert_text_tokens` provides the tokenized version of the input text, aligned with the multilingual BERT tokenizer, while `mbert_bio_labels` contains the corresponding BIO-format annotations used for sequence labeling. Each example also includes a unique `id` and a `language` tag to support multilingual training and evaluation.

| source_text | target_text | privacy_mask | span_labels | mbert_text_tokens | mbert_bio_labels | id | language | set |
|---|---|---|---|---|---|---|---|---|
| Subject: Group Messaging for Admissions Proces... | Subject: Group Messaging for Admissions Proces... | [{'value': 'wynqvrh053', 'start': 287, 'end': ... | [[440, 453, "USERNAME"], [430, 437, "TIME"], [... | [Sub, ##ject, :, Group, Mess, ##aging, for, Ad... | [O, O, O, O, O, O, O, O, O, O, O, O, O, O, ... | 40767A | English | train |
| - Meeting at 2:33 PM\n- N23 - Meeting at 11:29... | - Meeting at [TIME]\n- [USERNAME] - Meeting at... | [{'value': '2:33 PM', 'start': 13, 'end': 20, ... | [[74, 81, "TIME"], [50, 60, "USERNAME"], [40, ... | [-, Meeting, at, 2, :, 33, PM, -, N, ##23, -, ... | [O, O, O, B-TIME, I-TIME, I-TIME, I-TIME, O, O... | 40767B | English | train |
| Subject: Admission Notification - Great Britai... | Subject: Admission Notification - Great Britai... | [{'value': '5:24am', 'start': 263, 'end': 269,... | [[395, 407, "SOCIALNUMBER"], [358, 375, "EMAIL... | [Sub, ##ject, :, Ad, ##mission, Not, ##ificati... | [O, O, O, O, O, O, O, O, O, O, O, O, O, O, ... | 40768A | English | train |
| Card: KB90324ER\n Country: GB\n Building: ... | Card: [IDCARD]\n Country: [COUNTRY]\n Buil... | [{'value': 'KB90324ER', 'start': 6, 'end': 15,... | [[390, 393, "STATE"], [368, 378, "CITY"], [346... | [Card, :, KB, ##90, ##32, ##4, ##ER, \, n, Cou... | [O, O, B-IDCARD, I-IDCARD, I-IDCARD, I-IDCARD,... | 40768B | English | train |
| N, WA14 5RW\n Password: r}iD1#8\n\n...and so... | N, WA14 5RW\n Password: [PASS]\n\n...and so ... | [{'value': 'r}iD1#8', 'start': 26, 'end': 33, ... | [[336, 352, "DATE"], [26, 33, "PASS"]] | [N, ,, W, ##A, ##14, 5, ##R, ##W, \, n, Pass, ... | [O, O, O, O, O, O, O, O, O, O, O, O, O, B-PASS... | 40768C | English | train |

Figure 1: Example of the first five rows in the dataset

Although the raw data initially appeared clean, a sanity check was performed to verify the absence of missing values and exact duplicates. To conduct this, the training and test datasets were merged and examined, confirming that no such issues were present. A second check was then carried out to detect empty strings in the `source_text` field, revealing 25 instances, which were subsequently removed. Following these steps, the dataset was deemed fully cleaned and ready for preprocessing.

## 4.2 Exploratory Data Analysis (EDA)

First, to understand the structure and composition of our data, an initial exploratory analysis was conducted. Since the dataset contains text in multiple languages, the language distribution was examined to identify the most predominant language and assess balance across categories. While the overall distribution was relatively even, the models were restricted to English-language texts to ensure consistency and reduce complexity during preprocessing and modeling.

Secondly, to analyze the structure and distribution of PII types, each category was plotted to assess its frequency. As shown in Figure 2, the most common entities were `time`, `username`, and `email`, which are generally considered lower risk. In the mid-frequency range (7,500 to 10,000 occurrences), 17 other PII types appear, including higher-risk entities such as `passport` and `IP`. The least frequent PII categories were typically the most sensitive, such as `password` and `cardissuer`.

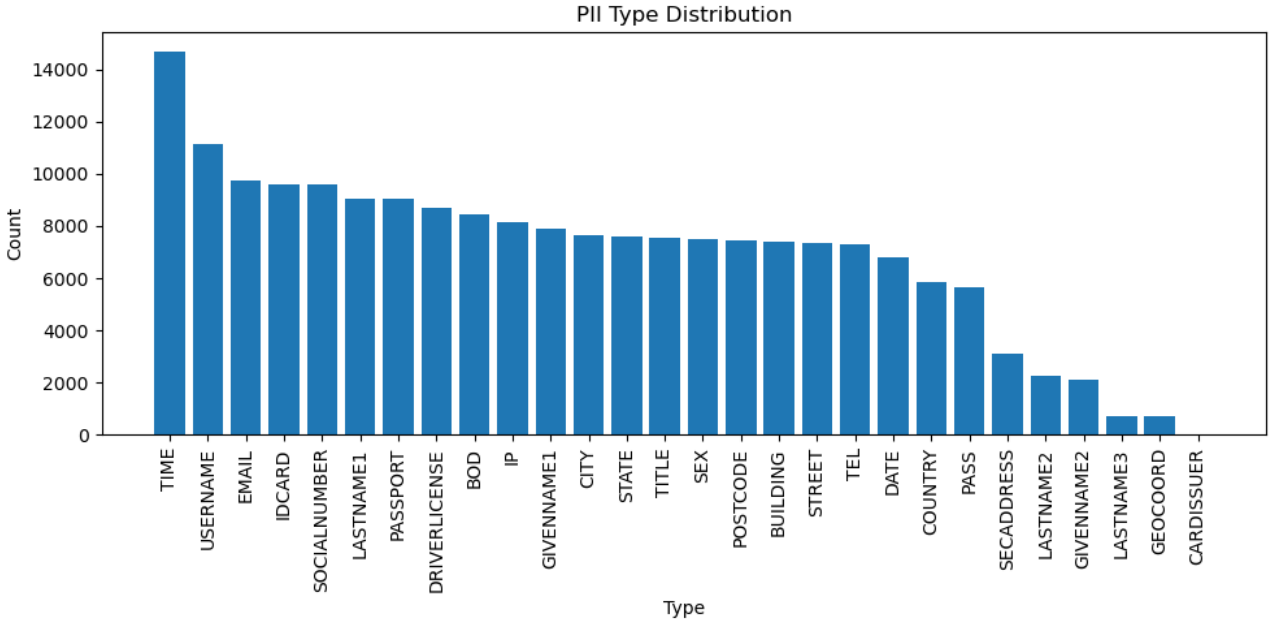


Figure 2: PII Type Distribution

Finally, to assess PII density at the document level, a histogram was plotted showing the number of PII entities per document. As illustrated in Figure 3, the distribution is right-skewed, with most documents containing fewer than 10 PII instances and a long tail extending up to 35 entities.

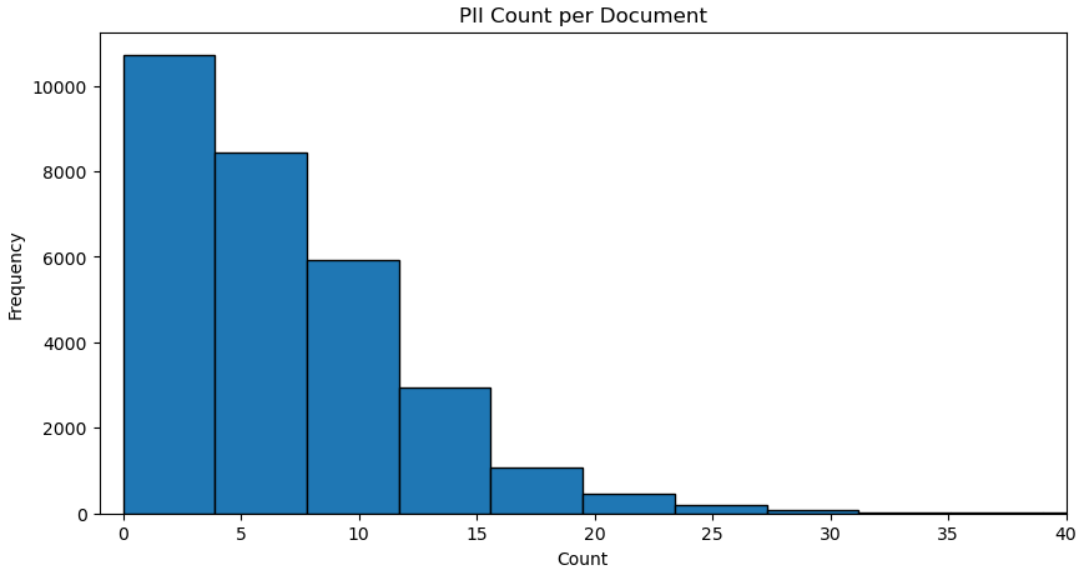


Figure 3: PII Count per Document

## 4.3 Preprocessing

Accurate tokenization and label alignment are essential for sequence tagging tasks (Lample et al., 2016), and the NER (Named Entity Recognition) pipeline provides a structured approach to achieve this. First, the raw text was cleaned to ensure consistency in the corpus and prepare it for modeling by applying a series of standard NLP preprocessing steps such as lowercase, trimming, and normalizing punctuation. After cleaning the document, simple whitespace splitting was applied and common English stopwords were removed to focus on more relevant words.

Furthermore, to properly assigned the importance of each PII in the document, a weight was assigned for each type according to its risk level and analyzed its frequency through a log-scaled inverse frequency formula. The range goes from 1 to 3, 3 being the highest risk. This ensures that a text containing password and IP addresses contributes higher to the document's overall risk level than a one containing only email addresses and names. The weights are used to compute a numeric risk score for each document by summing the weighted PII counts.

As a final step in the preprocessing, each document's risk score is turned into a category labeled either "High" or "Low", reflecting the privacy risk they carry and serving as targets for the supervised classification performed afterwards. To ensure class balance, threshold cut-off at the 50th percentile was computed based on whether the document's risk score falls above or below the median. The resulting class distributions for both training and test sets are shown in Figure 4

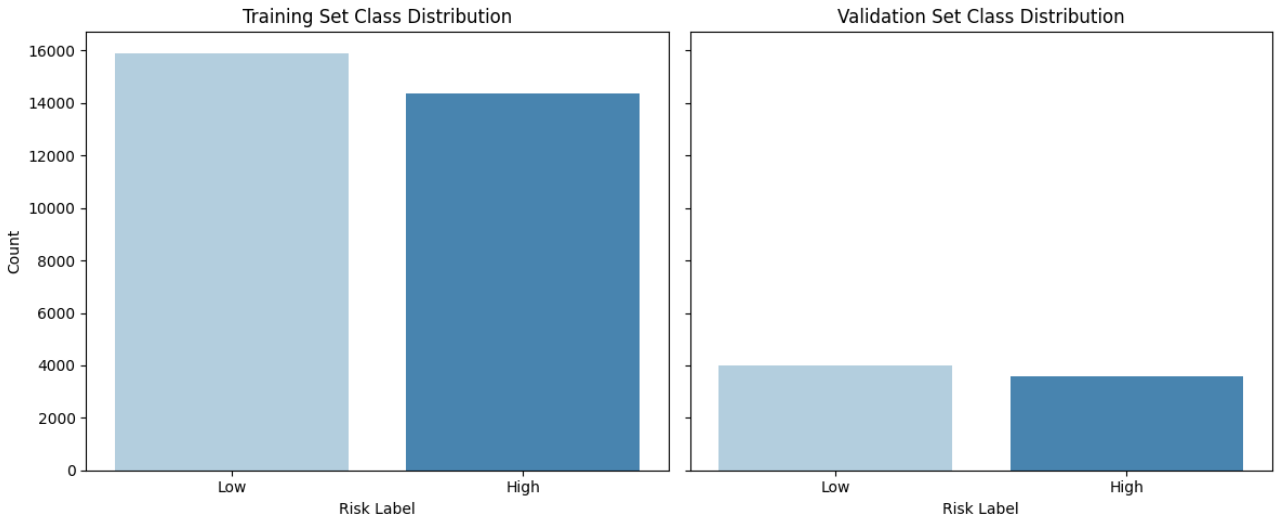


Figure 4: PII Count per Document

Once the dataset has been preprocessed and risk labels assigned, an additional step is required to prepare it for the second task, which involves predicting risk levels on test data using the non-masked text. Specifically, tokens must be lemmatized to reduce each word to its base form (e.g., ”running” becomes ”run”) and then converted into clean, space-separated strings. To do so, the small English `spaCy` model was used. This process produces a more compact and consistent vocabulary, enabling more effective model training and improving classification performance (Manning et al., 2008).

## 4.4 Model Selection

After cleaning and preprocessing the dataset, the next step was to select, train, and evaluate classification models. Three approaches were applied. First, two binary classification models using TF-IDF tokenization were implemented: Logistic Regression and Multinomial Naive Bayes. Second, a pre-trained transformer-based language model (BERT) was used to enhance the model's contextual understanding. Finally, the OpenAI GPT-3.5 API was employed to identify PII in the text, and its performance was compared to the previous models.

### 4.4.1 TF-IDF

For the first modeling approach, TF-IDF (Term Frequency–Inverse Document Frequency) was chosen, as it evaluates the importance of a word in a document relative to the entire corpus, unlike the Bag-of-Words model (Manning et al., 2008). This is essential for risk identification, as it reduces the influence of generic language while emphasizing terms that may indicate the presence of sensitive information. `TfidfVectorizer` was used to extract features with a specific configuration. The `ngram_range` was set to (1, 3) to capture unigrams, bigrams, and trigrams, modeling both individual

terms and short sequences. `max_df=0.9` was used to remove very common terms appearing in over 90% of documents. `max_features` was set to 5,000 to retain the most informative terms and reduce dimensionality. Finally, `sublinear_tf=True` applied logarithmic scaling to mitigate the dominance of high-frequency terms.

With this setup, logistic regression was selected as the first classification model. A machine learning pipeline was implemented to integrate the TF-IDF feature extraction step with the logistic regression classifier. The model was trained on the lemmatized text, with TF-IDF converting the input into weighted feature vectors, and logistic regression used to predict the risk level. Finally, L1 regularization was applied to promote sparsity and reduce overfitting, which is particularly important when working with high-dimensional text data. The second model using the TF-IDF approach was Multinomial Naive Bayes (MNB), a probabilistic classifier that assumes all features (e.g., word occurrences) are conditionally independent given the class (Manning et al., 2008). The model was configured with a small smoothing parameter ($\alpha = 0.01$) to increase sensitivity to subtle differences in word frequency.

### 4.4.2 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a language model developed by Google that processes input-text contextually (Devlin et al., 2019). It captures the meaning of each token based on its surrounding context, for example, distinguishing between “bank” in “river bank” and “bank account.” This is achieved through its bidirectional architecture, which considers both left and right context simultaneously, making BERT particularly effective for classification tasks. As a result, fine tuned BERT models often outperform traditional approaches such as logistic regression or Naive Bayes (Devlin et al., 2019).

DistilBERT was selected as the model for the BERT-based approach. As Sanh et al. (2019) describes, DistilBERT is a compact version of BERT that retains most of its performance while being significantly smaller and faster. The objective of the model is to assign BIO labels to individual tokens, rather than to classify entire documents. After token-level prediction, the ”B-” and ”I-” prefixes are stripped from the PII entities, and risk scores are computed based on predefined risk weights. These scores are then converted into ”Low” or ”High” risk categories, enabling document-level classification.

To set up the model, the input text was first tokenized and aligned with the original BIO labels, as DistilBERT (like all BERT-based models) operates on subword tokens and requires token-level supervision. Second, the `TrainingArguments` class from the `transformers` library was used to define the training configuration. These settings included evaluation at the end of each epoch, a low learning rate with logging every 50 steps to monitor training progress, and a `weight_decay` of 0.01, serving as regularization to prevent overfitting. Finally, the `Trainer` class, also from the

`transformers` library, was initialized to handle the training loop, model evaluation, and checkpoint management.

### 4.4.3 GPT -3.5 API

To evaluate the potential of a Large Language Model for the task of PII detection and privacy risk assessment, the GPT-3.5 model developed by OpenAI was integrated into the pipeline. The rationale behind this choice was based on the model's ability to interpret natural language prompts in a zero-shot setting and produce structured responses. This makes it particularly suitable for identifying entities in text without the need for supervised fine-tuning.

Among the available large language models, GPT-3.5 was selected due to its reliable performance in comparable use cases, the availability of extensive documentation, and ease of access through the OpenAI API. Compared to more advanced models such as GPT-4, GPT-3.5 represents a cost-effective alternative with sufficient capabilities for the needs of this project[1]. At the time of use, the API usage involved a small monetary cost, which was considered acceptable within the scope of the assignment.

To access the model, an OpenAI account was created and an API key was configured in the environment. The implementation relied on the use of the official Python package provided by OpenAI, which enabled programmatic access to the model as part of the processing pipeline.

Each document contained in the test set, was submitted to the API through a prompt requesting the extraction of all PII types present in the text. The model was instructed to return the output in the form of a JSON dictionary, where each key represented a PII type (such as EMAIL, PASSWORD, or USERNAME), and the corresponding value indicated the number of times it occurred in the input. These PII types were consistent with those used in the earlier rule-based counting phase.

This approach enabled the extraction of PII information in a structured format that was directly compatible with the existing risk computation framework. Once the model returned the count of each PII type per document, these counts were used as inputs to the predefined risk scoring formula, which applied specific weights to each PII category based on its sensitivity. The resulting numeric score was then used to assign the binary risk label – either high or low – following the same thresholding strategy applied in the rule-based method. This allowed for a direct comparison of the LLM-driven classification results with those obtained from the traditional pipeline.

## 4.5 Evaluation Techniques

Aligned with the use case outlined in Section 3, evaluation metrics were selected to prioritize the accurate identification of high-risk text. For the token-level NER task, particular emphasis was placed

[1] As of May 2025, the API pricing for GPT-3.5 is $0.50 per 1M input tokens and $1.50 per 1M output tokens.

on minimizing false negatives, where text containing sensitive PII is not labeled as risky. Therefore, recall was chosen as the primary evaluation metric, as performance in this context depends more on minimizing false negatives than on avoiding false positives. Recall is defined as follows:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{1}$$

Given the trade-off between precision and recall, where increasing recall may lead to more false positives, the $F_1$-score was used to monitor the balance between these metrics. Although the primary focus is on minimizing false negatives, it is also important to control false positives, particularly in the document-level classification task, where precision helps reduce unnecessary escalations while ensuring that sensitive cases are correctly identified. Lastly, accuracy was included to assess the overall correctness of the model's predictions. These metrics are defined as follows:

$$F_1 = \frac{2}{\text{Recall}^{-1} + \text{Precision}^{-1}} \tag{2}$$

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}} \tag{3}$$

# 5 Results

## 5.1 Model Performance

Table 2 summarizes the performance of four models across four key metrics. The DistilBERT model achieved the highest Recall (0.92), clearly outperforming the others. It demonstrates a strong ability to correctly identify positive cases while also maintaining high overall accuracy (0.89), making it the most effective model to minimize false negatives in comparison.

Logistic Regression and Multinomial Naive Bayes, both using TF-IDF features, showed consistent and well-balanced results, with scores of 0.85 and 0.83, respectively, across evaluation metrics. These results indicate strong baseline performance, with Logistic Regression slightly outperforming Naive Bayes across all metrics, maintaining an improved rate to identify both false positives and false negatives.

The OpenAI GPT model recorded the lowest performance overall, with a recall and $F_1$-score of 0.77 and an accuracy of 0.78. While its precision is relatively high at 0.82, the model underperforms in

capturing positive cases, which is critical when prioritizing sensitive in high-risk classifications.

| **Model** | **Recall** | **$F_1$-score** | **Accuracy** | **Precision** |
|---|---|---|---|---|
| Logistic Regression (TF-IDF) | 0.85 | 0.85 | 0.85 | 0.85 |
| Multinomial Naive Bayes (TF-IDF) | 0.83 | 0.83 | 0.83 | 0.83 |
| DistilBERT (BERT) | 0.92 | 0.87 | 0.89 | 0.84 |
| OpenAI GPT -3.5 API | 0,77 | 0,77 | 0.78 | 0,82 |

Table 2: Overall evaluation metrics by model

In addition to performance metrics, it was also evaluated the computational efficiency of each model by measuring total execution time. The traditional TF-IDF models demonstrated the fastest runtimes, with Logistic Regression completing in 27.2 seconds and Naive Bayes in 43.7 seconds. These results are expected, as both models operate on sparse vectorized inputs without any contextual embedding or external dependencies.

DistilBERT, while achieving the best predictive performance, required substantially more time, completing in 4,832.9 seconds. This increased runtime is attributable to the computational demands of transformer-based architectures, which process inputs using multi-head attention mechanisms across multiple layers. Despite this, DistilBERT remains the best model for deployment in time-sensitive environments due to its balance of accuracy and efficiency.

In contrast, the GPT-3.5 API approach took 22,071.1 seconds – over 4.5 times longer than BERT. This extended runtime is largely due to the API-based setup, where latency arises from network requests, external processing on OpenAI's servers, and serialization of results. Furthermore, the model processes each document individually in a conversational interface, contributing to the slower overall throughput. While informative for comparison purposes, the LLM-based method may not currently be suitable for large-scale, real-time deployment scenarios due to this processing overhead.

## 5.2 Best Model

As discussed previously, DistilBERT demonstrated the strongest overall performance among the four models, achieving the highest scores across all key evaluation metrics. Table 3 presents a detailed breakdown of its performance by class. The model performs particularly well on high-risk cases, with a precision of 0.99 and a recall of 0.87, resulting in an $F_1$-score of 0.92. This indicates that DistilBERT is highly effective at correctly identifying high-risk instances with minimal false positives, which aligns with the evaluation priorities outlined in Section 4.5. In contrast, performance on low-risk cases reflects a trade-off: while recall is very high (0.98), precision decreases to 0.70, yielding a lower $F_1$-score of 0.81. As illustrated in the confusion matrix in Figure 5, this reduction in precision is attributable to a higher number of false positives in the low-risk class, likely influenced by the model's emphasis on maximizing sensitivity to high-risk instances.

| | Precision | Recall | $F_1$-score | Support |
|---|---|---|---|---|
| **High-Risk (1)** | 0.99 | 0.87 | 0.92 | 5773 |
| **Low-Risk (0)** | 0.70 | 0.98 | 0.81 | 1798 |
| Accuracy | - | - | 0.89 | 7571 |
| Macro avg | 0.84 | 0.92 | 0.87 | 7571 |
| Weighted avg | 0.92 | 0.89 | 0.90 | 7571 |

Table 3: DistilBERT Model Classification Metrics

The test set contains 7,571 records, of which 5,773 are labeled as high-risk and 1,798 as low-risk. This class imbalance may influence the model's performance, particularly by reducing precision for the minority class. The confusion matrix in Figure 5 illustrates the model's predictions compared to the actual labels, revealing a moderate false negative rate. It correctly classifies 5,022 high-risk instances but also shows a notable number of false negatives (751), where high-risk cases were misclassified as low-risk. Conversely, the model performs exceptionally well on low-risk classifications, correctly identifying 1,763 instances while producing only 35 false positives. These outcomes are consistent with the classification report in Table 3, where the recall for low-risk cases is nearly perfect (0.98), while the recall for high-risk cases is slightly lower at 0.87.

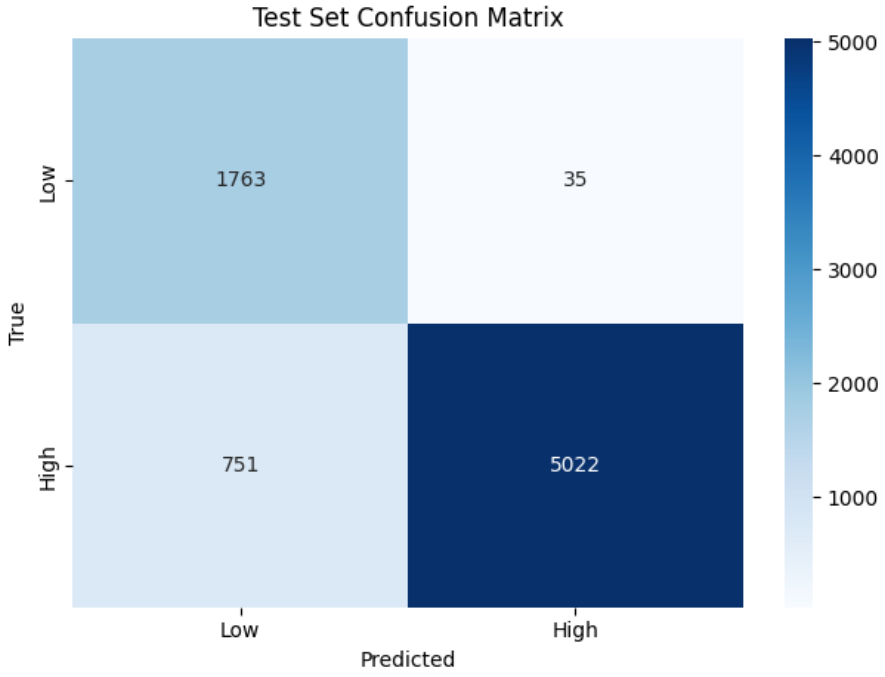


Figure 5: Confusion Matrix for DistilBERT Model

# 6 Discussion

Overfitting was a primary concern and was addressed by evaluating classification performance on both training and test sets. As shown in Table 4 in the Appendix, there is no significant variance across performance metrics for any of the models. To mitigate overfitting, regularization techniques were applied. For example, since logistic regression is prone to overfitting in high-dimensional text settings, L1 regularization (Lasso) was used to penalize large coefficients and improve generalization.

Throughout this study, several limitations were encountered that could impact the accuracy and generalizability of the results. First, for privacy reasons, the dataset consisted of synthetic data generated using proprietary algorithms, rather than real-world data. Although the data was manually reviewed and achieved a validation accuracy of approximately 98.3%, its synthetic nature may limit its applicability to real-world scenarios.

Additional challenges were observed in the TF-IDF-based models and the GPT-3.5 API model. The TF-IDF models (Logistic regression and Multinomial Naive Bayes) do not capture semantic or sequential relationships between words, so these models rely on word frequency, not meaning or context. As for the GPT-3.5 API, its integration posed practical limitations: the model operates as a black box, lacks fine-tuning capabilities, and incurs latency and cost constraints. Moreover, its responses may vary across calls, introducing potential inconsistencies in the results

# 7 Conclusions and Future Work

This study evaluated four models for binary classification of privacy risk in text. Logistic Regression and Multinomial Naive Bayes were implemented using TF-IDF features, while DistilBERT was used for contextual token classification. These were compared with the GPT 3.5 API used for external PII extraction. As explained in Section 5.1, while the TF-IDF-based models offered competitive performance and interpretability, DistilBERT emerged as the best-performing approach, achieving the highest values across most evaluation metrics (see Table 4 in the Appendix), likely due to its ability to capture bidirectional context in high-dimensional text. These results suggest that transformer-based models are better suited for nuanced PII detection and risk classification tasks, especially when semantic context plays a key role.

Regarding future work, given the strong performance achieved by the models, it is reasonable to extend the project to more practical applications. One potential direction is to deploy the risk classification pipeline as an API integrated into corporate platforms like email or Microsoft Teams. The API would process raw text input, classify PII using the trained model, and return a risk label, triggering actions such as anonymization, encryption, or redaction. This would broaden the applicability of the classification models and contribute to enhancing corporate privacy compliance.

# References


AI4Privacy. (2023). Pii masking 300k dataset [Accessed 21 May 2025]. https://huggingface.co/datasets/ai4privacy/pii-masking-300k

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186. https://aclanthology.org/N19-1423

European Parliament and Council. (2016). Regulation (eu) 2016/679 (general data protection regulation) [Accessed 22 May 2025].

Fraud and extortion risk after cyberattack on legal aid agency [Accessed 21 May 2025]. (2025). *The Times*. https://www.thetimes.co.uk/article/legal-aid-agency-cyberattack-pii-breach

Kulkarni, P., & K, C. N. (2021). Personally identifiable information (pii) detection using machine learning and rule-based techniques [Accessed 24 May 2025]. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *12*(9), 470–476. https://thesai.org/Downloads/Volume12No9/Paper_57-Personally_Identifiable_Information_PII_Detection.pdf

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *Proceedings of NAACL-HLT*, 260–270.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press. https://nlp.stanford.edu/IR-book/

Mohammedi, M. (2023). *Anonymizing text with nlp: Strategies and techniques for ensuring data privacy* [Accessed 24 May 2025]. https://medium.com/@manou.mohammedi/anonymizing-text-with-nlp-strategies-and-techniques-for-ensuring-data-privacy-598a99598b3a

OpenAI. (2025a). Gpt-3.5 turbo model details [Accessed 21 May 2025]. https://platform.openai.com/docs/models/gpt-3.5-turbo

OpenAI. (2025b). Openai api models overview [Accessed 21 May 2025]. https://platform.openai.com/docs/models

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*. https://arxiv.org/abs/1910.01108

U.S. Department of Labor. (2025). Protection of personally identifiable information (pii) [Accessed 22 May 2025].

World Economic Forum. (2023). *The global risks report 2023: 18th edition*. World Economic Forum. https://www.weforum.org/reports/global-risks-report-2023

# Appendix

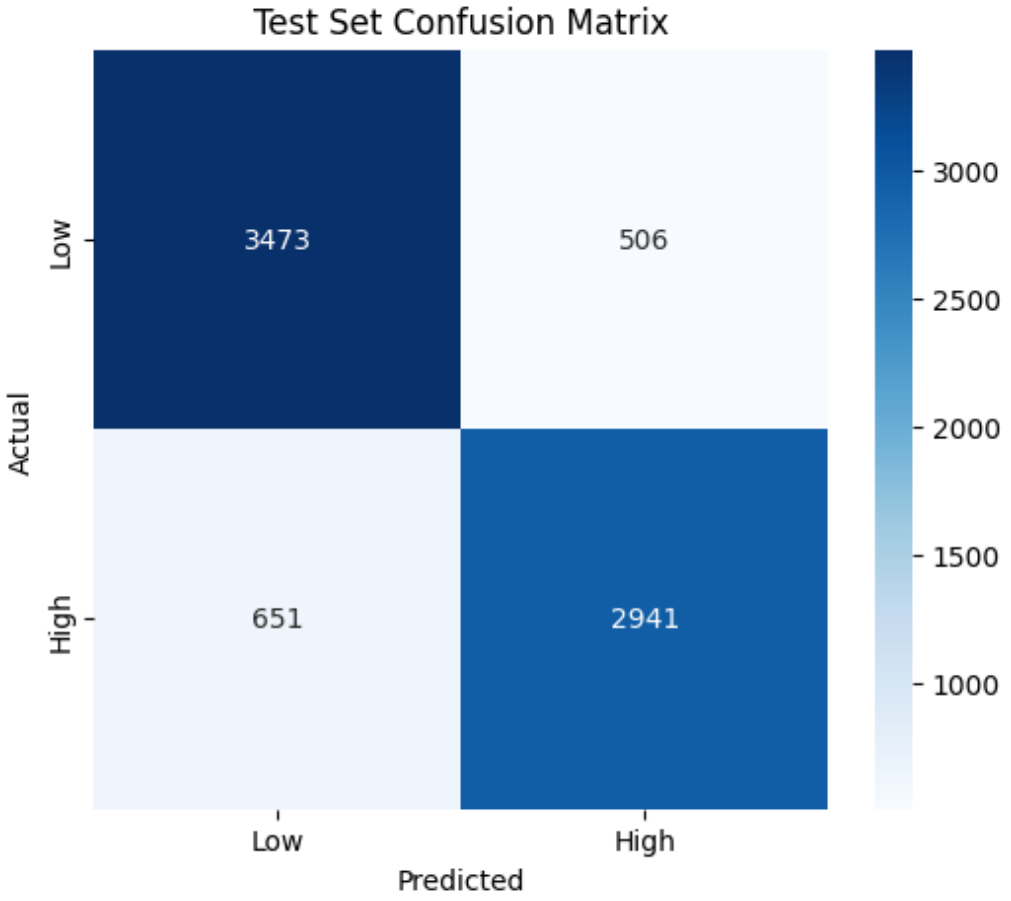


Figure 6: Confusion Matrix for Logistic Regression Model

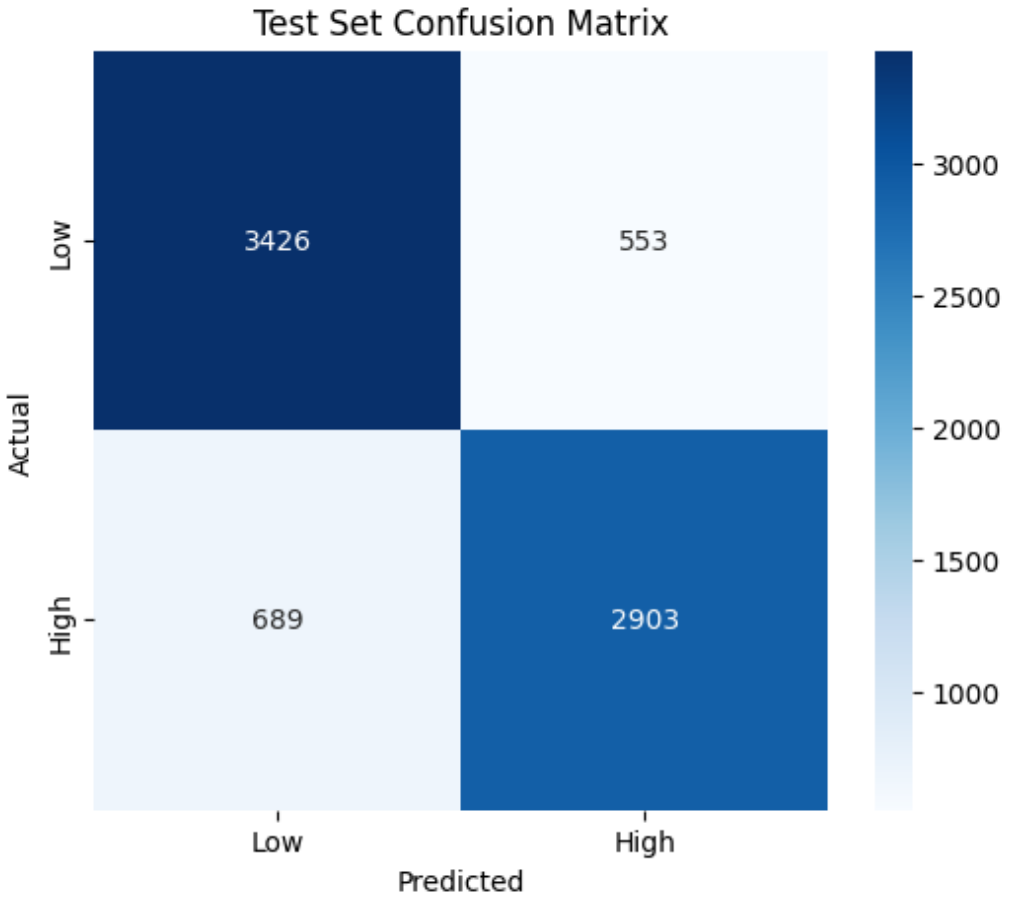


Figure 7: Confusion Matrix for Multinomial Naive Bayes Model

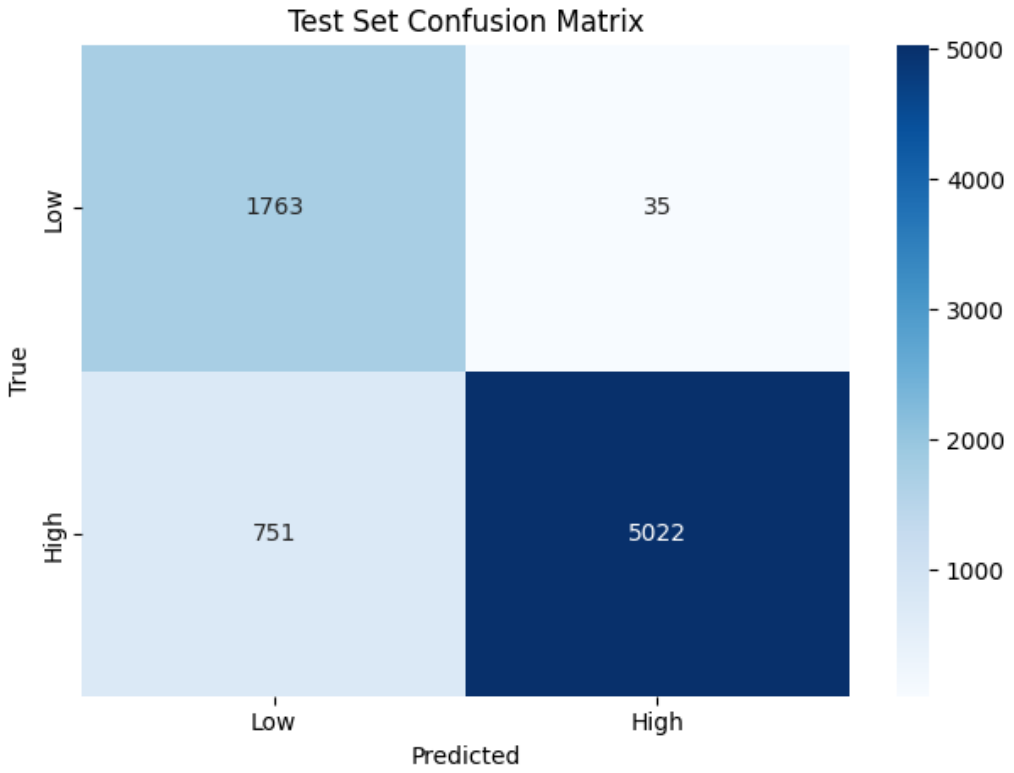


Figure 8: Confusion Matrix for DistilBERT Model

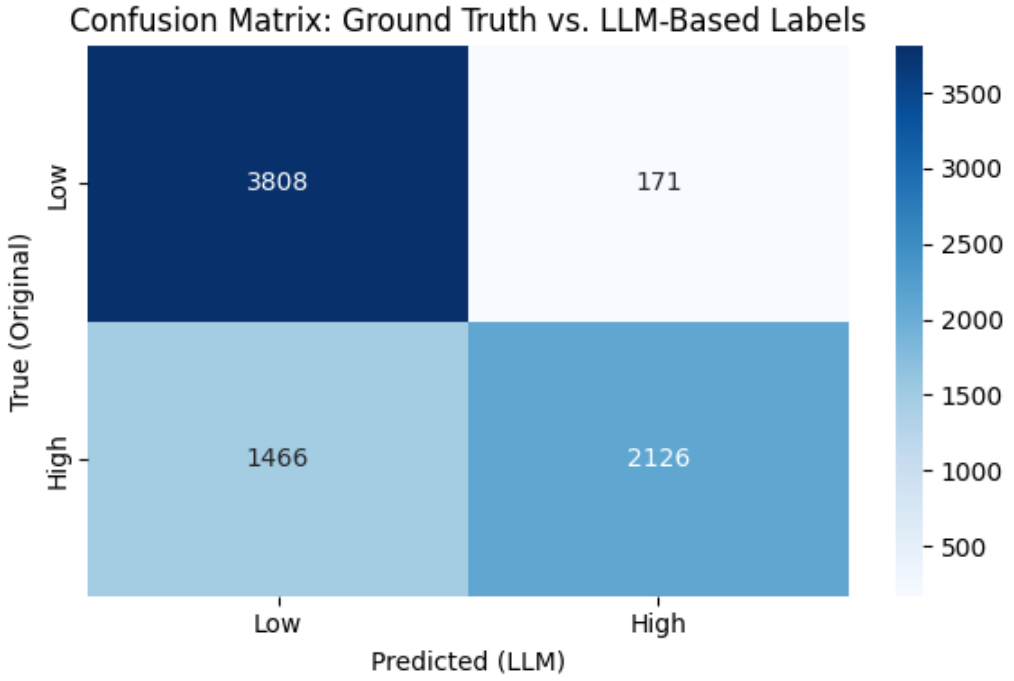


Figure 9: Confusion Matrix for GPT -3.5 API Model

| **Metric** | **Logistic Regression** | | **Naive Bayes** | | **DistilBERT** | |
|---|---|---|---|---|---|---|
| | Train | Test | Train | Test | Train | Test |
| Accuracy | 0.85 | 0.85 | 0.83 | 0.83 | 0.90 | 0.89 |
| Precision | 0.85 | 0.85 | 0.83 | 0.83 | 0.85 | 0.84 |
| Recall | 0.85 | 0.85 | 0.83 | 0.83 | 0.92 | 0.92 |
| F1 Score | 0.85 | 0.85 | 0.83 | 0.83 | 0.87 | 0.87 |

Table 4: Evaluation metrics on training and test sets for each model