## 1. Introduction)

**Project Overview**:
The sheer volume of academic paper submissions has created a bottleneck in the peer-review process, making it difficult for reviewers to evaluate papers efficiently. In many cases, the traditional methods for reviewing papers, which involve manual reading and analysis, are time-consuming and subjective. This project aims to develop an automated solution to tackle these challenges by classifying academic papers into categories such as "publishable" and "not publishable," suggesting appropriate conferences for submission, and generating an explanation for the decision (rationale). In addition, this system provides an automatic summarization of the abstract in one or two lines to give a quick overview of the paper's content.

**Motivation**:
The growth of global research has significantly increased the number of academic papers submitted to conferences and journals. Traditional review processes are not equipped to handle this ever-increasing volume, and researchers often have to wait weeks or even months to receive feedback on their submissions. By automating the classification process, researchers can get quicker feedback, and reviewers can focus on more critical tasks, thus enhancing the overall efficiency of the academic publication process. The motivation behind this project lies in the ability to use AI and NLP techniques to streamline and automate these processes, reducing the burden on academic reviewers and improving the speed and accuracy of paper evaluation.

In addition, the project is motivated by the need for transparency in decision-making. In academic publishing, the reasons behind the acceptance or rejection of a paper can often seem opaque. By generating rationales for classification decisions, this system provides clarity on why a paper was classified in a particular way, which can help authors improve their work.

**Project Goals**:

- **Publishability Classification**: The system should be able to predict whether a paper is ready for publication based on the content. The decision will be based on multiple factors, such as the relevance of the research, the strength of the methodology, and the quality of the abstract.
- **Conference Classification**: The model will classify the paper into one or more relevant conferences based on the content, specifically the terms and topics mentioned in the abstract.
- **Rationale Generation**: Each classification prediction will be supported by a rationale that explains the reasoning behind the classification. This will help ensure that the decision-making process is transparent and can be understood by the researchers and reviewers.
- **Abstract Summarization**: The system will automatically summarize the abstract of the paper, retaining its core information in one or two sentences. This is intended to provide a quick summary of the paper, making it easier for reviewers and researchers to get an overview without having to read the entire abstract.

## 2. Literature Review

**Existing Techniques in Paper Classification**:
Over the years, academic paper classification has been tackled using a variety of machine learning techniques. Early approaches relied heavily on traditional machine learning models, such as Naive Bayes, Support Vector Machines (SVM), and Random Forest, which typically use feature extraction methods like bag-of-words or TF-IDF (Term Frequency-Inverse Document Frequency) to represent documents as vectors of numerical values. These methods often worked well in structured domains but struggled with the complexity of academic text, where nuances, jargon, and multi-disciplinary terminology could make classification difficult.

Recent advancements in deep learning, particularly with the rise of neural networks, have led to more sophisticated techniques for paper classification. Models like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) have been used in text classification tasks, but more recently, Transformer-based models like BERT and GPT have demonstrated state-of-the-art performance on a variety of NLP tasks. BERT, in particular, is designed to understand context and relationships between words in a sentence, making it more effective in tasks that require understanding of the full meaning of a document rather than isolated word occurrences.

One notable area of improvement in NLP tasks is the development of **pre-trained models** like BERT and DistilBERT, which are trained on massive datasets and can be fine-tuned on specific tasks like paper classification. These models require less training data and computational resources than traditional machine learning models, making them more accessible for academic applications. However, the challenge remains in how to interpret the decisions made by these complex models, as they often act as “black boxes.”

**Challenges in Traditional Approaches**:
Traditional classification methods like Naive Bayes or SVMs rely on a hand-crafted set of features. These features are usually derived from the paper's metadata (title, authors, abstract, etc.), but they may not capture the depth and richness of the text's meaning. For example, the presence of highly technical jargon or domain-specific language in an academic paper might lead to misclassification when relying solely on basic feature extraction methods. Furthermore, even though these models can achieve reasonable performance, they are often unable to explain their reasoning, which can be problematic in academic settings where transparency is crucial.

Another issue with traditional methods is the **feature engineering process**, which requires significant manual effort to identify relevant features. This process is time-consuming and may not always capture the most useful aspects of the text. Deep learning methods, on the other hand, can automatically extract relevant features during training, allowing them to adapt to the underlying patterns in the data.

**Deep Learning Advancements**:
With the advent of Transformer models, such as BERT and DistilBERT, there has been significant progress in the ability to understand and process text in a more context-aware manner. BERT uses a bidirectional approach to text representation, meaning it looks at both the left and right context of a word when generating its representation. This is particularly useful in academic papers, where the meaning of words can vary depending on their surrounding context.

These models, however, still face challenges in **interpretability**. While they perform well on classification tasks, explaining why a model made a particular decision is not always straightforward. This is where research into **explainable AI (XAI)** comes into play. By developing techniques to generate transparent and understandable explanations for the model's predictions, researchers can improve trust in the system and make it easier for users to understand the rationale behind a classification.

---

## 3. Dataset Description

**Dataset Overview**:
The dataset used in this project consists of academic papers from multiple domains, including computer science, engineering, and life sciences. Each paper is associated with metadata such as the title, authors, abstract, and conference label. The conference label indicates the target conference to which the paper is suitable for submission. This dataset is diverse, with papers ranging from technical research on artificial intelligence to medical studies on disease treatment.

The dataset was curated to ensure that the papers represent a broad spectrum of academic topics and writing styles, which allows the model to learn patterns across different fields. Each paper in the dataset was labeled by academic experts who provided their evaluation on whether the paper is publishable or not, and if so, which conference it should be submitted to. These labels are crucial for training the model and ensuring that it can make accurate predictions.

**Preprocessing Pipeline**:
The first step in preprocessing the dataset was text extraction. Most academic papers are stored in PDF format, so the raw text needed to be extracted from these files. The PyMuPDF library was used for this purpose, as it allows for parsing PDFs while preserving the structure of the text, including headings, subheadings, and sections. This ensures that the text is correctly formatted for further processing.

After extraction, the text went through a series of cleaning steps. These steps removed unnecessary information like page numbers, footnotes, and figures, which are not relevant to the content of the paper. Additionally, common stopwords (e.g., "the," "and," "is") were removed, as they do not contribute to the meaningful analysis of the text. The remaining text was tokenized—split into smaller units called tokens, which are the building blocks of the model.

To prepare the data for use with a deep learning model, the text was normalized. This included converting the text to lowercase, removing punctuation, and stemming words (reducing them to their root form). Text normalization helps the model focus on the core meaning of the words rather than being distracted by different word forms.

**Embedding Extraction**:
To convert the raw text into a numerical format that the model can understand, the text was passed through a pre-trained **DistilBERT** model. DistilBERT generates **embeddings**, which are dense vector representations of words and sentences that capture their meaning. The embeddings produced by DistilBERT are context-aware, meaning that the model understands how the meaning of words changes depending on their surrounding context.

These embeddings serve as the input features for the classification model. DistilBERT has the advantage of being computationally efficient while still retaining the power of BERT's architecture. The embeddings capture the semantic meaning of the paper's content, which is essential for making accurate classifications and generating meaningful rationales.

**Labeling and Annotation**:
In order to label the data, a team of academic experts provided classifications for each paper in the dataset. These classifications included whether the paper was suitable for publication or not and which conference it would be most relevant to. This expert input is crucial, as it serves as the ground truth for training the model. Without accurate and consistent labeling, the model would not be able to make reliable predictions. The annotations were performed manually to ensure high-quality labels, although in the future, this process could potentially be automated using pre-trained models or crowd-sourcing platforms.

---

## 4. Methodology

**Model Architecture**:
The primary model architecture used for this project is based on the **DistilBERT** transformer model, which is a smaller, faster version of BERT. DistilBERT was chosen because of its efficiency in terms of computational resources while maintaining high accuracy. The model leverages the transformer-based architecture that uses **self-attention mechanisms**, enabling it to understand the relationships between different words in a sentence. This is particularly important in the context of academic papers, where the meaning of terms often depends on their context within the paper.

The architecture consists of three main components:

1. **Text Preprocessing Layer**: In this layer, the text is tokenized, normalized, and embedded using the pre-trained DistilBERT model. The model processes the input text by converting it into embeddings that capture the semantic meaning of the text.
2. **Classification Layer**: After generating embeddings, these are passed to a classification layer that assigns the paper to one of the predefined categories, such as "publishable" or "not publishable." The classification layer is typically a **fully connected neural network**

with a softmax activation function, which outputs a probability distribution over the possible categories.

3. **Rationale Generation Layer**: In parallel to the classification layer, a rationale generation layer is responsible for providing an explanation for the decision. This is achieved using an **attention mechanism**, which identifies the most critical sections of the paper that contributed to the model's decision.

**Training Procedure**:
The training process consists of several phases. First, the DistilBERT model is pre-trained on a massive corpus of text data. This pre-training allows the model to understand general language features like grammar, syntax, and semantic meaning. After pre-training, the model is fine-tuned on the specific academic paper dataset used for this project. Fine-tuning allows the model to learn domain-specific patterns that are crucial for academic paper classification, such as identifying research methodologies, key topics, and determining the overall publishability of a paper.

To fine-tune the model, we use **supervised learning**, where the model is trained on labeled data. The training set consists of academic papers that are labeled as either "publishable" or "not publishable" based on expert review. The model learns to minimize the difference between its predictions and the actual labels using a loss function, typically **cross-entropy loss**. During training, the model's parameters (weights) are updated using an optimization algorithm like **Adam**.

To evaluate the model's performance, we split the dataset into a training set and a validation set. The training set is used to train the model, while the validation set is used to test the model's generalization ability. Metrics such as **accuracy**, **precision**, **recall**, and **F1-score** are used to measure the model's performance. The model's performance is regularly evaluated during training to ensure it is not overfitting to the training data.

**Hyperparameter Tuning**:
Hyperparameters are crucial in determining the model's performance. Some of the important hyperparameters include the learning rate, batch size, number of training epochs, and the number of layers in the model. To find the optimal hyperparameters, we performed a **grid search** across various values of these hyperparameters. Grid search involves training the model with different combinations of hyperparameters and selecting the combination that provides the best results on the validation set.

Another important consideration in hyperparameter tuning is the **dropout rate**, which helps prevent overfitting by randomly "dropping" units during training, forcing the model to generalize better. Additionally, we experimented with different **optimizers**, such as Adam and RMSprop, to find the one that worked best for this task.

**Rationale Generation**:
In addition to classifying papers, the model generates a rationale for each classification decision. The rationale is a textual explanation that helps the user understand why a particular paper was classified as "publishable" or "not publishable." To generate the rationale, the model

uses an attention mechanism that highlights the most important sentences or phrases in the paper. This attention mechanism is based on the self-attention mechanism used in transformers, where the model assigns weights to different words or phrases based on their relevance to the classification decision.

The rationale generation process involves several steps:

1. The model first identifies key terms in the paper, such as research methods, results, and conclusions, using attention weights.
2. It then selects the most relevant sentences or passages that contributed to the classification.
3. Finally, the rationale is generated by extracting these sentences and rephrasing them into a coherent explanation.

The rationale provides insight into which sections of the paper were most influential in the decision-making process. This is particularly useful for authors, as it can help them improve their work by focusing on areas that need more clarity or strengthening.

---

## 5. Model Training

**Data Preparation and Preprocessing**:
Before the model can be trained, the data needs to be properly prepared. As mentioned earlier, the dataset consists of academic papers in text format. Each paper comes with an abstract, which is the key piece of text used for classification and summarization. The preprocessing pipeline begins with extracting the text from PDF files and cleaning it by removing irrelevant content. This includes eliminating page numbers, figures, and any other non-essential elements. Next, the text is tokenized, which involves breaking it down into individual words or subwords. Tokenization is an important step as it converts raw text into a format that can be fed into machine learning models.

Once tokenization is completed, the next step is **text normalization**. This includes converting all text to lowercase, removing stopwords, and performing **stemming** or **lemmatization**. Stemming involves reducing words to their root form, such as converting “running” to “run,” while lemmatization considers the full context of the word to return its base form. For example, lemmatization would reduce “better” to “good.” This helps reduce the complexity of the text and allows the model to focus on the core meaning of words.

After preprocessing, the text is passed through the **DistilBERT model** to generate embeddings. These embeddings capture the semantic meaning of the text and are used as the input for the classification model. The embeddings are fine-tuned during the training process to learn domain-specific features relevant to the academic paper classification task.

**Model Evaluation**:
The model is evaluated based on several key performance metrics. The most common metric used for classification tasks is **accuracy**, which measures the percentage of correct predictions.

However, accuracy alone does not give a complete picture of the model's performance, especially when the dataset is imbalanced (i.e., if one class is much more frequent than the other). In such cases, additional metrics like **precision**, **recall**, and **F1-score** are used to assess how well the model is performing across different classes.

- **Precision** measures the proportion of true positive predictions out of all positive predictions made by the model. A higher precision indicates that the model is correctly identifying positive samples.
- **Recall** measures the proportion of true positive predictions out of all actual positive samples. A higher recall indicates that the model is not missing any relevant papers.
- **F1-score** is the harmonic mean of precision and recall, providing a balance between the two metrics.

The model is trained and evaluated using a **validation set** to monitor its generalization ability. The training process involves several iterations or **epochs**, where the model's parameters are adjusted to minimize the loss function. During each epoch, the model is evaluated on the validation set to check if it is overfitting or underfitting. If the model is overfitting, techniques like **early stopping** and **dropout** are employed to prevent it from memorizing the training data and improving its generalization ability.

**Hyperparameter Tuning**:
During training, it is essential to fine-tune the hyperparameters of the model to achieve the best performance. Hyperparameters such as the learning rate, batch size, and the number of layers in the neural network play a crucial role in determining the model's performance. The learning rate controls how quickly the model learns, while the batch size determines how many training samples are used in each update step. By adjusting these hyperparameters, we can improve the model's ability to learn from the data and make accurate predictions.

## 6. Results and Discussion

**Model Performance**:
The trained model was evaluated on the test dataset, which consisted of academic papers labeled with their respective classifications. The evaluation metrics used include **accuracy**, **precision**, **recall**, and **F1-score**. The results showed that the model achieved high accuracy in classifying papers as either "publishable" or "not publishable." The precision and recall scores were also strong, indicating that the model is good at correctly identifying both positive and negative examples.

In terms of rationale generation, the model successfully highlighted relevant sections of the paper that contributed to its decision. The attention mechanism used in the model helped identify key sentences that had a significant impact on the classification decision. The rationales provided were coherent and aligned with the content of the paper, making them useful for authors seeking feedback on their work.

**Challenges Encountered**:
During the training and evaluation process, several challenges were encountered. One of the primary challenges was the **imbalance** in the dataset, where there were more "publishable" papers than "not publishable" papers. This imbalance can lead to biased predictions, where the model is more likely to classify papers as "publishable" due to their higher frequency in the dataset. To mitigate this, techniques like **oversampling** and **undersampling** were used to balance the dataset and prevent the model from being biased towards the majority class.

Another challenge was related to **interpretability**. While the model performed well in classification tasks, understanding the exact reasoning behind each decision was not always straightforward. Although the attention mechanism provided a rationale, it was sometimes difficult to pinpoint exactly why certain sections of a paper were considered more important than others. Future improvements could involve integrating more advanced explainability techniques, such as **LIME** or **SHAP** (SHapley Additive exPlanations), which can provide more transparent explanations of the model's decision-making process.

## 7. Conclusion

**Summary of Contributions**:
This project presented an AI-driven system for the classification of academic papers, capable of determining whether a paper is "publishable" or "not publishable," suggesting appropriate conferences, and generating an explanation for its decision. The system uses advanced **DistilBERT** transformer models for natural language processing, which are fine-tuned to handle academic text. Additionally, the model generates textual rationales for each classification, providing transparency in decision-making.

**Future Work**:
While the model performed well, there are several opportunities for future improvements. First, the rationale generation process could be enhanced by incorporating more advanced explainability techniques, allowing for clearer and more detailed explanations. Additionally, the dataset used for training could be expanded to include a more diverse range of academic papers from other domains to improve the model's generalization ability.

Another area of improvement is the integration of **multi-modal data** (e.g., including images and figures from papers) to enhance the classification decision. Many academic papers rely heavily on visuals to convey complex ideas, and incorporating such data could improve the model's ability to make accurate predictions.

Overall, this project provides a valuable tool for automating the peer-review process, reducing the workload of academic reviewers, and improving the efficiency of academic publishing.