# ****Resume Category Predictor****

# ****using NLP and ML****

## ****1. Title****

**Resume Category Predictor using NLP and Sentence-BERT**

## ****2. Problem Statement****

The primary objective of this project is to develop an **NLP-based application** capable of predicting the **most suitable job category** for a given resume. Resumes may be submitted as **text** or **PDF**, and the system provides a **single predicted job category**.

This application helps **recruiters and HR professionals** quickly classify resumes into appropriate categories, reducing manual effort and improving hiring efficiency.

## ****3. Dataset Details****

* **Source:** Kaggle – [Resume Dataset](https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset)
* **Size:** 2400+ resumes
* **Format:** CSV with Resume\_str, Resume\_html, Category columns + PDFs organized by category in folders
* **Description from Kaggle author:**

A collection of Resume Examples taken from livecareer.com for categorizing a given resume into any of the labels defined in the dataset. PDF files are stored in folders corresponding to their categories.

* **Columns inside CSV:**
  + ID: Unique identifier and PDF file name
  + Resume\_str: Resume text in string format
  + Resume\_html: Resume in HTML format
  + Category: Job category of the resume
* **Categories:**  
  HR, Designer, Information-Technology, Teacher, Advocate, Business-Development, Healthcare, Fitness, Agriculture, BPO, Sales, Consultant, Digital-Media, Automobile, Chef, Finance, Apparel, Engineering, Accountant, Construction, Public-Relations, Banking, Arts, Aviation

## ****4. Methodology****

To build the Resume Category Predictor, multiple NLP techniques and classification methods were explored. The workflow followed **three main phases**: text preprocessing, feature extraction, and classification.

### ****4.1 Text Preprocessing****

Text preprocessing is a crucial step to make the resumes suitable for NLP models. The following steps were applied:

1. **Lowercasing:** Convert all text to lowercase for uniformity.
2. **Remove URLs & whitespace:** Clean unnecessary characters and extra spaces.
3. **Tokenization:** Split text into individual words using nltk.word\_tokenize.
4. **Stopword Removal:** Common English stopwords were removed using nltk.corpus.stopwords.
5. **Lemmatization:** Words were reduced to their base form using WordNetLemmatizer.

**Reason for preprocessing:**

* Reduces noise in the text
* Helps models focus on meaningful keywords
* Improves classification performance

### ****4.2 Feature Extraction Methods Explored****

Multiple vectorization techniques were tested to convert text into numerical representations suitable for classification:

#### ****4.2.1 TF-IDF Vectorization****

* Converts resumes into vectors representing **term frequency weighted by inverse document frequency**.
* Captures the importance of words in each resume relative to the dataset.

**Observation:**

* Worked reasonably well, achieving ~74–76% accuracy with classifiers like XGBoost and Logistic Regression.
* Limitation: TF-IDF only captures word occurrence, **ignores semantic meaning**. Similar resumes with different wording could be misclassified.

#### ****4.2.2 Count Vectorization****

* Basic Bag-of-Words approach counting occurrences of words.
* Similar limitation as TF-IDF: **no context or semantics**.

**Observation:**

* Accuracy lower than TF-IDF (~67–74%)
* Predicted majority classes more often, failing for smaller categories

#### ****4.2.3 Sentence-BERT Embeddings****

* Replaced TF-IDF with **Sentence-BERT embeddings** (all-MiniLM-L6-v2) to capture **semantic meaning** of entire resumes.
* Converts each resume into a **dense vector of size 384**, preserving meaning beyond just word occurrence.

**Reason for switching:**

* Improved prediction for **small categories**
* Handles variations in wording better
* Captures context and meaning of sentences rather than individual words

### ****4.3 Classification Methods Explored****

Various classifiers were tested on the vectorized data:

1. **Random Forest**
2. **Gradient Boosting**
3. **Logistic Regression**
4. **SVM**
5. **Naive Bayes**
6. **XGBoost**
7. **Passive Aggressive Classifier**

**Observations:**

* **TF-IDF + XGBoost:** Highest accuracy ~74–75%, but sometimes predicted majority class (e.g., Business-Development) too often.
* **TF-IDF + Logistic Regression:** Accuracy slightly lower (~76%) but simpler and faster.
* **Sentence-BERT + Logistic Regression:** Best combination for **semantic understanding**, predicting diverse categories accurately, and handling small categories.

**Reason for final choice:**

* Logistic Regression with **Sentence-BERT embeddings** provides:
  + Single-category predictions
  + Semantic understanding
  + Fast inference for text and PDFs

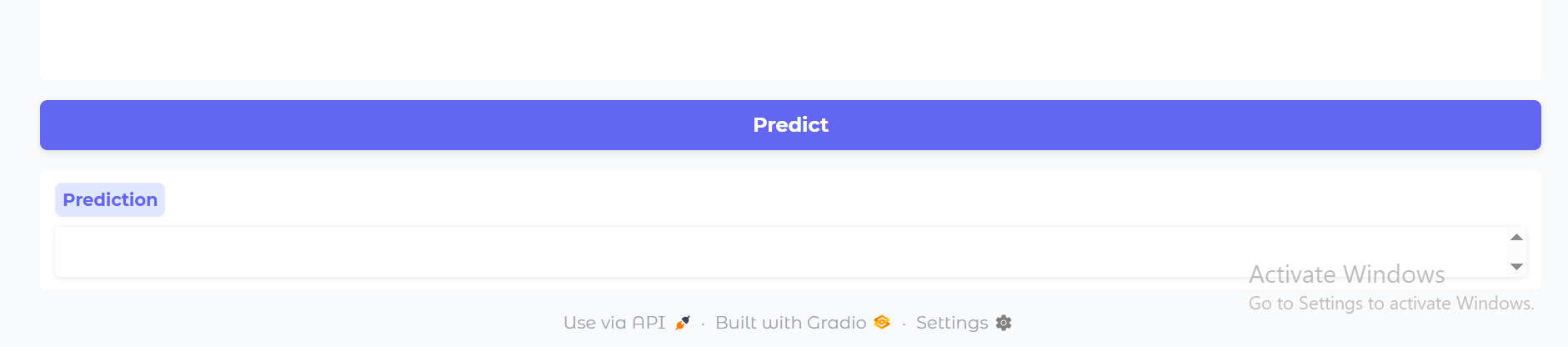
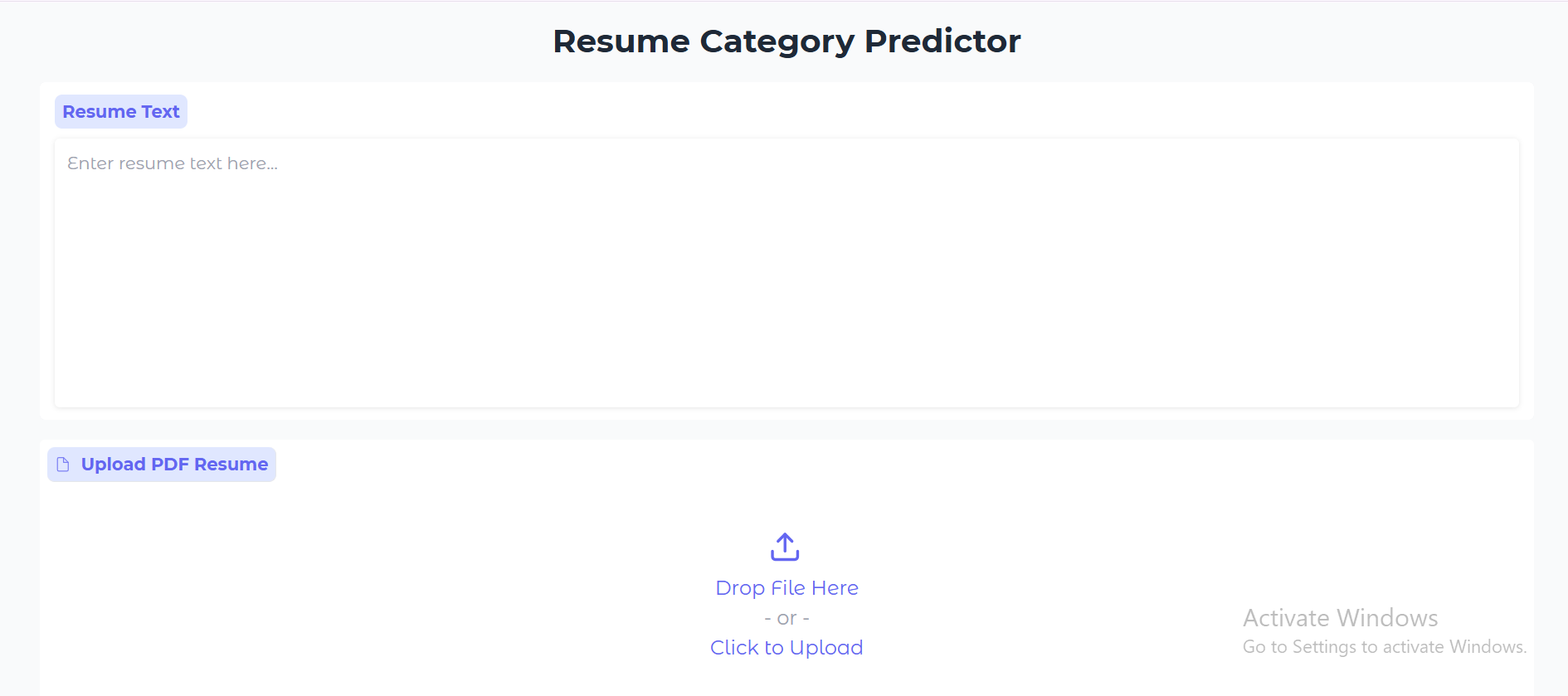
### ****4.4 UI Design with Gradio****

A **user-friendly interface** was developed using Gradio:

**Features:**

1. **Textbox** to paste resume text
2. **File upload** for PDF resumes
3. **Single “Predict” button** (blue)
4. **Output:** Displays the predicted category
5. **Theme:** Soft, colorful background for better UI experience

**Reason for UI design:**

* Simplifies usage for HR or recruiters
* Supports both text and PDF without additional options
* Clear, visually appealing output
* 

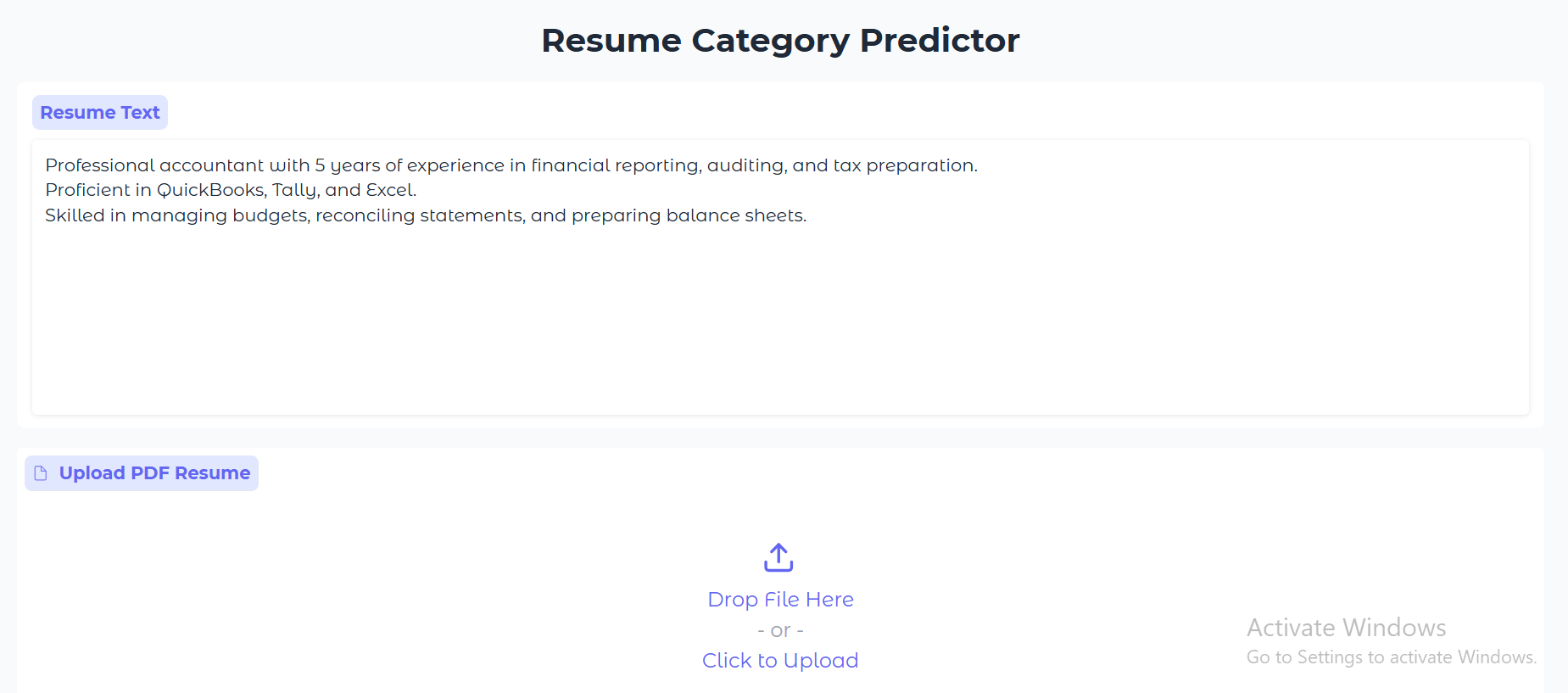
## ****5. Results****

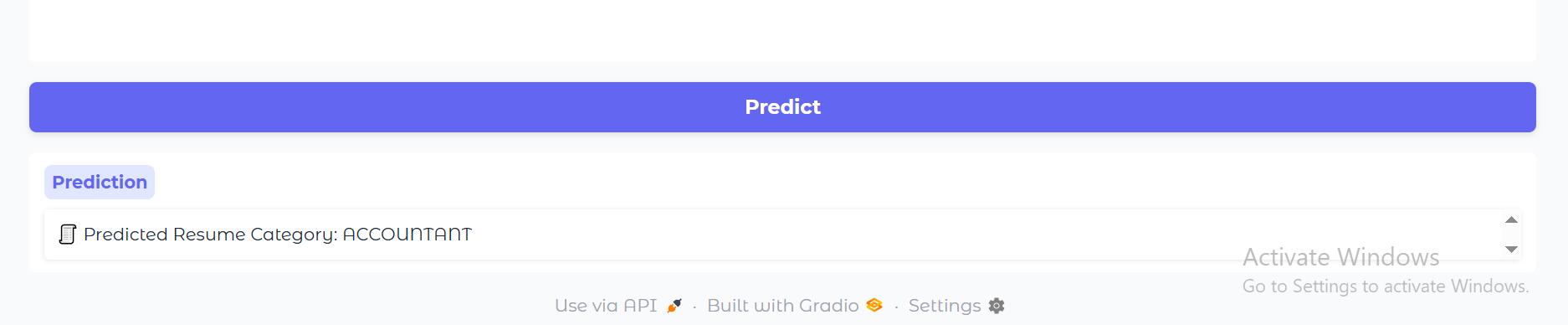
### ****5.1 Accuracy****

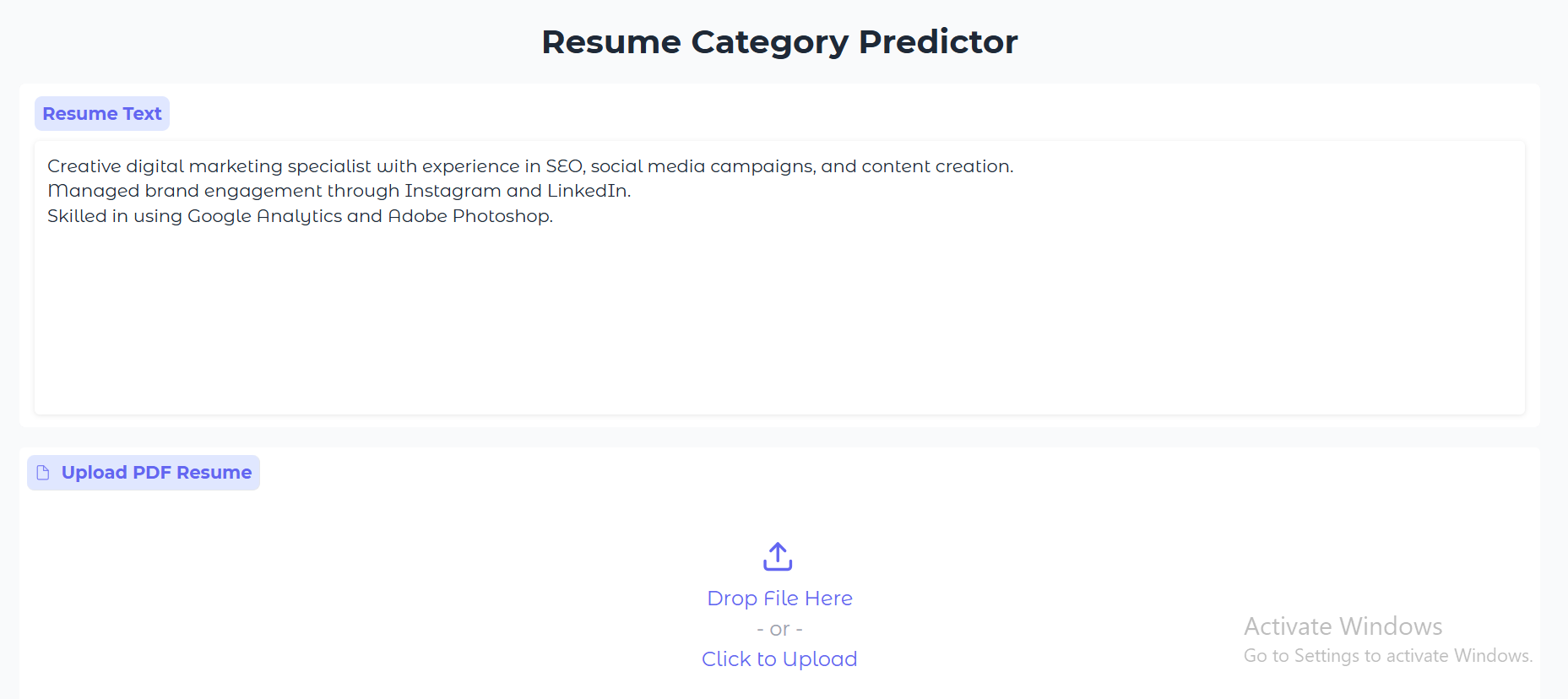
* Logistic Regression + Sentence-BERT embeddings achieved **~76–78% accuracy** on the test set.

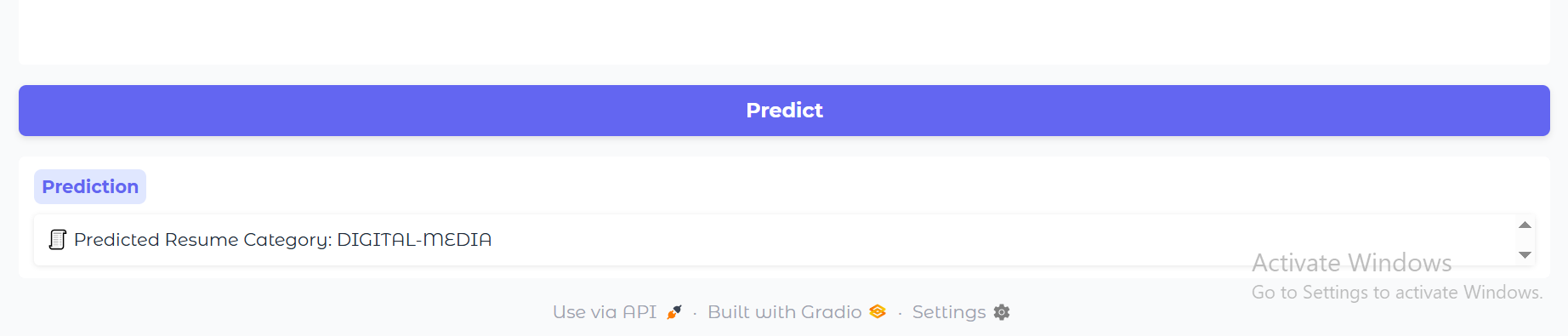
### ****5.2 Example Predictions****

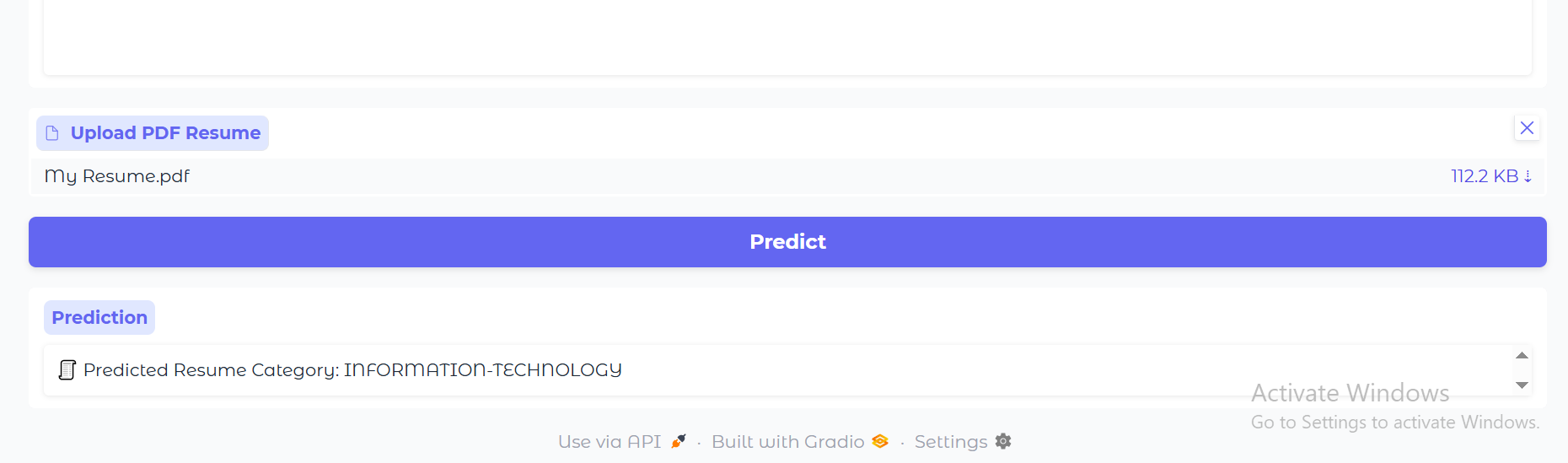
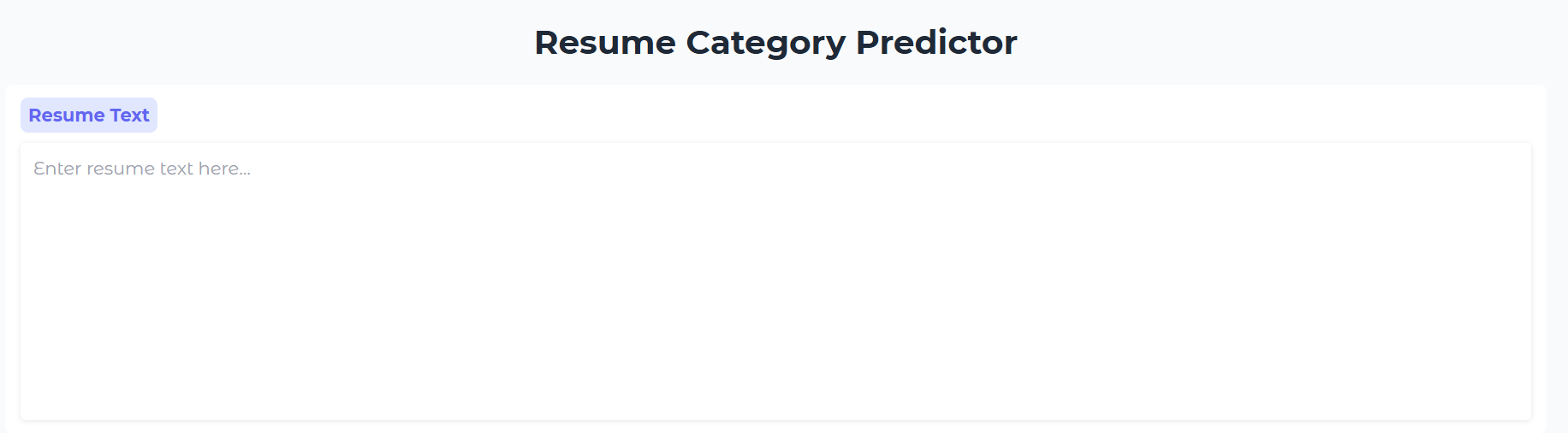
| **Resume Example** | **Predicted Category** |
| --- | --- |
| Software engineer with 4 years of Python, Django, React | ENGINEERING |
| Accountant with 5 years in financial reporting | ACCOUNTANT |
| High school English teacher | TEACHER |
| Digital marketing specialist | DIGITAL-MEDIA |
| Registered nurse | HEALTHCARE |
| Commercial pilot | AVIATION |

****

****

****

****

****

**Observation:**

* Semantic embeddings improved predictions for **all categories**
* Single-category prediction works reliably
* PDF input predictions match text input predictions

## ****6. Conclusion****

* Developed an **NLP-based Resume Category Predictor** using **Sentence-BERT embeddings** and **Logistic Regression**.
* The system can handle **both pasted text and uploaded PDFs**.
* Demonstrated understanding of key NLP concepts:
  + Text preprocessing (tokenization, stopword removal, lemmatization)
  + Vectorization (TF-IDF, Sentence-BERT embeddings)
  + Classification and semantic analysis
* UI is **user-friendly, visually appealing, and functional**
* Potential extensions:
  + Fine-tuning Sentence-BERT on the dataset
  + Adding batch predictions
  + Handling scanned PDF resumes using OCR

## ****7. References****

1. <https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset>
2. <https://www.kaggle.com/code/abhiramrayadurgam/resume-classification-using-nlp-and-ml/notebook>
3. <https://chatgpt.com/>