**AI Developer Technical Assessment (CodeX)**

**by**

**Arslan Khalid**

**1. Introduction**

Greeklish refers to Greek text written using the Latin alphabet (e.g., "ti kaneis" instead of "τι κάνεις"). This project aims to develop a classifier to distinguish between Greeklish and English text using machine learning techniques.

**2. Choice of Data Sources**

To build an effective classifier, we collected data from the following sources:

* **Greeklish Data**: Scraped from at least 3 different online forums and social media posts where Greeklish is commonly used.
* **English Data**: Collected from various English text sources, including Wikipedia articles and online news articles.
* **Minimum Dataset Size**: Ensured at least 300 unique sentences per class (600 total).

**3. Data Scraping Methods & Preprocessing Steps**

**Data Scraping**

* Implemented web scraping using Python's requests and BeautifulSoup libraries to collect Greeklish text.
* Used newspaper3k for extracting English sentences from news websites.
* Saved the scraped data into a CSV file.

**Preprocessing**

* **Text Cleaning**: Lowercased text, removed numbers and punctuation.
* **Sentence Splitting**: Split paragraphs into separate sentences using nltk.sent\_tokenize().
* **Data Splitting**: Used an 80-20 train-test split to evaluate model performance.

**4. Rationale for Model Selection & Training**

**Model Choice: Support Vector Machine (SVM)**

* SVM with an RBF kernel was chosen due to its effectiveness in text classification with limited data.
* TF-IDF vectorization with **character n-grams (1-3)** was used to capture transliteration patterns in Greeklish.

**Training Process**

* Tokenized text using TfidfVectorizer (character-level n-grams).
* Trained an SVM classifier using sklearn.svm.SVC with hyperparameter tuning.
* Evaluated using accuracy, precision, recall, and F1-score.

**Model Evaluation**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Accuracy | 99.5% |
| Precision | 100% |
| Recall | 99.2% |
| F1-score | 99.6% |
|  |  |

**5. Challenges & Solutions**

**1. Noisy Data in Greeklish Text**

* **Solution**: Used aggressive cleaning and removed special characters.

**2. Imbalanced Dataset**

* **Solution**: Ensured equal samples per class during data collection.

**3. Handling Mixed-Language Text**

* **Solution**: Trained on real-world mixed-language data to improve robustness.

**6. Conclusion**

This project successfully developed an SVM-based Greeklish vs. English classifier with high accuracy. Future improvements could include deep learning models like BiLSTM with attention to further enhance performance.