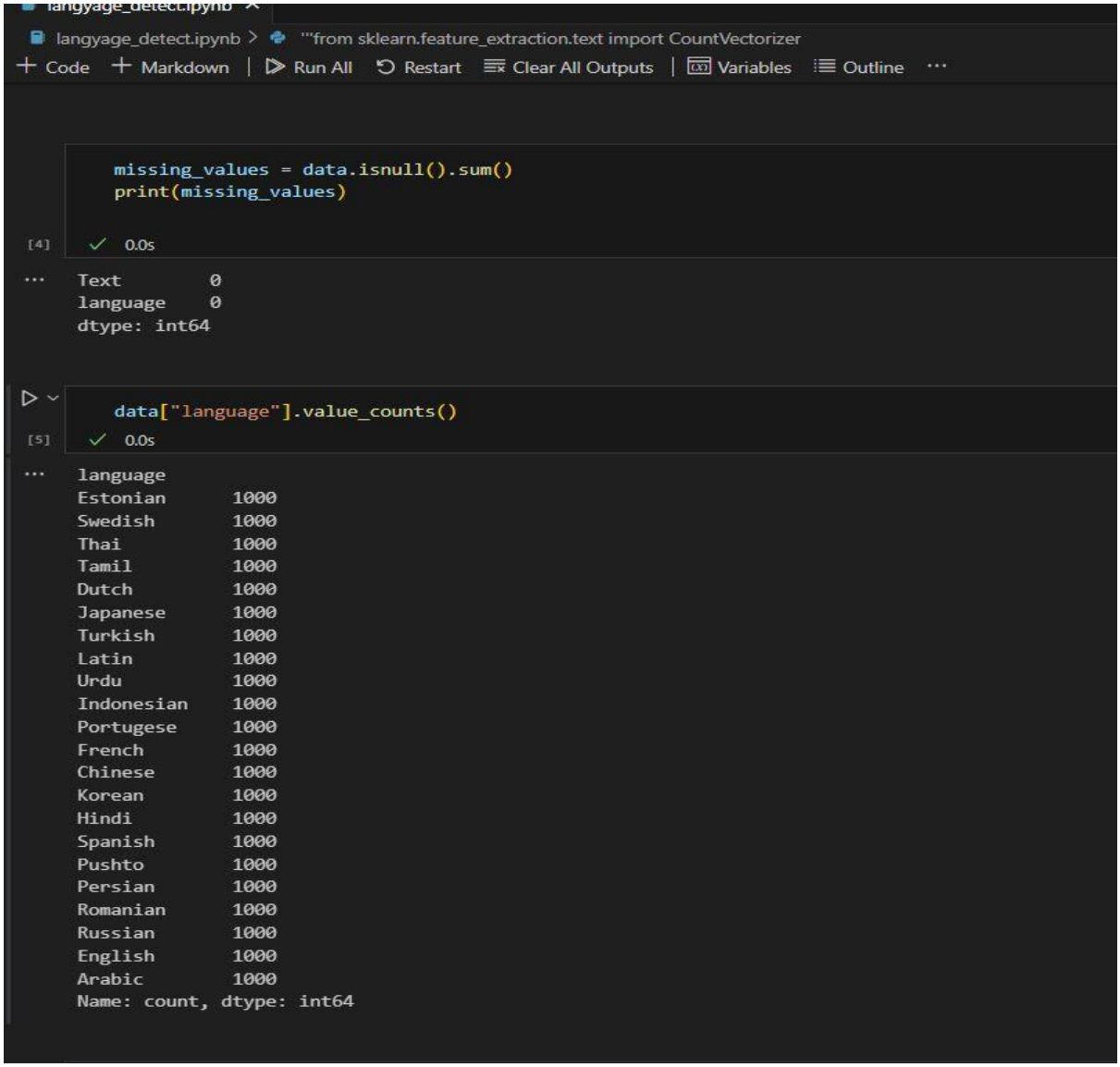
LANGUAGE DETECTION MODEL USING PYTHON + ML

By-Devansh Sharma

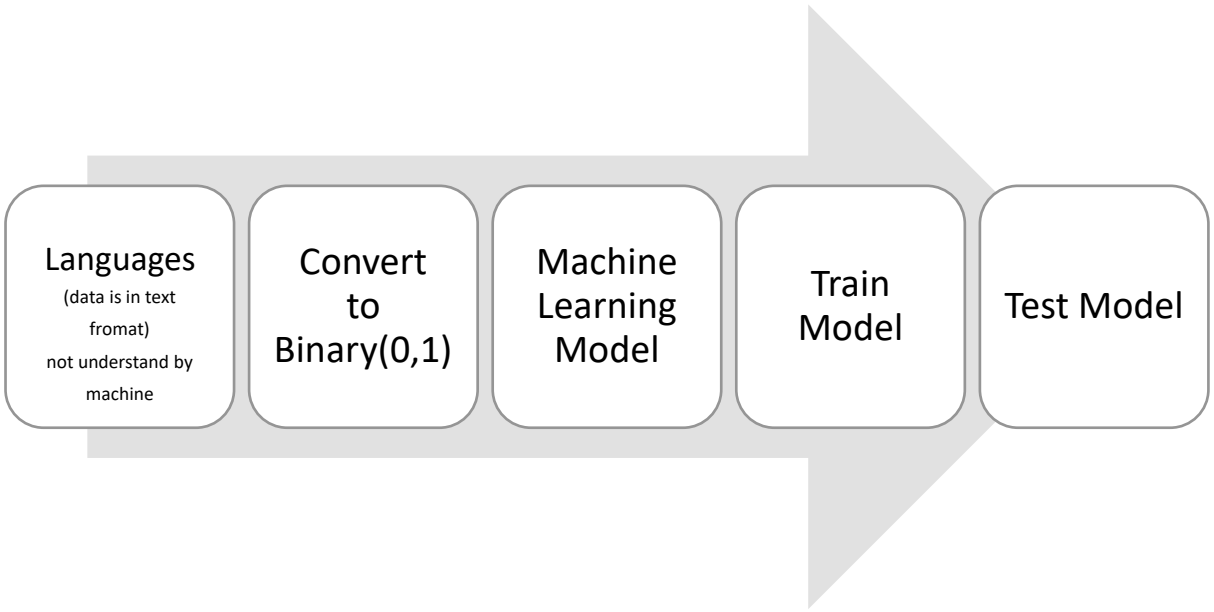
Git-Hub- https://github.com/devanshh7

Project Summary:

This project focuses on developing a language detection model capable of identifying 22 different languages using a dataset of 20,000 rows. The goal of the model is to accurately predict the language of a given text by leveraging machine learning techniques. The project uses various libraries and approaches for data manipulation, text processing, and model building. Over all this model has achieved 95.33% accuracy which train and testing.



PROCESS :



The image illustrates the process of building a language detection model. Here's a brief explanation of each step:

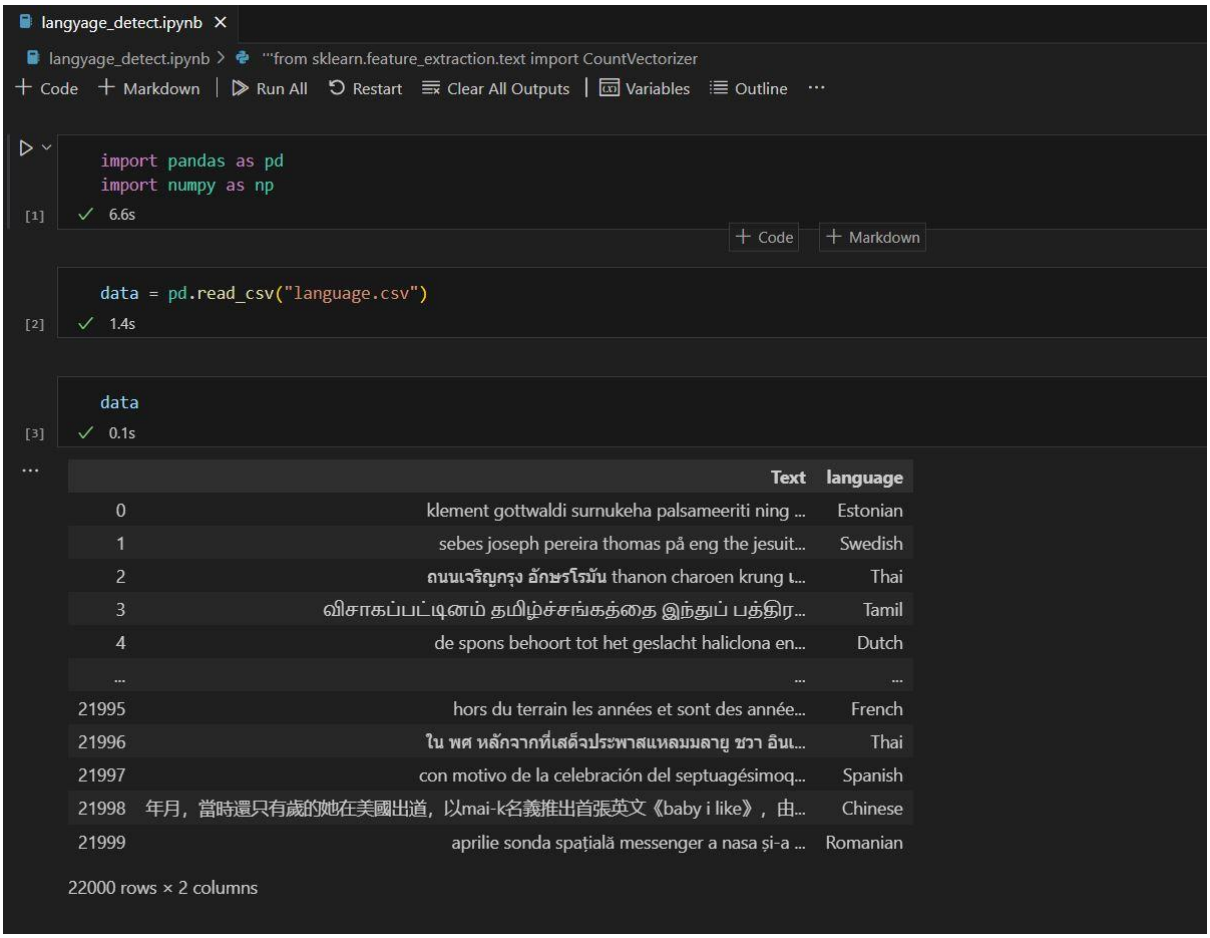
1. **Languages (Text Data)**: The data consists of text in different languages, which machines cannot understand directly.
2. **Convert to Binary (0,1)**: The text is transformed into a format that a machine can interpret. This is done using techniques like one-hot encoding or Bag of Words, which convert text into numerical representations (binary).
3. **Machine Learning Model**: A machine learning model is then built to process the numerical data and learn patterns that differentiate the languages.
4. **Train Model**: The model is trained on a dataset to learn the features of different languages.
5. **Test Model**: After training, the model is tested on unseen data to evaluate its accuracy in detecting languages.

This sequence ensures that the model can learn from data and generalize well to new text inputs.

Libraries Used:

Pandas: This library was used to load, manage, and manipulate the dataset efficiently. Pandas helped in handling large amounts of data, allowing for quick data cleaning and pre-processing.

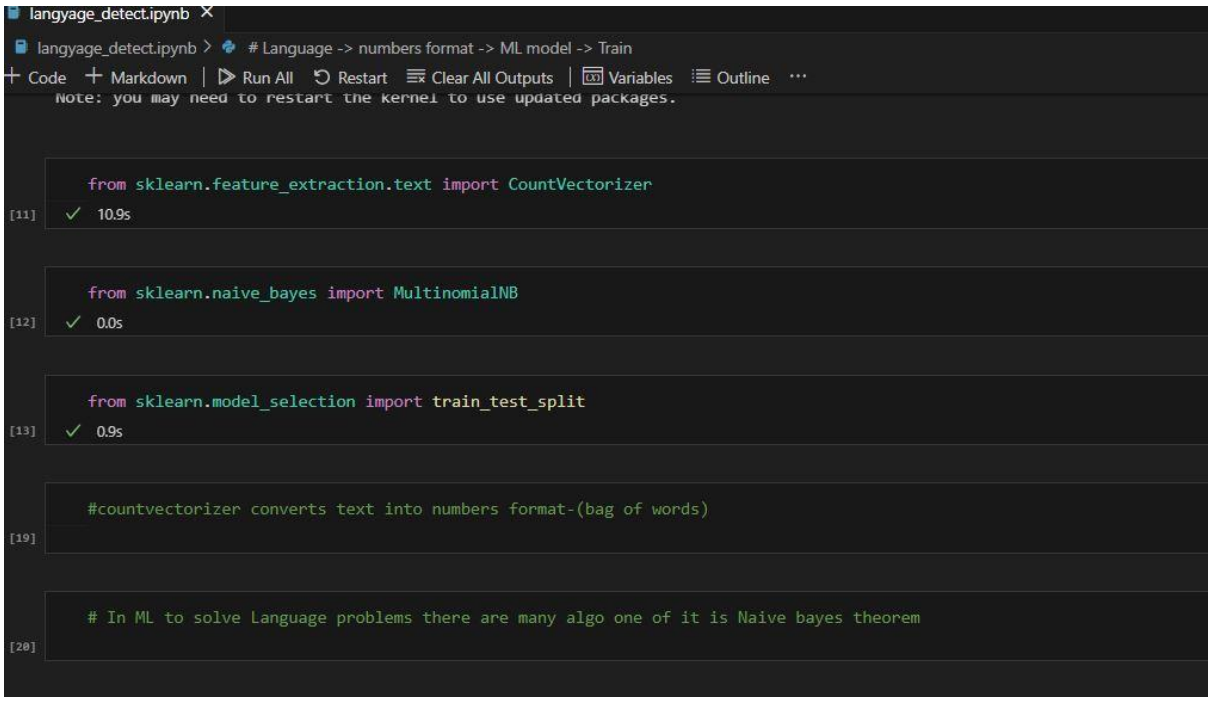
NumPy: NumPy provided support for numerical operations, enabling efficient handling of arrays and matrices, which are essential for data transformation and mathematical computations in machine learning.



	Text	language
0	klement gottwaldi surnukeha palsameeriti ning ...	Estonian
1	sebes joseph pereira thomas på eng the jesuit...	Swedish
2	ถนนเจริญกรุง อักษรโรมัน thanon charoen krung เ...	Thai
3	விசாகப்பட்டினம் தமிழ்ச்சங்கத்தை இந்துப் பத்திர...	Tamil
4	de spons behoort tot het geslacht haliclona en...	Dutch
...	...	...
21995	hors du terrain les années et sont des année...	French
21996	ใน พศ หลักจากที่เสด็จประพาสแหลมมลายู ชวา อินเ...	Thai
21997	con motivo de la celebración del septuagésimoq...	Spanish
21998	年月，當時還只有歲的她在美國出道，以mai-k名義推出首張英文《baby i like》，由...	Chinese
21999	aprilie sonda spațială messenger a nasa și-a ...	Romanian

22000 rows × 2 columns

⇨ **Scikit-learn (sklearn)**: A comprehensive library for machine learning, Scikit-learn was used for implementing various algorithms and utilities, including feature extraction, model building, and evaluation.

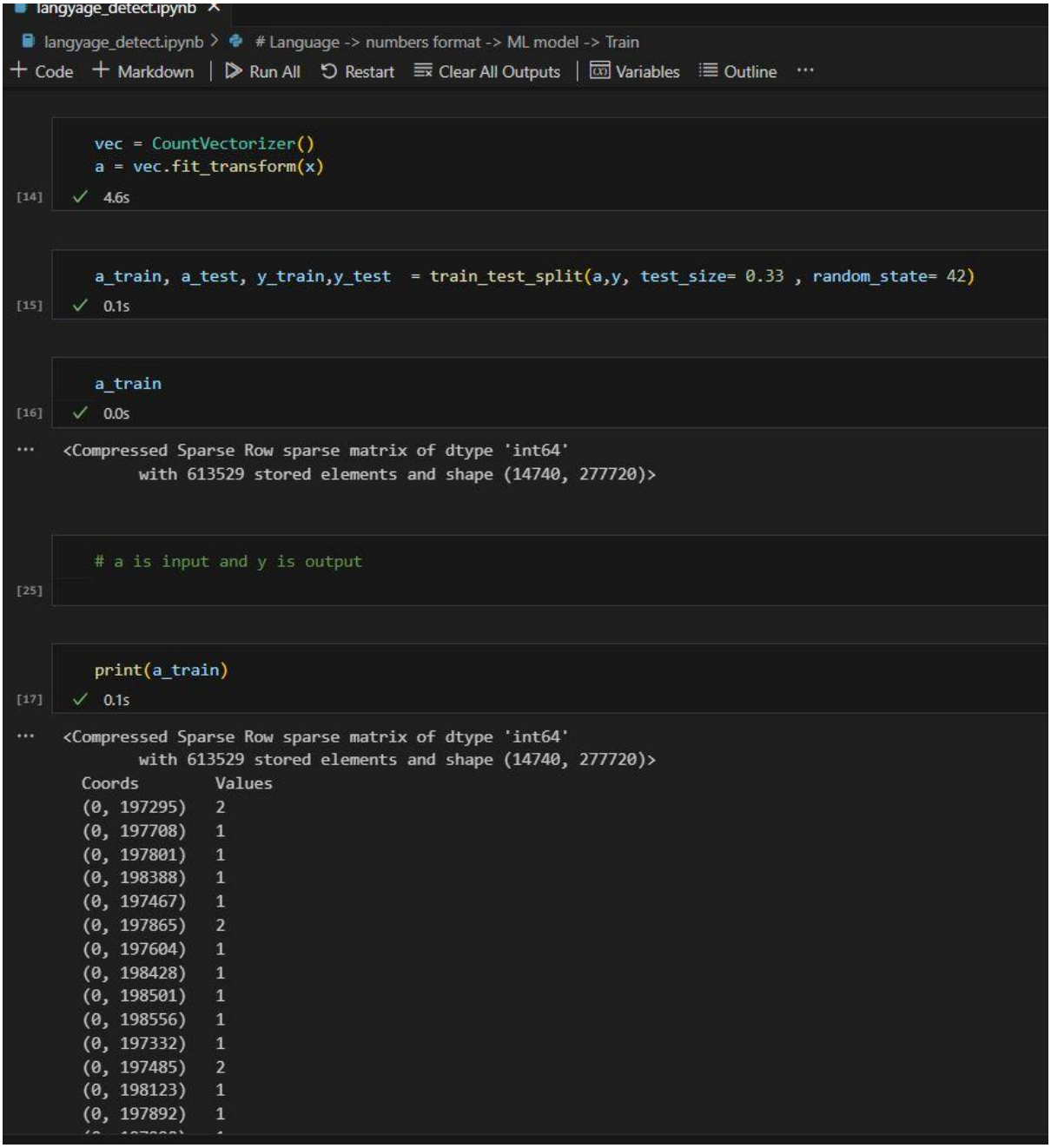


⇨ **Count Vectorizer** (from sklearn.feature_extraction.text): CountVectorizer is a tool for converting text data into a matrix of token counts. It transforms the dataset into numerical features by counting the frequency of each word in the text, which serves as input for the machine learning model.

Key Concepts:

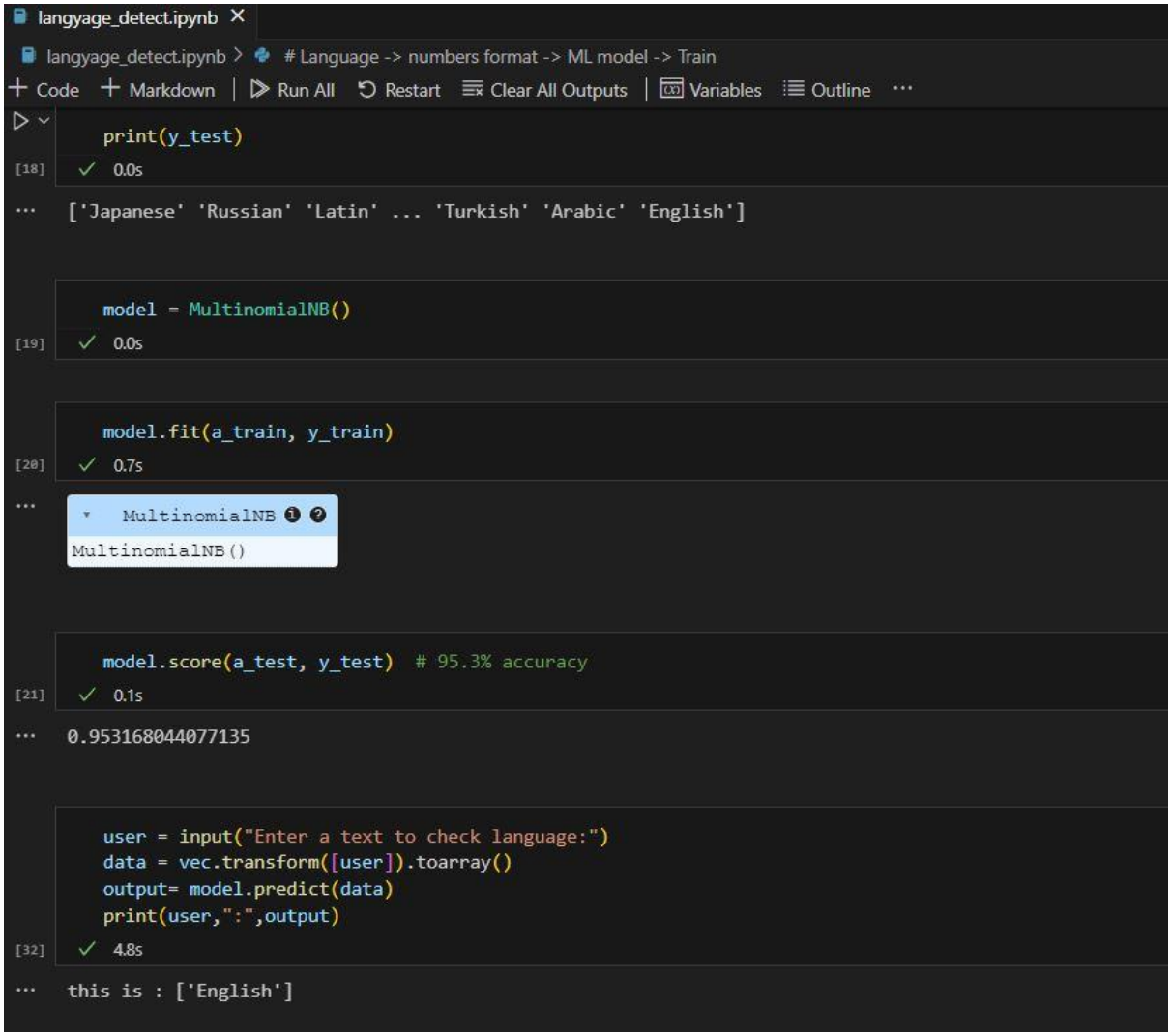
Bag of Words (BoW) Approach: The BoW approach was used to represent the text in a way that counts the occurrence of unique words across the dataset. It ignores grammar and word order, focusing on the frequency of words in a sentence or document. This technique helped in creating a numerical representation of text data, enabling the model to distinguish between languages based on word usage.

MultinomialNB (from sklearn.naive_bayes): Multinomial Naive Bayes is a probabilistic algorithm commonly used for text classification. In this project, it was applied to predict the language of a given text by estimating the likelihood of word frequencies belonging to different languages.



train_test_split: This function from Scikit-learn was used to divide the dataset into training and testing sets. The training set is used to train the model, while the testing set evaluates its performance. Splitting the data helps in ensuring that the model can generalize well to unseen data, improving accuracy and reducing overfitting.

This combination of tools and methods resulted in an effective language detection model that can accurately classify text into one of 22 languages.



Output Detecting Language

Accuracy is 95.33%.