AI-Based Prime Number Checker

Author: Prashant Kumar

Date: March 10, 2025

Introduction

This project aims to create an AI-based prime number checker using Logistic Regression.

The model is trained to identify whether a given number is prime by learning from a dataset generated

through the Sieve of Eratosthenes.

Methodology

1. Generate prime numbers using the Sieve of Eratosthenes.

2. Create a dataset where numbers are labeled as prime (1) or not prime (0).

3. Train a Logistic Regression model using this dataset.

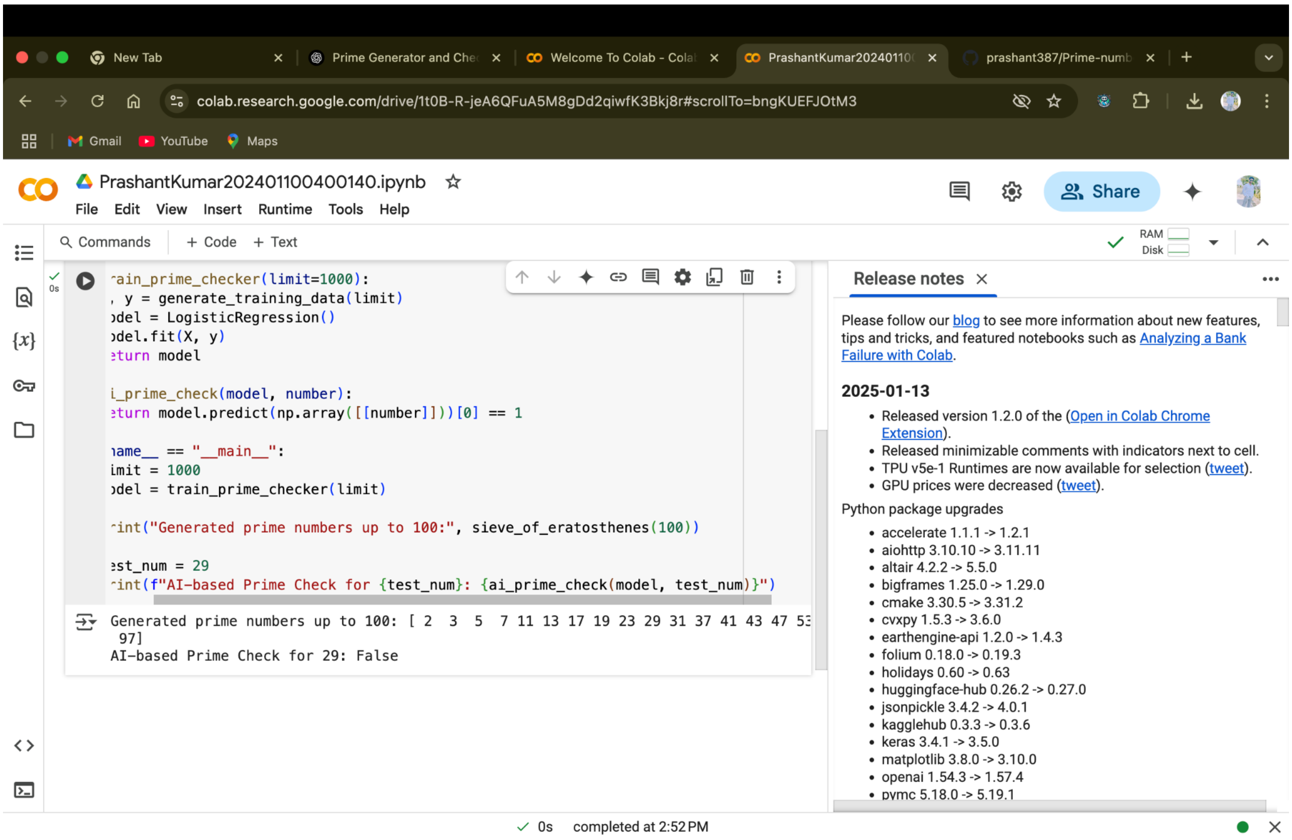
4. Use the trained model to predict whether a new number is prime.

Code

```
import numpy as np
from sklearn.linear_model import LogisticRegression

def sieve_of_eratosthenes(limit):
    primes = np.ones(limit + 1, dtype=bool)
    primes[0:2] = False
    for num in range(2, int(limit**0.5) + 1):
        if primes[num]:
            primes[num*num : limit+1 : num] = False
    return np.where(primes)[0]

def generate_training_data(limit):
    primes = sieve_of_eratosthenes(limit)
    X = np.arange(limit).reshape(-1, 1)
    y = np.isin(X.flatten(), primes).astype(int)
    return X, y

def train_prime_checker(limit=1000):
    X, y = generate_training_data(limit)
    model = LogisticRegression()
    model.fit(X, y)
    return model

def ai_prime_check(model, number):
    return model.predict(np.array([[number]]))[0] == 1

if __name__ == "__main__":
    limit = 1000
    model = train_prime_checker(limit)
    print("Generated prime numbers up to 100:", sieve_of_eratosthenes(100))
    test_num = 29
    print(f"AI-based Prime Check for {test_num}: {ai_prime_check(model, test_num)}")
```

Output

```
Generated prime numbers up to 100: [ 2  3  5  7 11 13 17 19 23 29 31 37 41
43 47 53 59 61 67 71 73 79 83 89 97]


AI-based Prime Check for 29: False
```



```
rain_prime_checker(limit=1000):
, y = generate_training_data(limit)
odel = LogisticRegression()
odel.fit(X, y)
eturn model

i_prime_check(model, number):
eturn model.predict(np.array([[number]]))[0] == 1

name__ == "__main__":
imit = 1000
odel = train_prime_checker(limit)

rint("Generated prime numbers up to 100:", sieve_of_eratosthenes(100))

est_num = 29
rint(f"AI-based Prime Check for {test_num}: {ai_prime_check(model, test_num)}")
```

```
Generated prime numbers up to 100: [ 2  3  5  7 11 13 17 19 23 29 31 37 41 43 47 53
 97]
AI-based Prime Check for 29: False
```

Release notes

Please follow our blog to see more information about new features, tips and tricks, and featured notebooks such as Analyzing a Bank Failure with Colab.

2025-01-13

- Released version 1.2.0 of the (Open in Colab Chrome Extension).
- Released minimizable comments with indicators next to cell.
- TPU v5e-1 Runtimes are now available for selection (tweet).
- GPU prices were decreased (tweet).

Python package upgrades

- accelerate 1.1.1 -> 1.2.1
- aiohttp 3.10.10 -> 3.11.11
- altair 4.2.2 -> 5.5.0
- bigframes 1.25.0 -> 1.29.0
- cmake 3.30.5 -> 3.31.2
- cvxpy 1.5.3 -> 3.6.0
- earthengine-api 1.2.0 -> 1.4.3
- folium 0.18.0 -> 0.19.3
- holidays 0.60 -> 0.63
- huggingface-hub 0.26.2 -> 0.27.0
- jsonpickle 3.4.2 -> 4.0.1
- kagglehub 0.3.3 -> 0.3.6
- keras 3.4.1 -> 3.5.0
- matplotlib 3.8.0 -> 3.10.0
- openai 1.54.3 -> 1.57.4
- pymc 5.18.0 -> 5.19.1

✓ 0s completed at 2:52 PM

References

1. Scikit-learn Documentation: https://scikit-learn.org/

2. NumPy Documentation: https://numpy.org/

3. Sieve of Eratosthenes Algorithm: https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes