

## Projenin amacı:

Bu projenin amacı internetten bağımsız olarak localde javascript kodu analiz edebilen codet5 tabanlı LLM modeli geliştirmektir. Projede genel olarak hazır olan modellerin kurulumu, çalıştırılması ve değerlendirilmesi yapılmıştır.

Bunun için toplamda 3 codeT5 bir codeT5+ modeli bulunmuş ve incelenmiştir. Bu modellerden üç tanesi kod analizi yapmak için, bir tanesi ise text girdisi ile kod yazmak için kullanılabilir.

Modeller temel görevler için çalışmakla beraber daha iyi performans için finetune edilmelidir.

## Projede kullanılan modeller:

- Model:** api/saved-pretrained-kde-cpp-tm | T5ForConditionalGeneration  
**Tokenizer:** Salesforce/codet5-base-multi-sum | RobertaTokenizer  
**Görev:** Girilen kodun analizi (code-to-text)  
**Link:** <https://github.com/tm243/CodeT5-KDE>
- Model:** Salesforce/codet5-base-multi-sum | T5ForConditionalGeneration  
**Tokenizer:** Salesforce/codet5-base-multi-sum | RobertaTokenizer  
**Görev:** Girilen kodun analizi (code-to-text)  
**Link:** <https://huggingface.co/Salesforce/codet5-base-multi-sum>
- Model:** Salesforce/codet5-base-codexglue-sum-javascript | T5ForConditionalGeneration  
**Tokenizer:** Salesforce/codet5-base | RobertaTokenizer  
**Görev:** Girilen metne göre kod üretilmesi. (code-to-text)  
**Link:** <https://huggingface.co/Salesforce/codet5-base-codexglue-sum-javascript>
- Model:** Salesforce/codet5p-770m-py | T5ForConditionalGeneration  
**Tokenizer:** Salesforce/codet5p-770m-py | RobertaTokenizer  
**Görev:** Girilen metne göre kod üretilmesi. (text-to-code)  
**Link:** <https://huggingface.co/Salesforce/codet5p-770m-py>

## Projenin bilgisayara kurulumu:

Github linki: [https://github.com/gokeay/hvlsn\\_2024](https://github.com/gokeay/hvlsn_2024)

- Proje klonlanır ve içine girilir.
  - Git clone [https://github.com/gokeay/hvlsn\\_2024](https://github.com/gokeay/hvlsn_2024)
  - cd hvlsn\_2024
- Sanal ortam oluşturulur ve ortama girilir. (İsteğe bağlı)
  - python -m venv myenv
  - .\myenv\Scripts\activate
- Gereklilikler indirilir.
  - pip install -r requirements.txt
- Her model ayrı dosyanın içerisinde. Main.py, main\_2.py, main\_3.py, main\_4 sırası ile model\_1, model\_2, model\_3 ve model\_4 ü çalıştırır.

Notlar:

- Her modelin içinde örnek kodlar yorum satırı içinde bulunabilir.
- model\_1 ağırlıkları github linki aracılığı ile kurulmalıdır.

## Model-1 için örnek girdiler ve modelin ürettiği çıktılar.

Girdi:\_1

```
code = """

function calculateSquares(numbers) {
  const result = {};

  numbers.forEach(number => {
    if (typeof number === 'number') {
      result[number] = number * number;
    } else {
      console.warn(`${number} is not a number`);
    }
  });

  return result;
}

console.log(squares);
"""
```

Çıktı\_1:

```
Result: This function calculates the squares of an array of numbers.

GPU execution time: 0.84 seconds
```

Girdi\_2.1: (Yorum satırları ile)

```
code = """

// program to find the largest among three numbers

// take input from the user
const num1 = parseFloat(prompt("Enter first number: "));
const num2 = parseFloat(prompt("Enter second number: "));
const num3 = parseFloat(prompt("Enter third number: "));
let largest;

// check the condition
if(num1 >= num2 && num1 >= num3) {
  largest = num1;
}
else if (num2 >= num1 && num2 >= num3) {
  largest = num2;
}
```

```

}
else {
    largest = num3;
}

// display the result
console.log("The largest number is " + largest);

"""

```

Çıktı\_2.1:

```

Result: This function retrieves the largest number from the user.
GPU execution time: 0.77 seconds

```

Girdi\_2.2: (Yorum satırları olmadan)

```

code = """

const num1 = parseFloat(prompt("Enter first number: "));
const num2 = parseFloat(prompt("Enter second number: "));
const num3 = parseFloat(prompt("Enter third number: "));
let largest;

if(num1 >= num2 && num1 >= num3) {
    largest = num1;
}
else if (num2 >= num1 && num2 >= num3) {
    largest = num2;
}
else {
    largest = num3;
}

console.log("The largest number is " + largest);

"""

```

Çıktı\_2.2:

```

Result: This function takes the first number in a list and gets the largest number among three numbers.
It allows to find the largest value within a 3rd parameter.
GPU execution time: 1.27 seconds

```

## Model-2 için örnek girdiler ve modelin ürettiği çıktılar.

Girdi\_1:

```
code = """

function calculateSquares(numbers) {
  const result = {};

  numbers.forEach(number => {
    if (typeof number === 'number') {
      result[number] = number * number;
    } else {
      console.warn(`${number} is not a number`);
    }
  });

  return result;
}

console.log(squares);
"""
```

Çıktı\_1:

```
Result: Function to calculate the squares of a sequence of numbers.
GPU execution time: 0.73 seconds
```

Girdi\_2.1: (Yorum satırları ile)

```
code = """

// program to find the largest among three numbers

// take input from the user
const num1 = parseFloat(prompt("Enter first number: "));
const num2 = parseFloat(prompt("Enter second number: "));
const num3 = parseFloat(prompt("Enter third number: "));
let largest;

// check the condition
if(num1 >= num2 && num1 >= num3) {
  largest = num1;
}
else if (num2 >= num1 && num2 >= num3) {
  largest = num2;
}
else {
  largest = num3;
}
"""
```

```
// display the result
console.log("The largest number is " + largest);

"""
```

Çıktı\_2.1:

```
Result: This function asks the user to find the largest among three numbers in a sequence and
displays it.
GPU execution time: 0.92 seconds
```

Girdi\_2.2: (Yorum satırları olmadan)

```
code = """

const num1 = parseFloat(prompt("Enter first number: "));
const num2 = parseFloat(prompt("Enter second number: "));
const num3 = parseFloat(prompt("Enter third number: "));
let largest;

if(num1 >= num2 && num1 >= num3) {
    largest = num1;
}
else if (num2 >= num1 && num2 >= num3) {
    largest = num2;
}
else {
    largest = num3;
}

console.log("The largest number is " + largest);

"""
```

Çıktı\_2.2:

```
Result: Reads in three numbers and returns the largest one.
GPU execution time: 0.66 seconds
```

## Model-3 için örnek girdiler ve modelin ürettiği çıktılar.

Girdi\_1:

```
code = """

function calculateSquares(numbers) {
  const result = {};

  numbers.forEach(number => {
    if (typeof number === 'number') {
      result[number] = number * number;
    } else {
      console.warn(`${number} is not a number`);
    }
  });

  return result;
}

console.log(squares);
"""
```

Çıktı\_1:

```
GPU Result: Calculate the squares of a list of numbers
GPU execution time: 0.74 seconds
```

s

Girdi\_2.1: (Yorum satırları ile)

```
code = """

// program to find the largest among three numbers

// take input from the user
const num1 = parseFloat(prompt("Enter first number: "));
const num2 = parseFloat(prompt("Enter second number: "));
const num3 = parseFloat(prompt("Enter third number: "));
let largest;

// check the condition
if(num1 >= num2 && num1 >= num3) {
  largest = num1;
}
else if (num2 >= num1 && num2 >= num3) {
  largest = num2;
}
else {
  largest = num3;
}
```

```
// display the result
console.log("The largest number is " + largest);

"""
```

Çıktı\_2.1:

```
GPU Result: Find the largest number in a sequence of three numbers
GPU execution time: 0.96 seconds
```

Girdi\_2.2: (Yorum satırları olmadan)

```
code = """

const num1 = parseFloat(prompt("Enter first number: "));
const num2 = parseFloat(prompt("Enter second number: "));
const num3 = parseFloat(prompt("Enter third number: "));
let largest;

if(num1 >= num2 && num1 >= num3) {
    largest = num1;
}
else if (num2 >= num1 && num2 >= num3) {
    largest = num2;
}
else {
    largest = num3;
}

console.log("The largest number is " + largest);

"""
```

Çıktı\_2.2:

```
GPU Result: Get the largest number of a group
GPU execution time: 0.62 seconds
```

## Model-4 için örnek girdiler ve modelin ürettiği çıktılar.

Girdi\_1:

```
code = """  
  
write javascript function to calculate the sum of given two numbers  
  
"""
```

Çıktı\_1:

```
GPU Result:  
def sum(a, b):  
    return a + b  
GPU execution time: 1.83 seconds
```

Girdi\_2:

```
code = """  
  
Function that takes a temperature in Celsius as input from the user and  
converts it to Fahrenheit.  
The program should print the converted temperature with an appropriate  
message.  
  
"""
```

Çıktı\_2:

```
GPU Result:  
def convert_celsius(c):  
    f = (9/5)*c + 32 #f is the value of Celsius in Fahrenheit.  
    return f  
  
print("Welcome to Temperature Converter")  
  
c = float(input("Enter a temperature in Celsius: "))  
f = convert_celsius(c)  
print("The temperature in Fahrenheit is", f, "F.")  
GPU execution time: 5.43 seconds
```