

# EXPLORING PROMPT ENGINEERING WITH PYTHON

Connect.it The Future of Python

Author: Eng. Carlos Andrés Sierra, M.Sc.

`carlos.andres.sierra.v@gmail.com`

Professor

Computer Engineer

Master on Computer Engineering

March 12th 2024

# Outline

# Outline

# Welcome!

Endava's Connect.IT is a series of events that bring together the best minds in tech to discuss the latest trends and innovations in the industry.

```
../images/endava_meet.png
```

# Meet the Speaker I

../images/endava\_hi.png

# Meet the Speaker II



- *PyCon Colombia* and *Python Bogotá* **Co-organizer**.
- Former **Software Engineer** and **Technical Leader** of Machine Learning and Data Science.
- **Professor** at *UD Francisco José de Caldas* and **MLOps Engineer** at *KLYM*.
- **Speaker** at IEEE conferences, colleges, meetups, ...

# Meet the Speaker II



- *PyCon Colombia* and *Python Bogotá* **Co-organizer**.
- Former **Software Engineer** and **Technical Leader** of Machine Learning and Data Science.
- *Professor* at *UD Francisco José de Caldas* and **MLOps Engineer** at *KLYM*.
- *Speaker* at IEEE conferences, colleges, meetups, ...



# Meet the Speaker II



- *PyCon Colombia* and *Python Bogotá* **Co-organizer**.
- Former **Software Engineer** and **Technical Leader** of Machine Learning and Data Science.
- **Professor** at *UD Francisco José de Caldas* and **MLOps Engineer** at *KLYM*.
- **Speaker** at IEEE conferences, colleges, meetups, ...

# Meet the Speaker II



- *PyCon Colombia* and *Python Bogotá* **Co-organizer**.
- Former **Software Engineer** and **Technical Leader** of Machine Learning and Data Science.
- **Professor** at *UD Francisco José de Caldas* and **MLOps Engineer** at *KLYM*.
- **Speaker** at IEEE conferences, colleges, meetups, ...

# Outline

## Neural Networks

- **Artificial Neural Networks** are a computational model inspired by the **human brain**.
- They are based on a **collection** of connected units or nodes called **artificial neurons**.
- Each connection can **transmit a signal** from one artificial neuron to another.

## Neural Networks

- **Artificial Neural Networks** are a computational model inspired by the **human brain**.
- They are based on a **collection** of connected units or nodes called **artificial neurons**.
- Each connection can **transmit a signal** from one artificial neuron to another.

## Neural Networks

- **Artificial Neural Networks** are a computational model inspired by the **human brain**.
- They are based on a **collection** of connected units or nodes called **artificial neurons**.
- Each connection can **transmit a signal** from one artificial neuron to another.

## Natural Language Processing

- **Natural Language Processing (NLP)** is a subfield of linguistics, computer science, and artificial intelligence.
- It is concerned with the interactions between **computers and human language**.
- In particular, the **programming of computers to process and analyze** large amounts of **natural language data**.

## Natural Language Processing

- **Natural Language Processing (NLP)** is a subfield of linguistics, computer science, and artificial intelligence.
- It is concerned with the interactions between **computers and human language**.
- In particular, the **programming of computers to process and analyze** large amounts of **natural language data**.



## Natural Language Processing

- **Natural Language Processing** (NLP) is a subfield of linguistics, computer science, and artificial intelligence.
- It is concerned with the interactions between **computers and human language**.
- In particular, the **programming of computers** to **process and analyze** large amounts of **natural language data**.

# Inputs, Parameters, Outputs

In the context of LLMs, we can say that:

- **Inputs** are sequences of tokens.
- **Parameters** are weights and biases.
- **Outputs** are sequences of tokens.

# Foundational Models

## What are foundational models?

- **Foundational models** are the **first** and **largest** language models.
- They are **pre-trained** on **large corpora** of text.
- They could be **fine-tuned** on **specific tasks**.

# Foundational Models

## What are foundational models?

- **Foundational models** are the **first** and **largest** language models.
- They are **pre-trained** on **large corpora** of text.
- They could be **fine-tuned** on **specific tasks**.

# Foundational Models

## What are foundational models?

- **Foundational models** are the **first** and **largest** language models.
- They are **pre-trained** on **large corpora** of text.
- They could be **fine-tuned** on **specific tasks**.

```
../images/endava_break.png
```

# Outline

## What is Prompt Engineering?

- **Prompt engineering** is the process of **designing** and **constructing prompts** for **large language models**.
- Make the model **more effective** and **efficient**, also avoiding **bias** and **unwanted outputs**. . . or **hallucinations**.



## What is Prompt Engineering?

- **Prompt engineering** is the process of **designing** and **constructing prompts** for **large language models**.
- Make the model **more effective** and **efficient**, also avoiding **bias** and **unwanted outputs**. . . or **hallucinations**.

# Outline

# Chain of Thought Prompting

## Keypoints

- **Chain of Thought Prompting** is a technique that involves **prompting** a model with a **series of questions**.
- The **answers** to the **questions** are **used as inputs** to **subsequent questions**.
- This technique is **useful** for **eliciting detailed** and **coherent responses**.

# Zero-Shot Prompting

- **Zero-shot prompting** is a technique that involves **prompting** a model with a **single question**.
- The model is **expected** to **generate** a **coherent** and **relevant response**.

# Few-Shot Prompting

- **Few-shot prompting** is a technique that involves **prompting** a model with a **small number of examples**.
- This technique is **useful** for **eliciting specific** and **targeted responses**.

# Outline

# Star Coder from Hugging Faces

- **Star Coder** is a large language model developed by Hugging Face.
- It is pre-trained on a large corpus of code. Also, it is fine-tuned on a large dataset of code.
- Star Coder 2 support more than 80 programming languages.
- The largest version of this *family* has 15B parameters. There are another versions of 3B and 7B parameters.
- Those versions had been trained using between three and four trillion of tokens. The Stack is used as main source of data.

# Setup Python & Any IDE

- Here you need to have **Python** installed and **VS Code** and **Jupyter Notebook** extension.
- It works at any python version, but it is recommended to use **python 3.10** or newer.
- As IDE, it is recommended to use **VS Code**. It is FOSS, and have a lot of useful **extensions**.
- The required packages are:

```
# no one
```



# Open Assistant's Dataset... for your own chat assistant

Open Assistant's dataset is a **large dataset** of **code** that useful to make **fine-tuning** on **Star Coder**. It is **available** on **Hugging Face's model hub**.

**Open Assistant's** has more than 40000 conversations, switching roles between **human** and **assistant**.

**Open Assistant's** has a permissive licence and had been totally produced by humans.

**Link:** <https://huggingface.co/datasets/OpenAssistant/oasst1>

# Steps to Make the Tests

- 1 Create an account at **Hugging Face**.
- 2 Create a **bearer token**, and install **StarCoderEX** extension in **VS Code**.
- 3 Open in the web browser the **Code Llama Playground**.
- 4 It is time to interact with the code generators.

# Steps to Make the Tests

- 1 Create an account at **Hugging Face**.
- 2 Create a **bearer token**, and install **StarCoderEX** extension in **VS Code**.
- 3 Open in the web browser the **Code Llama Playground**.
- 4 It is time to interact with the code generators.

# Steps to Make the Tests

- 1 Create an account at [Hugging Face](#).
- 2 Create a [bearer token](#), and install [StarCoderEX](#) extension in [VS Code](#).
- 3 Open in the web browser the [Code Llama Playground](#).
- 4 It is time to interact with the code generators.

# Steps to Make the Tests

- ❶ Create an account at [Hugging Face](#).
- ❷ Create a [bearer token](#), and install [StarCoderEX](#) extension in [VS Code](#).
- ❸ Open in the web browser the [Code Llama Playground](#).
- ❹ It is time to interact with the code generators.

# Let's Give a Shoot... or a few

Please, try next **prompts**:

- 1 *"Create in Python a function to sum two numbers."*
- 2 *"Create in Python a function called my\_sum\_2 to calculate the sum of two integer parameters called num\_1 and num\_2, adding doctring to the function."*

# Let's Give a Shoot... or a few

Create a multiplication function, but with another examples.

Q. Create in Python a function to calculate the sum of two numbers.

A. 

```
def my_own_sum(num_1: int, num_2: int) -> int:
    """This method sums two numbers.
    Parameters:
    - num_1 (int): The first number to be added.
    - num_2 (int): The second number to be added.
    Returns:
    - int: The sum of the two numbers.
    """
    return num_1 + num_2
```

# Let's Give a Shoot... or a few

Create a multiplication function, but with another examples.

Q. Create in Python a function to calculate the subtraction of two numbers.

A. 

```
def my_own_subtract(num_1: int, num_2: int) -> int:
    """This method subtracts two numbers.
    Parameters:
    - num_1(int): The first number to be subtracted.
    - num_2(int): The second number to be subtracted.
    Returns:
    - int: The subtract of the two numbers.
    """
    return num_1 - num_2
```



# Outline

# Thanks!!



Linkedin: *casierrav*

`../images/endava_qa.png`

`../images/endava_panel.png`

```
../images/endava_feedback.png
```