|  |  |
| --- | --- |
| Prompt Engineering with CHATGPT  A STEEL MANUFACTURING DATA SCIENCE PROJECT | Abstract  Mini proyecto para aprender a usar ChatGPT en un proyecto de ciencia de datos de la vida real, dándole indicaciones para la planificación del proyecto, el análisis de datos, el preprocesamiento de datos, la selección de modelos, el ajuste de hiper parámetros y el desarrollo de una aplicación web.  ANED ESQUERRA ARGUELLES #1985276  DIVULGACIÓN CIENTÍFICA |

## Contexto

Con este mini proyecto, aprenderemos a usar ChatGPT en un proyecto de ciencia de datos de la vida real, usaremos indicaciones para la planificación del proyecto, el análisis de datos, el preprocesamiento de datos, la selección de modelos, el ajuste de hiper parámetros y el desarrollo de una aplicación web.

Todos conocemos la popularidad de ChatGPT y cómo los profesionales de todas las áreas lo usan para incrementar su productividad y la de sus equipos de trabajo. Como se ha mencionado previamente en este mini proyecto usaremos ChatGPT para trabajar en un caso de uso de ciencia de datos paso a paso en el cual emplearemos varias indicaciones para crear un esquema de proyecto, escribir código Python, realizar investigaciones y depurar la aplicación.

Estaremos usando un dataset de la industria del acero, específicamente en el proceso de fundición o realización de coladas, que no es más que cuando al horno de fundición se la añade chatarra ferro metálica y materia prima (HRD) para producir un planchón o bloque de metal (acero), previo a la inclusión de la llamada “receta” y especificaciones acerca del tipo de acero requerido por los clientes finales.

Este proyecto no se enfoca en la veracidad (prueba/corrida/revisión/optimización/feedback de errores) de los códigos de Python generados por el modelo, es más orientado a establecer una guía para poder interactuar con este tipo de LLMs de forma productiva que acelere el desarrollo de cualquier trabajo, en este caso particular decidimos centrarnos en un proyecto de Ciencia de datos con datos reales.

## Elementos (conceptos y términos) a conocer cuando hacemos prompting engineering

El prompting engineering es una práctica empleada en el campo del procesamiento del lenguaje natural (NLP) y específicamente en el contexto de trabajar con modelos de lenguaje como GPT-3.5. Implica diseñar y formular indicaciones o instrucciones efectivas para obtener las respuestas deseadas del modelo de lenguaje. Estos son algunos conceptos y términos clave relacionados con la ingeniería rápida:

* Aviso: un aviso es una entrada de texto o una instrucción proporcionada a un modelo de lenguaje para generar una respuesta. Establece el contexto y proporciona orientación para la salida del modelo. Las indicaciones pueden ser simples o complejas, desde una sola oración hasta varios párrafos.
* Prefijo: el prefijo se refiere a la parte inicial del mensaje que establece el contexto o el marco para la respuesta del modelo de lenguaje. Ayuda a guiar el comportamiento del modelo y se puede utilizar para especificar el formato o la estructura deseada de la salida.
* Instrucción: Las instrucciones son directivas específicas dadas al modelo para guiar su respuesta. Pueden ser explícitos, proporcionando pautas precisas, o implícitos, basándose en la comprensión del modelo de la tarea o el dominio.
* Ejemplos: los ejemplos son instancias concretas o ejemplos de entradas y salidas proporcionadas al modelo durante el entrenamiento o el ajuste. Al incluir ejemplos relevantes en el aviso, el modelo puede aprender a generalizar a partir de ellos y producir las respuestas deseadas.
* Indicadores de nivel del sistema: los indicadores de nivel del sistema definen el comportamiento de alto nivel o la personalidad del modelo de lenguaje. Ayudan a dar forma a las respuestas del modelo al establecer un contexto coherente, como pretender ser un personaje específico o adoptar un estilo de escritura particular.
* Códigos de control: los códigos de control son fichas especiales que se insertan en el indicador para influir en el comportamiento del modelo. Se pueden usar para indicarle al modelo que realice tareas específicas, cambie su estilo o altere su salida de varias maneras.
* Mensajes contextuales: los mensajes contextuales implican proporcionar al modelo un texto anterior relevante o un historial de conversación para garantizar la continuidad y la coherencia en las respuestas generadas. Ayudan al modelo a comprender el contexto en curso y generan más respuestas conscientes del contexto.
* Condicionamiento: el condicionamiento se refiere al proceso de incorporar información específica o restricciones en el aviso para guiar la respuesta del modelo. Implica estructurar la entrada de una manera que influya en la salida y haga cumplir ciertos requisitos.
* Ajuste fino: el ajuste fino implica un entrenamiento adicional de un modelo de lenguaje previamente entrenado en conjuntos de datos o tareas específicas para adaptarlo a un caso de uso o dominio particular. Permite la personalización y mejora el rendimiento del modelo en indicaciones o instrucciones específicas.
* Refinamiento iterativo: el refinamiento iterativo es un enfoque en el que el indicador se ajusta o refina gradualmente en función de los resultados anteriores del modelo. Implica un ciclo de retroalimentación iterativo para generar respuestas, evaluarlas y actualizar el aviso para lograr los resultados deseados.

El prompting engineering efectivo requiere una consideración cuidadosa de estos conceptos y términos para obtener respuestas precisas, coherentes y contextualmente apropiadas de los modelos de lenguaje. Implica experimentar con diferentes avisos, instrucciones y formatos para optimizar el comportamiento del modelo para aplicaciones o tareas específicas.

### Tipos de prompts

**Root promtp:** Es un prompt de introducción donde se provee de contexto al modelo generativo y se le indica de una manera general cual será el alcance y las especificaciones de la conversación, permite darle una “zona” de búsqueda no tan amplia para pueda dar respuestas acordes a la situación para la cual se le está usando. Ver Root prompt de este proyecto.

**Follow-up prompt:** Básicamente son prompts donde se le indica un paso específico a realizar, es una tarea en la cual se le pide que haga una acción. Ejemplos:

Write a Python section of code to load the dataset from csv file named "furnace-information.csv" and perform an extensive and detailed Exploratory data analysis (EDA) on it.

Write a Python code to perform feature engineering, using PCA to describe the 90% of dataset variance, also perform a RFECV to determine the 30 most important features of the dataset, using a tree-based algorithm as estimator.

**Update prompt:** Es un prompt que usamos cuando nos faltó u omitimos alguna indicación o especificación en el follow-up prompt y esta impacta en el resultado, con el update prompt podemos corregir esa omisión. Ejemplos:

Only include the part of the code with the hyper tuning in model X.

## Planeación

Por la extensión del root prompt lo hemos añadido al final del documento

**Prompt**: Root prompt

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

**Updated promtp:** I will be creating a web app using Gradio and deploying it on Spaces and we won't be monitoring the model in production.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Análisis de datos exploratorio (EDA)

**Follow-up prompt**: Write a Python section of code to load the dataset from csv file named "furnace-information.csv" and perform an extensive and detailed Exploratory data analysis (EDA) on it.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Preprocessing and cleaning dataset

**Follow-up prompt:** Now write a Python code to clean and preprocess the furnace dataset, include a missing values imputation with MICE technique and Outliers detection and winsorisation of outliers using Median interpolation.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Featuring Engineering

**Follow-up prompt**: Write a Python code to perform feature engineering, using PCA to describe the 90% of dataset variance, also perform a RFECV to determine the 30 most important features of the dataset, using a tree-based algorithm as estimator.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Updated prompt:** only add feature engineering part.

## Selección de los modelos.

**Follow-up prompt:** Write a follow-up python code for model selection. Try xgboost, Catboost, LightGBM, or a Multilayer perceptron neural network for regression, the target variable is 'temperature\_values'.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Afinación de los híper parámetros y evaluación del modelo

**Follow-up prompt**: Select XGBoost and write a python code for model evaluation. Use mae, mape, rmse scores as metrics and cross-validation to ensure the model is not overfitting to the training data. Also includes a section of hyperparameter hypertuning for XGBoost regressor.

A screenshot of a computer

Description automatically generated

Creación de la pagina web

**Follow-up prompt:** Write the Python code to create a Gradio web app for the Multilayer perceptron and XGboost regressors.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Updated-prompt**: Also include hyperparameter tuning in the above code and save the best-performing model for the previous Multilayer perceptron neural network add a callbacks section to early stop the training, create checkpoints saving the best model, and change the learning ratio to avoid the overfitting.

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

## Creación de una aplicación web usando Gradio

**Follow-up prompt:** Write the Python code to create a Gradio web app for the Multilayer perceptron and XGboost regressors.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Update prompt:** Write the Python code to create a Gradio web app for the Multilayer perceptron and XGboost regressors, but instead of retraining load previously saved models.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

## Deployment de la aplicación web en Hugging Face Spaces.

**Follow-up prompt:** Make a complete and detailed step-by-step workflow on deploying the Gradio app to huggingface Spaces*.*

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

**Update prompt:** *Please list a couple of online tutorials (URLs) on performing deploying the Gradio app to huggingface Spaces with more details.*

A screenshot of a computer

Description automatically generated

## Sugerencias para escribir prompts efectivos para ChatGPT

* Debemos siempre que sea posible redactar las indicaciones de manera claras y concisa, asegurándote de explicar todo en detalle desde el comienzo de la conversación.
* Planifica y crea una historia del proyecto, recuerda que ChatGPT es un chatbot por tanto debes establecer un historial de manera que pueda el chatbot pueda contextualizar el problema a resolver.
* Al no existir una forma estándar de escribir los prompts, debes iterar en la manera en la que te comunicas con el chatbot para llevar a un prompt ideal, se sugiere comenzar con un prompt root y de manera incremental mejorarlos.
* Es importante la retroalimentación sobre los errores de código que resulten de la iteración e interacción con el ChatGpT, si el código generado presenta errores a ser ejecutado es importante mencionarle el error en el mensaje de seguimiento, como parte de la retroalimentación para que al chatbot aprenda y busque mejores soluciones.
* Realice cambios manualmente en los códigos generados si considera que hay obsolescencia en las líneas que retorna, y principalmente use el chatbot para tareas comunes, ahí existe una mayor probabilidad de éxito con las asignaciones.
* De manera genaral solicítele al chatbot que explique las cosas nuevas y/o que le brinde tutoriales de cómo hacer el proceso, como medida de aprendizaje.

## Prompts

### Root prompt

I have a signals dataset consisting of 1854874 rows and 60 columns: [

'casting\_id', 'temperature\_values','measure\_date','measure\_date\_shft','start\_date','end\_date',

'duration','fecha','hf1\_vp\_an2l\_ijx\_lone\_cal\_fqi\_tot\_pv','hf1\_vp\_an2l\_ijx\_lone\_gn\_fqi\_tot\_pv',

'hf1\_vp\_an2l\_ijx\_lone\_o2e\_fqi\_tot\_pv','hf1\_vp\_an2l\_ijx\_mone\_crb\_fqi\_tot\_pv',

'hf1\_vp\_an2l\_ijx\_mone\_gn\_fqi\_tot\_pv','hf1\_vp\_an2l\_ijx\_mone\_o2e\_fqi\_tot\_pv',

'hf1\_vp\_an2l\_ijx\_mone\_o2p\_fqi\_tot\_pv','hf1\_vp\_an2l\_ijx\_pot\_cesta\_mwh',

'hf1\_sp\_an2e\_he1\_mel\_peso\_obj\_caldol','hf1\_n2\_mod\_ah\_carga\_fundida',

'hf1\_n2\_ultimo\_ppmo','hf1\_n2\_ultimo\_temp','hf1\_sp\_an2e\_alco1\_150f\_peso\_req',

'hf1\_sp\_an2e\_he1\_mel\_peso\_obj\_caldolo','hf1\_sp\_an2e\_he1\_mel\_peso\_obj\_calside',

'hf1\_sp\_an2e\_he1\_mel\_peso\_obj\_fb101','hf1\_sp\_an2e\_he1\_mel\_ref\_vel\_alim\_caldo',

'hf1\_sp\_an2e\_he1\_mel\_ref\_vel\_alim\_caldol','hf1\_sp\_an2e\_he1\_mel\_ref\_vel\_alim\_calsid',

'hf1\_sp\_an2e\_he1\_mel\_ref\_vel\_alim\_fb101','hf1\_sp\_an2l\_he1\_alc\_ref\_velocidad\_fesp',

'hf1\_vp\_an2l\_alco1\_150f1\_pv\_vel\_kg\_min','hf1\_vp\_an2l\_alco1\_150f2\_peso\_brut',

'hf1\_vp\_an2l\_alco1\_150f2\_peso\_met','hf1\_vp\_an2l\_alco1\_150f2\_pv\_vel\_kg\_min',

'hf1\_vp\_an2l\_alco1\_150f\_peso\_met\_alim','hf1\_vp\_an2l\_alco1\_176l2\_vel\_retro',

'hf1\_vp\_an2l\_alco1\_est\_desc\_tva\_sel','hf1\_vp\_an2l\_he1\_alc\_velocidad\_real\_fesp',

'hf1\_vp\_an2l\_he1\_be\_tmpo\_conectado','hf1\_vp\_an2l\_he1\_car\_ton\_acero\_oper\_ppal',

'hf1\_vp\_an2l\_he1\_com\_fact\_potencia\_prim','hf1\_vp\_an2l\_he1\_com\_pot\_activa\_primario',

'hf1\_vp\_an2l\_he1\_com\_volt\_prom\_primario','hf1\_vp\_an2l\_he1\_kwh\_colada\_anterior',

'hf1\_vp\_an2l\_he1\_mel\_peso\_acu\_caldol','hf1\_vp\_an2l\_he1\_mel\_peso\_acu\_caldolomi',

'hf1\_vp\_an2l\_he1\_mel\_peso\_acu\_calsider','hf1\_vp\_an2l\_he1\_mel\_peso\_acu\_fb101',

'hf1\_vp\_an2l\_he1\_mel\_vel\_real\_caldol','hf1\_vp\_an2l\_he1\_mel\_vel\_real\_caldolm',

'hf1\_vp\_an2l\_he1\_mel\_vel\_real\_casider','hf1\_vp\_an2l\_he1\_mel\_vel\_real\_fb101',

'hf1\_vp\_an2l\_he1\_pot\_mwh\_acumulad\_colada','hf1\_vp\_an2l\_he1\_pot\_mwh\_colada\_anterior',

'hf1\_vp\_an2l\_he1\_pot\_potencia\_real\_cd','hf1\_vp\_an2l\_he1\_pot\_tabla\_potencia\_cd',

'hf1\_vp\_an2l\_he1\_pot\_tabla\_pot\_cd','hf1\_vp\_an2l\_he1\_tmpo\_colada',

'hf1\_vp\_an2l\_ij1\_o2p\_fqi112\_tot\_pv','hf1\_vp\_n2\_hf1\_ener\_elec\_kwh\_tcm','hf1\_vp\_n2\_hf1\_tcm\_obj'

]. I would like you to list the detailed step-by-step workflow I have to follow to develop an end-to-end data science project for my portfolio.