

Prueba técnica

Teoría:

1. En la empresa GA, en el área de compras necesitan CLASIFICAR y organizar los correos que llegan a la bandeja de entrada entre 4 tipos de correos (Compras cementos, Compras energía, Compras concretos y correos generales o de otra índole). Esta tarea se le encomienda a usted, gracias a su rol puede solicitar al área interesada los recursos humanos que necesite para llevar a cabo este proyecto, también puede solicitar en tecnología todo lo que necesite, además tiene las bandejas de entrada de correos históricos de los analistas que reciben estas solicitudes con aproximadamente: 5500 correos de compras cementos, 2700 correos de compras de energía, 1100 correos de compras concretos y 12876 correos generales o de otra índole.

Explique como resolvería este problema, metodología, algoritmos, modelos, arquitectura del proyecto etc.

2. Seis meses después de haber desplegado un modelo de regresión en producción, los usuarios se dan cuenta que las predicciones que este está dando no son tan acertadas, se le encarga a usted que revise que puede estar sucediendo.

¿Cree que el modelo esté sufriendo Drift?

¿Cómo puede validarlo?

¿De ser así, que haría usted para corregir esto?

Explique sus respuestas.

3. Su equipo de trabajo está trabajando en un chatbot con generación de texto utilizando el modelo GPT-3.5, según cómo funciona este modelo, ¿cómo haría usted para hacer que las respuestas del chatbot estén siempre relacionadas a conseguir cierta información particular del usuario y no empiece a generar texto aleatorio sobre cualquier tema?

Explique su respuesta.

Practica:

Summa-sci requiere construir un servicio (Backend) para una aplicación de experiencia del colaborador, esta aplicación lo que hace es recibir en la interfaz de usuario (Frontend) unos parámetros de una lista desplegable ingresados por el usuario y con esto un valor en números que corresponde al pronóstico de la demanda de las compras de la compañía Cementos Argos, asociado a esos parámetros.

La información es enviada desde la interfaz hacia el servicio, este la recibe en formato json y utiliza estos datos para generar una clasificación (Alpha, Beta) que son códigos que le indican al analista

que tipo de compras de materiales debe hacer, y en que cantidades, finalmente el servicio debe responderle a la interfaz de usuario con un json en el que se especifica cual es el código.

Hay 3 archivo útiles para realizar este proyecto:

`dataset_demand_acumulate.csv`

Este archivo contiene la información de la demanda entre el 2017-01 hasta el 2022-04 siendo (año-mes).

`dataset_alpha_betha.csv`

Este archivo contiene la información de todas las variables involucradas para realizar la clasificación de si un registro es Alpha o Betha, este cuenta con más de 7000 registros.

`to_predict.csv`

Este archivo es lo que buscamos predecir, cuenta con 3 registros los cuales ya tienen toda la información completa, excepto la demanda y la clase.

Para este proyecto usted deberá:

1. Deberá crear un modelo que pueda pronosticar la demanda de 2022-05, 06 y 07, de Cementos Argos. El código de esto deberá hacerlo en un Jupyter notebook, el output debe ser un archivo con todos los datos incluyendo los pronósticos, además de un gráfico donde se pueda apreciar cual fue su data de entrenamiento, de validación y pronóstico, dentro de este gráfico se debe apreciar las métricas con las que decidió que su modelo estaría dando un resultado confiable (Puede usar las que considere).

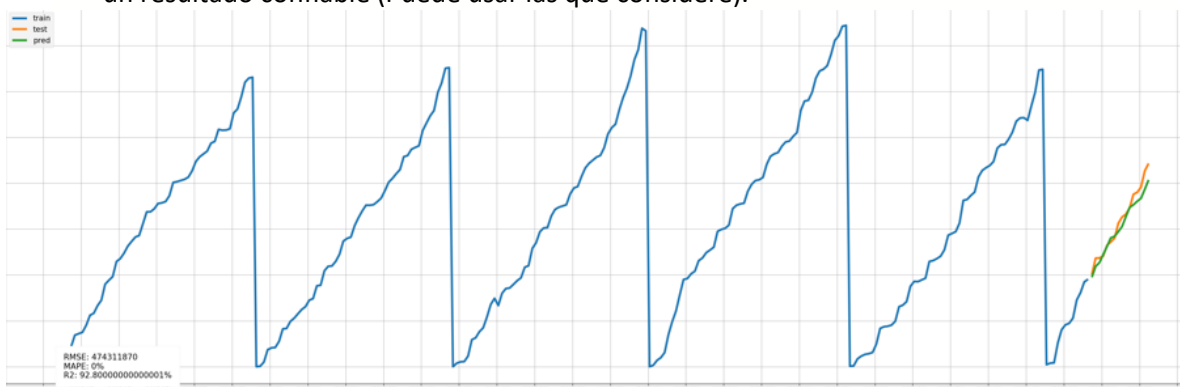


Ilustración 1. Ejemplo visualización

2. Deberá entrenar un modelo de clasificación para generar las respuestas que hará el servicio, el código deberá hacerlo en un Jupyter notebook, el output de esta ejecución será un archivo con el modelo entrenado y un txt con las métricas generadas por este, también indicaciones de que modelo usó, cantidad de datos de entrenamiento, testeo, etc. Es libre de hacer reducción de dimensionalidad, escalado, aumento de data, balanceo de data o cualquier técnica que mejore su pronóstico. (Tenga en cuenta que si redujo la

cantidad de variables predictoras para el momento de hacer el pronóstico debe hacerlo también a esos datos)

3. Deberá crear un proyecto en el cual exponga un api (Puede usar flask, FastApi, django, o cualquier librería para exponer apis) y sea ejecutable desde un archivo main.py, este proyecto deberá consumir el modelo que creó en el punto 2, recibir el json de una solicitud post, realizar la clasificación y entregar la respuesta. El entregable será el repositorio del proyecto y unas instrucciones para ejecutarlo, con archivo readme y requirements. Tanto el json del post como el del output usted puede organizarlos como quiera, solo necesita que tenga las variables que el modelo usa.
4. Complete el archivo “to_predict.csv” con la información de la demanda que pronosticó en el punto 1. Y realice las 3 peticiones al servicio que diseñó, finalmente entregue un archivo con los resultados que generó su servicio con el modelo predictivo.
5. **(Opcional) De ser posible, cree un contenedor en Docker con la aplicación, así puede ser ejecutada y desplegada en ambiente productivo de manera más sencilla, deberá entregar el Docker file (Docker compose), para que nuestro equipo haga el build y ejecute este contenedor. (Opcional)**
6. Los entregables de este proyecto deben subirse a su repositorio de github, trate de hacer buen manejo de las ramas, commits, pr, etc, y también buen manejo de las carpetas dentro del repositorio, el archivo de respuestas teóricas debe ser subido en formato pdf con el nombre “teoria_.pdf”