|  |  |
| --- | --- |
| Nombre del participante: | Eduardo Cedillo Paz |
| Fecha de elaboración: | 5 de Abril, 2024 | 11 de Abril, 2024 |
| Nombre del proyecto: | Mantenimiento Predictivo |

# Dominio de aplicación

*En función a tu trayectoria selecciona el dominio de aplicación de tu Proyecto*

|  |  |  |
| --- | --- | --- |
| **Trayectoria** | **Tipos de aprendizaje / Dominio de aplicación** | |
| Citizen Data Scientist |  | Modelación estadística |
|  | Aprendizaje automático |
| Data Scientist |  | Aprendizaje no supervisado |
| X | Aprendizaje supervisado |
|  | Analítica de texto |
|  | Analítica de redes sociales |
|  | Aprendizaje profundo para procesamiento de lenguaje (Tensor Flow / PyTorch) |
|  | Aprendizaje profundo para visión computacional (Tensor Flow / PyTorch) |
| Senior Data Scientist |  | Grandes volúmenes de datos (PySpark) |
|  | Aprendizaje por refuerzo |

|  |
| --- |
| **Meta prevista**  *Son los objetivos específicos que se establecen antes de iniciar el proyecto y que se espera lograr a través del desarrollo y la implementación de un modelo de ML.* |
| **Ejemplos:**   * Identificar los factores que impulsan la rotación o abandono de los empleados. * Desarrollar un modelo para predecir si un empleado abandonará o no la empresa/organización. * Construir un modelo que pueda predecir las ventas de cada producto en una tienda en particular. |
| Desarrollar un modelo para implementar un sistema de mantenimiento predictivo que permita predecir y prevenir fallas en los equipos antes de que ocurran, con el objetivo de reducir costos asociados con reparaciones no planificadas y mejorar la eficiencia operativa de la empresa. |

|  |  |
| --- | --- |
| **Criterio de Éxito**  *Los criterios de éxito en modelos de Machine Learning pueden variar dependiendo del contexto y los objetivos específicos del proyecto. Comúnmente pueden hacer referencia a las métricas de rendimiento (accuracy, precision, recall, F1-score, R-squared, MAE, MSE, …), al tiempo de entrenamiento o la interpretabilidad del modelo.* | |
| NOTA: Si el objetivo del proyecto es mejorar un modelo existente, es recomendable indicar el umbral que debe superarse para lograr un impacto positivo en el negocio. Por ejemplo, si el modelo actual tiene una precisión del 75%, cualquier incremento habrá de considerarse relevante. | |
| **CONTEXTO** | **CRITERIO DE ÉXITO** |
| En un proyecto de mantenimiento predictivo para equipos (clasificación de fallas)  Nota: Este proyecto será de clasificación multiclase. | ***Recall*** *(Sensibilidad):* Si bien la precisión es crucial, también se debe maximizar el recall para asegurar que se detecten la mayor cantidad posible de fallas reales en los equipos. Un alto recall garantiza que el modelo capture la mayoría de las fallas, lo que contribuye a minimizar el riesgo de tiempos de inactividad no planificados y pérdidas económicas asociadas.  ***Accuracy*** *(Precisión):* Se busca maximizar la precisión del modelo para garantizar que las predicciones de fallas sean lo más precisas posible. Esto ayudará a minimizar los falsos positivos y reducir los costos asociados con intervenciones innecesarias en equipos que no están en riesgo de falla.  ***F1-score****:* Se busca maximizar el F1-score, ya que proporciona un equilibrio entre ***accuracy y recall***. Un alto F1-score indica un modelo que logra un buen rendimiento general al identificar tanto fallas reales como no fallas, manteniendo un bajo número de falsos positivos y falsos negativos.  ***Matriz de confusión***: Se utilizará la matriz de confusión para una evaluación detallada del rendimiento del modelo en términos de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. Proporcionará información valiosa sobre cómo el modelo está clasificando las instancias de falla y no falla, lo que permitirá ajustes y mejoras adicionales según sea necesario.  **Nota:**  Dado que es más importante minimizar los falsos negativos en este contexto, la métrica más importante sería ***Recall***(Sensibilidad).  Un alto **Recall** (Sensibilidad) significa que el modelo es capaz de detectar la mayoría de las fallas reales, lo que ayuda a minimizar los falsos negativos y garantiza que se tomen medidas adecuadas cuando ocurren fallas.  Sin embargo, también es importante considerar otras métricas, como la ***Accuracy*** (Precisión), para evaluar la ***Accuracy*** (precisión) general del modelo en la clasificación de las clases positivas. |
|  |  |
|  |  |

| **Subcompetencia** | **Descripción** | **Marca lo que aplicará** | **Descripción breve de como planea aplicarlo** |
| --- | --- | --- | --- |
| **Competencia: Conceptos fundamentales de ciencia de datos** | | | |
| 1.1  Comandos de Linux y Programación Shell | * Interpretar scripts de Shell en UNIX para realizar operaciones sobre el sistema de archivos, manipulación del contenido de archivos y procesos, para navegar sobre el sistema de archivos y para automatización de tareas. * Diseñar scripts de Shell en UNIX para realizar operaciones sobre el sistema de archivos, manipulación del contenido de archivos y procesos, para navegar sobre el sistema de archivos y para automatización de tareas. |  |  |
| 1.2  Paradigma de Programación Orientado a Objetos | * Interpretar los elementos relacionados a programación orientado a objetos en programas desarrollados en lenguaje de programación Python, que cumpla con los requerimientos de la aplicación de ciencia de datos. * Desarrollar programas en lenguaje de programación Python, aprovechando las características orientado a objetos del lenguaje y que cumpla con los requerimientos de la aplicación de ciencia de datos. | X | Para este proyecto, utilizaría Programación Orientada a Objetos (POO) para organizar y gestionar las distintas partes del sistema.  Esto implica identificar las entidades clave, como equipos, procesos y tipos de fallas, y convertirlas en clases con atributos y comportamientos definidos.  La interacción entre objetos se definiría para gestionar la comunicación y el manejo de fallas. Finalmente, se implementaría la lógica de negocio y se realizarían pruebas exhaustivas para asegurar el correcto funcionamiento del sistema. |
| **Competencia: Extracción, Procesamiento y Manipulación de Datos** | | | |
| 2.1  Extracción y Transformación de Datos de Sitios Web | * Interpretar programas en lenguaje de programación Python desarrollados sobre el ambiente de programación notebook para que se realice la extracción datos de sitios WEB, y su transformación que cumpla con los requerimientos de la aplicación de ciencia de datos. * Diseñar programas en lenguaje de programación Python desarrollados sobre el ambiente de programación notebook para que se realice la extracción datos de sitios WEB, y su transformación que cumpla con los requerimientos de la aplicación de ciencia de datos. | X | Se aplicaría para recopilar datos de sitios web relevantes, como registros de mantenimiento en línea o información sobre equipos, y transformar estos datos en un formato adecuado para su análisis y modelado.  **Nota:**  En este caso como se utilizará una Data Set no sería necesario recopilar datos de una página web, pero se requerirá hacer ETL de ese Data Set. |
| 2.2  Manipulación de Datos MongoDB | * Interpretar programas con enfoque NoSQL que involucren llamadas (queries) a las bases de datos no estructuradas. * Manipular una base de datos con enfoque NoSQL que involucre llamadas (queries) a las bases de datos y estatutos para concatenar y unir bases de datos, para que se cumpla con los requerimientos de la aplicación de Ciencia de Datos. |  |  |
| **Competencia: Aplicaciones Web para Ciencia de Datos** | | | |
| 3.1  Desarrollo de Aplicaciones Web en Python, Streamlit y Firebase | * Interpretar el programa de una aplicación Dashboard en Web usando plataforma Streamlit de Python y la base de datos no relacional de Firebase para llevarla a producción. * Crear una aplicación Dashboard en Web usando plataforma Streamlit de Python y la base de datos no relacional de Firebase para llevarla a producción. |  |  |
| **Competencia: Deep Learning y Reinforcement Learning** | | | |
| 4.1  Introducción a Deep Learning con CNN, RNN y GAN | * Interpretar programas de Python sobre Keras que utilicen Deep Learning utilizando modelos pre-entrenados, análisis de Series de Tiempo y generación de datos sobre la plataforma Keras sobre el ambiente Notebook, que cumple con las especificaciones de diseño del problema a resolver. * Desarrollar programas de Python que utilicen GAN (Generative Adversial Networks) desarrollados en lenguaje Python sobre Keras sobre el ambiente Notebook, que cumple con las especificaciones de diseño del problema a resolver. | X | En problemas de clasificación supervisada se puede utilizar FCNN. Esto se analizará  **Nota:**  Es posible utilizar FCNN (Feedforward Neural Networks o Redes Neuronales de Propagación Hacia Adelante).   Puede ser efectivo para problemas de clasificación, especialmente cuando hay una gran cantidad de datos disponibles y se desea capturar relaciones complejas entre las características de entrada y las etiquetas de salida.  Sin embargo, su rendimiento puede depender de diversos factores, como el tamaño y la calidad del conjunto de datos, la arquitectura de la red y los hiperparámetros utilizados. |
| 4.2  Reinforcement Learning | * Identificar y distinguir los algoritmos, así como las aplicaciones, en el paradigma de aprendizaje por refuerzo. * Utilizar modelos de aprendizaje automático, entrenados por refuerzo, en un problema de Mercadotecnia, Publicidad, Finanzas o Manufactura. |  |  |