Fecha: 10/Marzo/23

Proyecto Bootcamp: **ML en Aviation Safety (Predictive Model)**

Nombre de Alumno: Enrique Segura

**1. Título:** Uso de ML para el Desarrollo de un Modelo Predictivo enfocado en la Reducción de la Probabilidad de Ocurrencia de Incidentes y Accidentes Aéreos desde la Perspectiva de Operaciones de Vuelo, Mantenimiento e Inspección de Aeronaves y Componentes.

**2. Etapas:** El proyecto se divide en 7 etapas.

1. Identificación del (los) Problema(s) a resolver.
2. Obtención y Recopilación de Datos. (Definir los Tipos de Datos).
3. Procesamiento y Análisis de Datos.
4. Definir el Modelo de ML (Elección de un algoritmo o conjunto de algoritmos apropiados y definir hiperparámetros).
5. Entrenamiento del Modelo.
6. Evaluación del Modelo (Validación del Modelo y la optimización de su rendimiento).
7. Implementación del Modelo.

* **Brainstorm**

- Sector Aeronáutico:

* Predicción de Peligros, Errores, Fallas, Condiciones Latentes y Amenazas que puedan generar Incidentes y Accidentes de Aeronaves en cualquier Fase de Vuelo.

- Sector Aeroespacial:

* Predicción temprana de Fallas, Errores o Condiciones Latentes que puedan afectar la integridad operacional del Cohete en cualquier Etapa de la Misión.

|  |  |  |
| --- | --- | --- |
| **Fase de la Propuesta** | **Objetivo de la Fase** | **Propuesta** |
| **Diseño del DAaaS** (Definición la estrategia del DAaaS) | *Definir el catálogo de servicios que proporcionará la plataforma DAaaS, que incluye incorporación de datos, limpieza de datos, transformación de datos, datapedias, bibliotecas de herramientas analíticas y otros.* | **Estrategia**  1. Tareas iniciales requeridas para entrar en contexto (Información y datos estadísticos iniciales):  a. Realizar Gráfico de 3 Variables con datos anuales de los últimos 8 años:  - Cantidad de Vuelos  - Tasa de Accidentalidad  - Tiempo en Años  b. Describir brevemente el Sistema Aeronáutico  c. Enfatizar la Rama (Subsector) del Sector Aeronáutico sobre la que se enfocará el Proyecto  d. Definir las Categorías de Incidentes/Accidentes (Consecuencias del Peligro).  e. Definir las Categorías de Peligro (Condicionales Latentes, Amenazas, Precursores).  f. Definir el Porcentaje de Contribución de los Peligros (Condicionales Latentes, Amenazas, Precursores).  g. Definir los Retos Técnicos (Para cada Etapa del Proyecto)  h. Definir el concepto de Detección de Anomalías (Generación de Alertas Tempranas)  2. Objetivos Principales:  a. Identificar, Analizar, Controlar y Mitigar los Peligros de Seguridad Operacional (Condiciones Latentes, Amenazas, Fallas, Errores).  b. Crear Alertas Tempranas (Preventivas-Predictivas) que permitan tomar decisiones técnicas y estratégicas. Graficar los datos obtenidos e identificar patrones y correlaciones entre los datos.  c. Disminuir la Probabilidad de Ocurrencia de Incidentes y Accidentes Aéreos (Tasa de Accidentalidad).  3. Subsectores dentro de la Industria Aeronáutica en los que se enfoca el Proyecto:  a. Operaciones de Vuelo (Aerolínea).  b. Mantenimiento e Inspección de Aeronaves y Componentes (MRO y TAR). |
| **Arquitectura DAaaS** | *Definir la selección de componentes, la definición de procesos de ingeniería y el diseño de interfaces de usuario. Diseño y ejecución de Proofs-of-Concept (PoC) para demostrar la viabilidad del enfoque DAaaS.* | **Componentes**  1. Dataset: Debido a limitantes en la *Confidencialidad de la Información*, se creará una BD tomando como referencia bases de Datos Reales de dos empresas de esos dos Subsectores.  Fuentes de información. Los Datos son obtenidos tanto en formatos físicos como digitales (almacenados en BD interna provenientes de diferentes fuentes):  a. Operaciones de Vuelo:  -FDR y CVR: Los datos migran desde los FDR y CVR hacia la BD (Descarga directa de datos de vuelo).  -Logbooks: El piloto genera los registros manualmente, son recopilados por ingeniería y mantenimiento.  b. Mantenimiento e Inspección de Aeronaves y Componentes:  -Safety Inspections and Audits (Datos obtenidos por el personal de Aviation Safety)  -MOR (Datos obtenidos por el personal de Aviation Safety proveniente del Personal Técnico, de Inspección y Pilotos)  -Voluntary Reports (Datos obtenidos por el personal de Aviation Safety proveniente del Personal Técnico, de Inspección y Pilotos)  -Inspection and Maintenance / In-Service Records (Datos obtenidos por el personal de Aviation Safety proveniente del Personal Técnico e Inspección)  2. Debido al alto volumen de datos, se dispone utilizar los recursos de Google Cloud Platform (GCP):  a. Definir Proyecto en GCP (Configurar Acceso)  b. Establecer instancia de MV en GCP (Establecer especificaciones técnicas de Software).  c. Conectar a la instancia de MV.  d. Instalar Python en instancia de MV.  e. Instalar Herramientas (Pandas, Hadoop, NumPy, Scikit-learn, TensorFlow, Matplotlib).  f. Crear instancia de Tableau Server en GCP Marketplace. (Validar el uso de Tableau Online).  g. Configurar Python (Herramientas) y Tableau para realizar la interfaz de los datos y poder visualizarlos en GCP.  En instancia de MV en GCP:   1. Pre-procesamiento, Procesamiento y Análisis de Datos: 2. Hadoop o (MongoDB\*)   -Hadoop: Procesar datos distribuidos.  b. Pandas y NumPy  -Pandas: Limpiar y preparar los datos.  -Numpy: Realizar cálculos numéricos.   1. Definir, Entrenar y Evaluar el Modelo de ML:   -Scikit-learn y TensorFlow   1. Visualización de Datos y Presentación de Resultados:   -Google Colaboratory--Python--Matplotlib  -Tableau o (Power BI\*). |
| **DAaaS Operating Model Design and Rollout** | *Personalizar los modelos operativos DAaaS para cumplir con los procesos, la estructura organizacional, las reglas y el gobierno de los clientes individuales. Realizar seguimiento de consumo y mecanismos de informe.* | Para obtener el resultado y cumplir con los procesos, se propone:  1. Obtener y reunir los datos (Dataset) extrayendo a mano los archivos (formatos: .CSV, Excel, BDSQL) desde las diferentes fuentes de información.  a. Leer datos de un archivo CSV, lanzar *Script* en *G.Colab* usando *Pandas*:  -import pandas as pd  -df = pd.read\_csv('basededatos.csv')  2. Realizar una exploración y validación preliminar de los datos para identificar patrones, correlación o datos anómalos.  a. Importar o cargar los datos en un *DataFrame* lanzar *Script* en *G.Colab* usando *Pandas*:  -import pandas as pd  df = pd.read\_csv('basededatos.csv')  b. Visualizar el comportamiento de una o más variables:  -ej. Trazar un histograma de una variable, definiendo una función *df()*.  c. Validar la relación entre dos o más variables:  -ej. Realizar un gráfico de dispersión con *plot.scatter()* y especificar los datos de las *variables (X y Y)* dentro de *df().*  3. Realizar en *G.Colab* el pre-procesamiento con *Scikit-Learn* de los datos para identificar y eliminar *null*, estandarizar *(normalizar)* los datos, entre otros:  -Importar *módulo* de *preprocessing* from *sklearn*  -Validar valores faltantes *NaN* o *null*  -Reemplazar o transformar los datos faltantes por un valor real promedio (*mean*)  -Guardar datos en variable (ej: *df\_data*)  4. Utilizar *función* de *Scikit-Learn* para separar la información en set de datos y realizar *train* y *test*.  -Importar \_*train \_test* from *sklearn*  -Cargar datos en *DataFrame* (Usando *Pandas*)  -Dividir los datos para entrenamiento y prueba (ej. Variable X[train\_, test\_])  -Definir el tamaño (*Test size*) de los datos para prueba  -Realizar *test* y validar resultados  5. Utilizar *Scikit-Learn* paraseleccionar *función* y entrenar un Modelo (ajustado al tipo de datos) y en conformidad con el problema a resolver.  -Importar *algortimo* from *sklearn*  -Seleccionar el *Modelo* conforme al *algoritmo* de prueba  -Entrenar el *Modelo* usando *función*  6. Evaluar el rendimiento.  a. Utilizar métricas de precisión dentro de *Scikit-Learn* evaluar y validar el rendimiento del *Modelo*:  -Importar *métricas, modelo (algoritmo)* from *sklearn*  -Cargar el *dataset*  -Entrenar el *Modelo*  -Definir un *set* para *test*  -Realizar *predicción del set* *(ej. predict(\_test))*  -Detección de datos anómalos o fuera de rango (Alerta Temprana)  -Definir y calcular las *métricas* sobre el *set de test*  -Imprimir las *métricas* de evaluación  b. Utilizar Keras de TensorFlow para crear un modelo basado en RN:  -Importar *tensorflow*  -Importar clase de *keras* from *tensorflow*  -Cargar el *dataset*  -Pre-procesar los datos con *función* según tipo de datos y requerimiento.  -Definir el Modelo de prueba  -Compilar el Modelo  -Entrenar el Modelo  -Evaluar el Modelo *(dataset)*  -Imprimir el Modelo (optimizado y ajustado (*precisión*))  7. Utilizar *Scikit-Learn* para optimizar y revalidar el *Modelo* mediante ajuste de *hiperparámetros*.  -Importar *función* from *sklearn* para identificar *hiperparámetros* y validar la optimización  -Cargar el *dataset*  -Definir *algoritmo*  -Ajustar y validar todas las combinaciones posibles de los valores del *Modelo*  -*Hiperparámetros* definidos (maximización de la precisión en el resultado)  8. Utilizar herramientas GCP:  a. GCMonitoring – GStackdriver para Monitorización de los Datos/Resultados, Generación de Alertas y Validación del Desempeño del Modelo y Mejoramiento Continuo.  -Establecer Dashboard  -Personalizar el Panel de Control.  -Definir Gráfico de Control y establecer las Métricas que van a ser monitoreadas.  -Configurar Alertas (Recibir notificación en tiempo real) que permitan detectar y solucionar problemas rápidamente.  - Visualizar y monitorear el rendimiento del Modelo de ML. |
| **Desarrollo de la plataforma DAaaS.**  (Ligera descripción del desarrollo) | *Construcción iterativa de todas las capacidades de la plataforma, incluido el diseño, desarrollo e integración,* ***pruebas****, carga de datos, metadatos y población de catálogos, y despliegue.* | 1. Diseño:  a. Requerimientos Normativos de Industria.  b. Componentes:  - Dataset  - Pre-procesamiento, Procesamiento y Análisis de Datos  - Definir, Entrenar y Evaluar el Modelo de ML  - Visualización de Datos y Presentación de Resultados  2. Desarrollo e integración:  a. Lenguaje de Programación: Python  b. Librerías y Herramientas:  - Referencia: PyPI (Análisis de Requerimientos de Librerías)  - Desarrollo del proyecto y análisis de datos: G.Colab, VSC, Jupyter Notebook o Spyder.  - Procesamiento de Datos: Pandas y NumPy  - Desarrollo de Modelos Predictivos: Scikit-learn y TensorFlow.  3. Pruebas:  a. Herramientas para test y filtrado: Scikit-learn, TensorFlow y Pytest  4. Carga de datos:  a. Limpieza y Procesamiento de datos: Pandas y NumPy  b. Inserción de Datos en tiempo real: Software interno, Hadoop y Apache Hive  5. Despliegue: GCP |
| **Diagrama** | *Diagrama Representación de Modelo ML* | Diagrama\_BDArchitecture\_(Proyecto ML).drawio |