Departamento de Ingeniería Industrial

UNIVERSIDAD DE ANTIOQUIA
1803
Facultad de Ingeniería

Análisis de Logs de las APIs de TABi Connect usando Machine Learning

PRACTICANTE: Jhon Alexander Bedoya Carvajal

ASESORES: Emerson Giraldo y Pablo Saldarriaga

PROGRAMA: Ingeniería Industrial

MODALIDAD DE PRÁCTICA: Semestre de Industria

TABi Connect es una plataforma innovadora desarrollada por Hubtek que simplifica y optimiza las operaciones logísticas al automatizar tareas repetitivas y de gran volumen. Esta solución permite a los proveedores de servicios logísticos capturar y analizar el 100% de las solicitudes de tarifas (cotizaciones) que fluyen a través de sus procesos, asegurando mayor eficiencia operativa y reduciendo la carga manual en actividades clave.

Introducción

TABi Connect optimiza operaciones logísticas automatizando tareas repetitivas, incluyendo la gestión de cotizaciones a través de APIs. Sin embargo, el monitoreo manual de logs es ineficiente debido a su alto volumen y complejidad. Este proyecto busca implementar un sistema basado en IA y Machine Learning No Supervisado para analizar los logs, optimizando la identificación de problemas y reduciendo costos operativos. Se emplea la metodología CRISP-DM para estructurar el desarrollo, desde la comprensión del negocio hasta la evaluación de modelos. A pesar de desafíos como el almacenamiento y la interpretación de patrones, la solución propuesta promete mejorar el monitoreo de APIs y la eficiencia operativa.

Objetivos

- Comprender la importancia de los logs de las APIs de TABi Connect y cómo un análisis automatizado y una rápida identificación de problemas puede beneficiar a la empresa..
- Implementar un pipeline de datos que extraiga, transforme y cargue los logs a un entorno de análisis adecuado desde Amazon CloudWatch.
- Ajustar un modelo No Supervisado de Machine Learning para la identificación de problemas en los logs de las APIs de TABi Connect.
- Ajustar un modelo No Supervisado de Machine Learning para la identificación de problemas en los logs de las APIs de TABi Connect.

Metodología

El proyecto se desarrolla siguiendo un enfoque cuantitativo, debido a la naturaleza del Análisis de Datos y el uso de modelos de Machine Learning. La metodología cuenta con cuatro etapas clave de la metodología CRISP-DM:

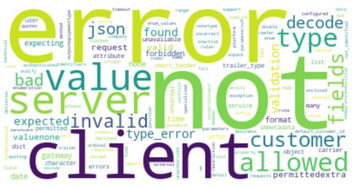
Fase	Objetivo	Actividades
1. Comprensión del problema	Comprender la importancia de los logs de las APIs de TABi Connect	Acceder a los logs en Amazon Cloudwatch
		Estudiar el proceso actual de identifiación de problemas en las APIs usando los logs
2. Entendimiento y preparación de datos	Implementar un pipeline de datos que extraiga, transforme y cargue los logs a un entorno de análisis adecuado	Conectarse a Amazon CloudWatch desde Python usando Boto3
		Extraer los logs en Python y almacenarlos en formato parquet
		Limpiar y transformar los logs en Python
		Realizar un análisis exploratorio de los logs ya transformados
3. Modelamiento	Ajustar un modelo No Supervisado de Machine Learning para la identificación de problemas en los logs	Preprocesar los logs para llevarlos a un formato adecuado para el ajuste del modelo
		Ajustar modelos No supervisados de Machine Learning
		Evaluar necesidad de hacer ajustes en la transformación y preprocesamiento de datos
		Evaluar modelos y seleccionar el mejor
4. Validación del modelo	Analizar y evaluar los resultados obtenidos del modelo No Supervisado de Machine Learning seleccionado	Analizar y evaluar los resultados obtenidos del modelo No Supervisado de Machine Learning seleccionado
	Concluir si el modelo tiene un buen desempeño y permite mejorar el proceso de identifiación de problemas en las APIs	Concluir si el modelo tiene un buen desempeño y permite mejorar el proceso de identifiación de problemas en las APIs

Resultados

Log documento:

response db 400 json *decode error expecting value db_api_gotabi_ai*

Bag of Words



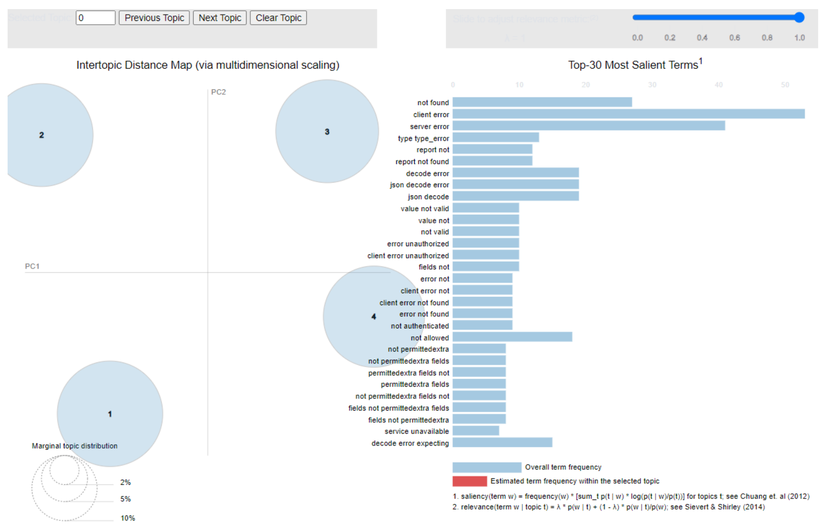
TF-IDF

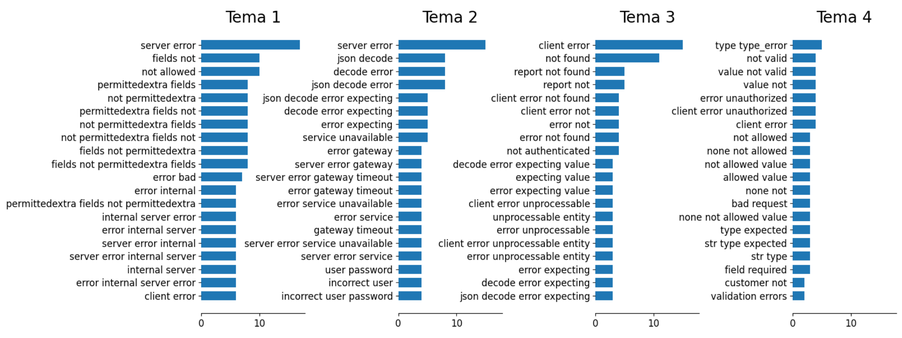
Clustering - DBSCAN

Técnica	eps	min_samples	silhouette	calinski	clusters
Bag of Words	7	2	0,900344	17974,7	5
TF-IDF	7	3	0,900461	17965,7	5
Word2Vec	7	2	0,90097	18062,7	4

Buen rendimiento técnico, pero el análisis de los clusters con conocimiento del negocio no muestra patrones coherentes ni interpretables.

Topic Modeling - LDA





- **Tema 1:** Errores de Validación de Datos
- **Tema 2:** Excepciones Internas en el Procesamiento
- **Tema 3:** Errores de Servidor y Respuestas Críticas
- **Tema 4:** Errores en Solicitudes Externas

Conclusiones

- Se logró tener un entendimiento y comprensión del proceso de monitoreo de las APIs que permitió diseñar una solución basada en Machine Learning No Supervisado para optimizar dicho monitoreo.
- A pesar de las limitaciones debido al gran volumen de información, fue posible diseñar e implementar un pipeline que extraiga los logs desde CloudWatch Python.
- El modelado de temas con LDA ofreció una segmentación clara y útil desde la perspectiva del negocio, mientras que Clustering con DBSCAN mostró limitaciones en la interpretación de sus resultados.
- Una correcta jerarquización de los errores permite priorizar su resolución, evitando que problemas iniciales escalen a fallos críticos.
- Se requiere de patrones menos generales para lograr que el modelo logre un proceso de monitoreo de las APIs más automático o que requiera mínima intervención humana.

DATOS DE CONTACTO DEL AUTOR:

+57 3106033231 Alexander.bedoya1@udea.edu.co https://www.linkedin.com/in/jhon-alexander-bedoya-carvajal-17a724202/