

Informe de Análisis por Niveles

Tipo de Proyecto: Machine Learning

1. Estadísticas del Repositorio

Total Commits: 301

Lenguajes: Jupyter Notebook (57.85%), HTML (41.9%), Python (0.25%), Dockerfile (0.0%)

2. Análisis por Niveles

Nivel Esencial

Compleitud: 75%

Requisitos Cumplidos:

- ✓ Un modelo de ML funcional que prediga el grado de satisfacción de los clientes
- ✓ Un análisis exploratorio de los datos.
- ✓ Una solución que productivice el modelo (Una aplicación de Streamlit)
- ✓ Un informe del rendimiento del modelo con sus métricas y la explicación de su pe

Requisitos Faltantes:

- ✗ Overfitting inferior al 5%

Nivel Medio

Compleitud: 0%

Requisitos Faltantes:

- ✗ Un modelo de ML con técnicas de ensemble
- ✗ Uso de técnicas de Validación Cruzada.
- ✗ Optimización del modelo escogido con técnicas de ajuste de hiperparámetros
- ✗ Un sistema de recogida de feedback que monitorice la performance del modelo en p
- ✗ Un sistema de recogida de datos nuevos para futuros entrenamientos.

Nivel Avanzado

Compleitud: 0%

Requisitos Faltantes:

- ✗ Una versión dockerizada del programa.
- ✗ Guardado en bases de datos de los datos recogidos por la aplicación
- ✗ Despliegue en Cloud de las soluciones aportadas.
- ✗ Incluir test unitarios.

Nivel Experto

Compleitud: 0%

Requisitos Faltantes:

- ✗ Experimentos o despliegues con modelos de redes neuronales.
- ✗ Sistemas de entrenamiento y despliegue automático de nuevas versiones del modelo

3. Análisis Técnico

Calidad Código: Se observa un código claro y bien estructurado en la aplicación de Streamlit.

Mejores Practicas: Se observan algunas mejores prácticas, como la separación de entrenamientos y validación.

Experimentacion: Se implementan diferentes modelos para clasificar la satisfacción del cliente.

Manejo Datos: Se realiza un preprocesamiento adecuado de los datos, incluyendo la normalización.

Optimizacion: No se observa una evaluación explícita del overfitting en los modelos, lo que podría mejorar.

Etica Sesgos: No se observan consideraciones éticas o de sesgo en el proyecto. Se recomienda evaluarlos.

4. Recomendaciones

- Evaluar el overfitting en el modelo final y tomar medidas correctivas.
- Mejorar la documentación en el código fuente.
- Implementar técnicas de ensemble y validación cruzada para mejorar el rendimiento.
- Optimizar los hiperparámetros de los modelos utilizados.
- Agregar sistemas de feedback y recolección de datos para futuros entrenamientos.
- Dockerizar el proyecto para facilitar el despliegue y la escalabilidad.
- Almacenar los datos recogidos en una base de datos.
- Desplegar las soluciones en la nube.
- Agregar test unitarios para mejorar la calidad del código.
- Experimentar con modelos de redes neuronales y evaluar su rendimiento.
- Implementar sistemas de entrenamiento y despliegue automático de nuevas versiones.
- Evaluar y documentar posibles sesgos en los datos y el modelo.

Puntuación de Madurez: 65/100