Taller 3 MLOPS

Eldilgardo Camacho
Linda Castaño

Punto 1

1.a. Comparación de modelos (usando PyCaret y Scikit-learn):

En el taller pasado se compararon modelos tanto con PyCaret como con Scikit-learn. Los resultados clave fueron:

- **Con PyCaret**:
  - El mejor modelo fue **Extra Trees Classifier**, con un **Combined Score** de **0.8347**, destacando en Recall (87.12%), F1 Score (85.93%), y AUC (91.29%). Este modelo tiene ventajas en datos complejos y alta capacidad de generalización.
- **Con Scikit-learn**:
  - La **Regresión Logística** tuvo el mejor desempeño con **Accuracy: 86%**, **Recall: 88%**, y **F1 Score: 86.27%**, mostrando un equilibrio sólido entre precisión y sensibilidad. Este modelo es más interpretable y fácil de desplegar.

1.b. Tres mejores modelos (hiperparámetros):

1. **Extra Trees Classifier** (PyCaret):
   - No requiere ajuste significativo; funciona bien por su diseño aleatorio.
2. **Regresión Logística** (Scikit-learn):
   - Ajustar parámetros como C=5.682 y el solver lbfgs.
3. **Random Forest**:
   - Mejora notable tras ajustes en max_depth=20, min_samples_split=5, y n_estimators=50.

1.c. Definición del mejor modelo:

- **Extra Trees Classifier (PyCaret)** fue el mejor por:
  - Balance entre métricas clave.
  - Robustez ante datos complejos y reducción de sobreajuste.
  - Menor tiempo de entrenamiento (0.036 segundos).

1.d. Resultados en Scikit-learn:

- Aunque la regresión logística tuvo un desempeño competitivo, **Extra Trees** fue superior al capturar interacciones complejas entre variables.

Punto 2 - Implementación con MLFlow

Explicación de los Resultados Obtenidos

1. División de Datos

- **Columna Objetivo**: Usamos Bankrupt? como variable objetivo (y) para predecir si una empresa se declara en quiebra.

- **División**: Se dividió el dataset en un 80% para entrenamiento y un 20% para prueba, asegurando que el modelo se entrene con una mayoría de los datos y valide su desempeño en un conjunto separado.

---

2. Entrenamiento Inicial

- **Modelo**: Se utilizó un modelo de **Random Forest** con hiperparámetros básicos.
- **Métricas Iniciales**:
    - **Accuracy (97%)**: El modelo clasificó correctamente la mayoría de los datos. Sin embargo, esta métrica no es suficiente en un dataset desbalanceado.
    - **Recall (20%)**: El modelo solo identificó el 20% de las empresas en quiebra correctamente, lo cual es bajo para un problema donde los falsos negativos (no detectar quiebras) son críticos.
    - **F1 Score (31%)**: Muestra un balance limitado entre precisión y recall, indicando que el modelo necesita ajustes.

Observación: El alto accuracy puede estar influido por el desbalance de clases (muchas más empresas no en quiebra). En este caso, métricas como recall y F1 son más importantes.

---

3. Optimización con Optuna

- **Proceso de Optimización**:
    - Optuna buscó los mejores hiperparámetros (max_depth, min_samples_split, y n_estimators) mediante validación cruzada para maximizar el desempeño del modelo.
    - **Mejores Parámetros**:
        - max_depth: 27
        - min_samples_split: 9
        - n_estimators: 107
- **Métricas del Modelo Optimizado**:
    - **Accuracy (97%)**: Se mantuvo igual, indicando que el modelo sigue clasificando correctamente la mayoría de las empresas.
    - **Recall (14%)**: Bajó en comparación con el modelo inicial, lo cual es preocupante ya que implica que menos empresas en quiebra fueron identificadas.
    - **F1 Score (24%)**: También disminuyó, reflejando un desequilibrio entre precisión y recall.

Observación: Aunque Optuna logró optimizar los hiperparámetros, el modelo sigue teniendo dificultades para manejar el desbalance de clases y mejorar el recall.

---

4. Registro en MLFlow

- **Modelos Registrados**:
    - Se registraron tanto el modelo inicial como el modelo optimizado en **MLFlow**, junto con sus métricas. Esto facilita el análisis y la comparación entre diferentes versiones.

Conclusión

1. **Desempeño General**:
    - El modelo muestra un alto accuracy debido al desbalance de clases, pero su recall bajo es preocupante para este caso.
    - El objetivo principal debería ser mejorar el recall, ya que es crucial identificar correctamente las empresas en riesgo de quiebra.
2. **Modelo Optimizado**:
    - Aunque Optuna encontró parámetros que maximizan accuracy, no mejoró significativamente el recall ni el F1 Score, mostrando que el desbalance de clases aún afecta el desempeño.
3. **Siguientes Pasos**:
    - **Manejo del Desbalance**:
        - Usar técnicas como sobremuestreo (SMOTE) o submuestreo para equilibrar las clases.
    - **Modelos Alternativos**:
        - Probar con algoritmos como Gradient Boosting o LightGBM, que manejan mejor datos desbalanceados.
    - **Optimización del Recall**:
        - Ajustar la métrica objetivo en Optuna para priorizar el recall en lugar del accuracy.