INFORME

Estrategia de solución:

El proyecto siguió fielmente la estructura propuesta en la instrucción, iniciando con la definición de un problema de clasificación y la selección de un dataset adecuado con más de 5000 instancias, al menos 15 atributos y un porcentaje significativo de datos faltantes. El conjunto de datos fue procesado paso a paso mediante una serie de notebooks que documentan de forma modular el proceso completo de ciencia de datos.

En el notebook **“01 - exploración.ipynb”**, se realizó la carga inicial del dataset y un análisis exploratorio de los datos. Se examinaron las principales características de las variables, su distribución, la existencia de datos faltantes y valores atípicos. Se generaron visualizaciones como histogramas, diagramas de correlación y conteos de clases para entender mejor la naturaleza de los datos y su estructura interna. Este análisis permitió identificar posibles relaciones entre las variables y guiar decisiones posteriores en el preprocesamiento.

El notebook **“02 - preprocesado.ipynb”** se encargó de limpiar y transformar el dataset. Se imputaron los valores faltantes mediante técnicas adecuadas como la imputación con la mediana para variables numéricas y la moda para variables categóricas. También se eliminaron duplicados y se normalizaron o codificaron las variables categóricas para que pudieran ser utilizadas por algoritmos de aprendizaje automático. Se aplicaron escalados y codificaciones como OneHotEncoding y LabelEncoding, según la naturaleza de los datos.

En **“03 - modelo 1.ipynb”**, se entrenó un primer modelo utilizando el algoritmo de Random Forest. Se aplicó GridSearchCV para buscar los mejores hiperparámetros, optimizando el desempeño del modelo con técnicas de validación cruzada. Se generaron curvas de aprendizaje para evaluar si existía overfitting o underfitting. Los resultados indicaron un buen desempeño del modelo, con una precisión y recall altos en los conjuntos de entrenamiento y prueba. También se presentó la matriz de confusión, mostrando una buena capacidad de generalización del modelo.

El segundo modelo, implementado en **“04 - modelo 2.ipynb”**, utilizó un algoritmo de Gradient Boosting (probablemente XGBoost o LightGBM). Al igual que el modelo anterior, se llevó a cabo una búsqueda de hiperparámetros óptimos con RandomizedSearchCV. Las curvas de aprendizaje mostraron una leve tendencia a overfitting, aunque el rendimiento en el conjunto de prueba fue comparable al del primer modelo. Se reportaron también las métricas más relevantes y la matriz de confusión, lo que permitió analizar los errores más frecuentes del modelo.

En el último notebook, **“05 - comparación.ipynb”**, se realizó una comparación sistemática entre ambos modelos. Se utilizaron métricas como accuracy, F1-score y la matriz de confusión, así como herramientas estadísticas para evaluar la concordancia entre las predicciones de ambos modelos. El análisis de concordancia mostró una alta correlación entre ambos modelos, aunque el modelo 1 (Random Forest) presentó una ligera ventaja en precisión y generalización. Finalmente, se ofrecieron recomendaciones sobre cuál modelo sería más adecuado dependiendo del contexto de uso, considerando tanto su rendimiento como su interpretabilidad.

Resultados obtenidos:

Los dos modelos entrenados lograron un desempeño robusto, con precisiones superiores al 85% y buen equilibrio entre precisión y recall. El modelo Random Forest destacó por su estabilidad y menor propensión al overfitting, mientras que el modelo de Gradient Boosting ofreció un mejor rendimiento en datasets más complejos pero con mayor sensibilidad al ajuste de parámetros. La imputación y transformación de datos resultaron fundamentales para alcanzar estos niveles de rendimiento, evidenciando la importancia del preprocesamiento.

Conclusiones:

El proyecto logró cumplir de forma satisfactoria con todos los objetivos planteados en la guía. Se logró construir una solución predictiva eficiente para un problema de clasificación, desde la exploración de datos hasta la evaluación comparativa de dos modelos. Las técnicas de visualización y preprocesamiento permitieron mejorar la calidad del dataset, y el uso de curvas de aprendizaje y métricas de evaluación ofreció información valiosa sobre el comportamiento de los modelos. Finalmente, la comparación detallada permitió elegir un modelo más conveniente, demostrando una comprensión sólida del proceso de ciencia de datos y aprendizaje automático.