Informe de Aprendizaje Automático

Chavez, Mauro
a@gmail.com

Lewkowicz, Iván
a@gmail.com

Drelewicz, Santiago
a@gmail.com

Torrez, Matías
matiastorrez157@gmail.com

Culaciati, Dante
a@gmail.com

28 de abril de 2025

Índice

1. Ejercicio 1

En este inciso se pide separar los datos en conjuntos de entrenamiento y evaluación, donde no se debe utilizar la libreria `train_test_split` de `sklearn`.

Primero se realizo una exploracion de los datos, donde se observa que el dataset posee 200 features, todas numericas, y 500 filas. Se observa que el dataset no tiene valores nulos y que ademas se trata de un problema desbalanceado, donde el %70 de los datos pertenecen a la clase 1 y el % restante pertenece a la clase 0, por lo que no es necesario realizar un preprocesamiento de los datos. Se decide entonces utilizar el 80 % de los datos para entrenamiento y el 20 % restante para evaluacion.

Como la proporción de los datos es desbalanceada, realizamos un `stratified split` en la separación de los datos, procurando mantener la proporción del dataset original para los datos de entrenamiento y evaluación.

2. Ejercicio 2

Para la primera parte de este ejercicio, entrenamos un arból de decisión con altura máxima 3 y estimamos la performance del modelo con K fold cross validation para distintas métricas. Las metricas utilizadas son *Accuracy*, *AUPRC* y *AUC ROC* y se realizo un *K-fold* con $K = 5$.

En la tabla 1 se muestran los resultados obtenidos para cada una de las 5 permutaciones de los datos, asi como el promedio de cada métrica para todas las permutaciones y el resultado global, el cual se obtiene al calcular las metricas utilizando el conjuto de predicciones formado a partir de concatenar las predicciones de cada fold.

Permutación	**Accuracy** (training)	**Accuracy** (validación)	**AUPRC** (training)	**AUPRC** (validación)	**AUCROC** (training)	**AUCROC** (validación)
1	0.8125	0.6375	0.6710	0.3226	0.8058	0.5298
2	0.8406	0.5875	0.7337	0.3337	0.8458	0.5246
3	0.825	0.6875	0.6431	0.3437	0.7513	0.5811
4	0.8188	0.7	0.6573	0.3626	0.7877	0.5938
5	0.8438	0.65	0.6958	0.4144	0.8085	0.5967
Promedios	0.8281	0.6525	0.6802	0.3554	0.7998	0.5651
Global	(NO)		(NO)		(NO)	

Cuadro 1: Resultados por permutación y métricas

Se observa que este modelo presenta un buen desempeño en el conjunto de entrenamiento, pero su desempeño en el conjunto de validación es bastante bajo, lo que podría indicar que el modelo está sobreajustado a los datos de entrenamiento.

Para la segunda parte del ejercicio, se exploraron diferentes combinaciones de hiperparámetros para el modelo de árbol de decisión, utilizando `GridSearchCV` de `sklearn`. Se probaron diferentes valores para la profundidad máxima del árbol y el cirterio de corte. Se utilizó `StratifiedKFold` con $K = 5$ para la validación cruzada. En la tabla 2 se muestran los resultados obtenidos para cada combinación de hiperparámetros, así como el promedio de *Accuracy* para cada combinación.

Altura máxima	**Criterio de corte**	**Accuracy** (training)	**Accuracy** (validación)
3	Gini	0.6375	0.6710
5	Gini	0.5875	0.7337
Infinito	Gini	0.6875	0.6431
3	Entropía	0.7	0.6573
5	Entropía	0.65	0.6958
Infinito	Entropía	0.828125	0.828125

Cuadro 2: Resultados por permutación y métricas

3. Ejercicio 3

En el tercer ejercicio compararemos distintas combinaciones de algoritmos, con diferentes configuraciones a fin de encontrar el mejor modelo para cada familia de algoritmo. La métrica a usar para evaluar estas combinaciones será AUCROC. Por otro lado, para estimar la performance utilizaremos Nested Cross Validation, ya que de esta forma no elegimos hiperparámetros y entrenamos sobre el mismo fold. En su lugar para cada modelo a evaluar, realizamos *Stratified-CV* sobre nuestros datos, y a su vez realizamos en cada fold *Stratified 4-Fold* nuevamente pero esta vez para probar distintos hiperparámetros. Luego elegimos los mejores hiper parámetros y evaluamos en un fold externo.

Árboles de decisión

En nuestra búsqueda de hiperparámetros, tuvimos en cuenta las siguientes posibilidades:

- Función de ganancia: Gini, Entropía
- Splitter: Best, Random
- Máxima Profundidad: Números aleatorios entre 1 y 20
- Cantidad mínima de ejemplares por división: Números aleatorios entre 1 y 20
- Cantidad mínima de ejemplares por hoja: Números aleatorios entre 1 y 20
- Pesos por clase: Ninguno, Balanceados

Luego de realizar Nested CV con 500 iteraciones, encontramos que la mejor combinación de hiperparámetros es la siguiente

Hiperparámetro	Valor
Class Weight	Balanced
Criterion	Entropy
Max Depth	19
Min Samples Leaf	16
Min Samples Split	8
Splitter	Best
ROC AUC Promedio	0.581
Varianza	0.069
ROC AUC Test	0.662

Cuadro 3: Mejores hiperparámetros encontrados para árboles de decisión.

KNN

Para KNN encontramos que la mejor combinación de hiperparámetros es la siguiente

Hiperparámetro	Valor
Metric	Manhattan
N-Neighbors	9
P	1
Weights	distance
ROC AUC Promedio	0.859
Varianza	0.063
ROC AUC Test	0.836

Cuadro 4: Mejores hiperparámetros encontrados para KNN.

Support Vector Machines

Para SVM encontramos que la mejor combinación de hiperparámetros es la siguiente

Hiperparámetro	Valor
C	141.9443
Class Weight	Ninguno
Gamma	0.0001655355812491995
Kernel	Radius Based Function
ROC AUC Promedio	0.921
Varianza	0.020
ROC AUC Test	0.914

Cuadro 5: Mejores hiperparámetros encontrados para SVM.

4. Ejercicio 4

Presente los resultados obtenidos, como métricas de evaluación, gráficos de desempeño, etc.

5. Ejercicio 5

Analice los resultados, las limitaciones del modelo y posibles mejoras.

6. Conclusión

Resuma los hallazgos principales y las conclusiones del informe.

Referencias

Incluya las referencias bibliográficas utilizadas en el informe.