I302 - Aprendizaje Automático y Aprendizaje Profundo

Trabajo Práctico 2: Clasificación y Ensemble Learning

Ignacio Elian Gremes

14 de abril de 2025

Ingeniería en Inteligencia Artificial

1. Diagnóstico de Cáncer de Mama

Resumen

En este trabajo se busco modelar una regresion logistica capaz de predecir el diagnostico de cancer de mama de un panciente. Las muestras obtenidas para modelar esto estaban des balanceadas, por lo tanto ser probo distintos metodos de re balanceo y se estudio cual performaba mejor. Las metricas indicaron que el mejor metodo es el cost re-weighting.

1.1. Introducción

Para este trabajo la entrada del algoritmo fue las características de células de los pacientes. Se utilizo un modelo de regresión logística binaria con regularización L2 para obtener una predicción del diagnosis de cáncer de mama de una paciente, a partir de las características de sus células. Lo que se busco resolver en este trabajo es que método utilizar para re balanceo de clases cuando se trabaja con muestras des balanceadas ya que en este contexto hay una mayor cantidad de muestras con diagnostico negativo que positivo. En este experimento se estudio re balanceo por medio de Undersampling, Oversampling por duplicacion, Oversampling por SMOTE y Cost re-weighting.

1.2. Métodos

1.3. Algoritmos de Clasificación

1.3.1. Regresión Logística

La regresión logística en su forma binaria, modela la probabilidad de que una observación pertenezca a la clase positiva frente a la clase negativa. La probabilidad se expresa mediante la función sigmoidea:

$$P(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}+b)}}$$

donde $\mathbf{x}$ es el vector de características, $\mathbf{w}$ es el vector de pesos, y b es el sesgo. Los parámetros $\mathbf{w}$ y b se estiman minimizando una función de pérdida, típicamente la entropía cruzada, utilizando métodos como el descenso de gradiente.

En este trabajo, se implementó una regresión logística con regularización L2 para evitar el sobreajuste.

1.4. Técnicas de Remuestreo para Clases Desbalanceadas

1.4.1. Submuestreo (*Undersampling*)

El submuestreo consiste en reducir el número de muestras de las clases mayoritarias para igualarlas al número de muestras de la clase minoritaria.

1.4.2. Sobremuestreo por Duplicación (*Oversampling by Duplication*)

El sobremuestreo por duplicación aumenta el número de muestras de las clases minoritarias duplicando aleatoriamente las muestras existentes hasta igualar el tamaño de la clase mayoritaria.

1.4.3. Sobremuestreo con SMOTE

SMOTE (*Synthetic Minority Oversampling Technique*) es una técnica de sobremuestreo que genera muestras sintéticas para las clases minoritarias. Para cada muestra de una clase minoritaria, SMOTE selecciona k vecinos más cercanos (usando la distancia euclidiana) y genera una nueva muestra como una combinación lineal de la muestra original y uno de sus vecinos. En este trabajo, se aplicó SMOTE con $k = 5$.

1.4.4. Re-ponderación de Costos (*Cost Re-weighting*)

La re-ponderación de costos ajusta el peso de las muestras durante el entrenamiento para dar mayor importancia a las clases minoritarias.

$$w_k = \frac{1}{\text{frecuencia de la clase } k}$$

donde w_k es el peso de la clase k.

1.5. Métricas de Evaluación

1.5.1. Exactitud (*Accuracy*)

$$\text{Exactitud}_{\text{macro}} = \frac{1}{K} \sum_{k=1}^{K} \frac{\text{Correctas}_k}{\text{Total}_k}$$

donde K es el número de clases.

1.5.2. Precisión (*Precision*)

$$\text{Precisión}_{\text{macro}} = \frac{1}{K} \sum_{k=1}^{K} \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k}$$

donde K es el número de clases

1.5.3. Recall

$$\text{Recall}_{\text{macro}} = \frac{1}{K} \sum_{k=1}^{K} \frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k}$$

donde K es el número de clases

1.5.4. F-Score

El F-score es la media armónica de la precisión y el recall, con un parámetro β que pondera la importancia del recall frente a la precisión. En este trabajo, se usó $\beta = 1$ (F1-score).

$$\text{F1}_{\text{macro}} = \frac{1}{K} \sum_{k=1}^{K} \text{F1}_k, \quad \text{F1}_k = 2 \cdot \frac{\text{Precisión}_k \cdot \text{Recall}_k}{\text{Precisión}_k + \text{Recall}_k}$$

1.5.5. AUC-ROC

$$\text{AUC-ROC}_{\text{macro}} = \frac{1}{K} \sum_{k=1}^{K} \text{AUC-ROC}_k$$

1.5.6. AUC-PR

$$\text{AUC-PR}_{\text{macro}} = \frac{1}{K} \sum_{k=1}^{K} \text{AUC-PR}_k$$

Ambas métricas AUC se calcularon utilizando las probabilidades predichas por los modelos, lo que permite evaluar su capacidad de clasificación en diferentes umbrales.

1.6. Conjunto de datos e hiperparametros

Para este experimento se utilizaron los datasets, 'cell diagnosis dev imbalanced' y 'cell diagnosis test imbalanced' para el estudio de clases des balanceadas y 'cell diagnosis dev' y 'cell diagnosis test'. Para entrenamiento se utilizaron los 'dev' y se los separo en entrenamiento y validación, con una proporción de 80 % y 20 % respectivamente. Tambien se utilizo KNN con k = 3, para reemplazar Nans ya que la cantidad de Nans era muy alta, no era viable eliminar las muestras con datos faltantes y tampoco iba a ser beneficio reemplazar por la mediana ya que era muy alta la cantida de datos faltantes y podria generar un sesgo en el modelo. Ademas se normalizo por medio de la media y la desviación estándar. Tambien se removió los outlayers por medio de boxplots con Q1 = 0.25 y Q3 = 0.75. El hiperparametro utilizado para regularizaion L2 fue de 0.01, y para la regresión se utilizo un learing rate de 0.01 y 1000 iteraciones max. Para oversampling SMOTE se uso K = 5 y para cost re-weighting se utilizo cost fn = 2.

1.7. Resultados

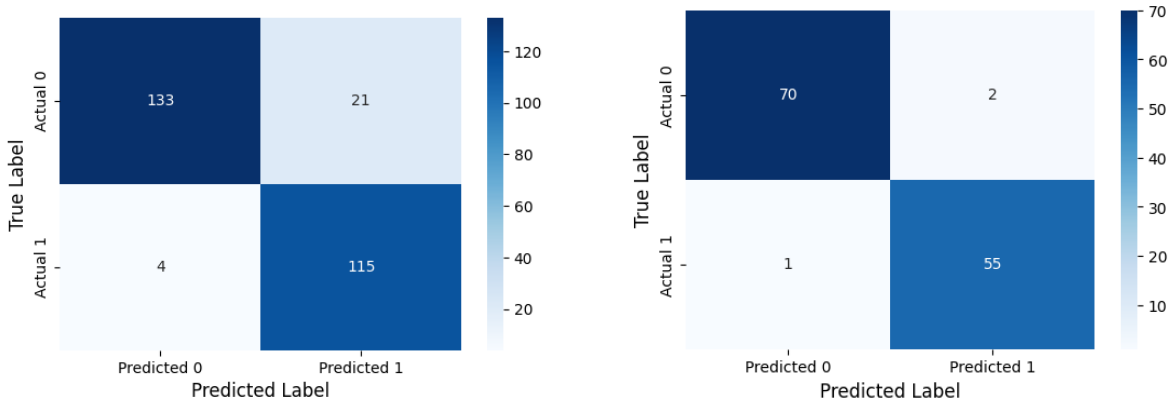
1.7.1. Clases balanceadas

Las metricas obtenidadas en el set de validation y de test son las siguientes:

Modelo	Accuracy	Precision	Recall	F-Score	AUC-ROC	AUC-PR
Validation	0.908	0.846	0.966	0.902	0.937	0.887
Test	0.977	0.965	0.982	0.973	0.972	0.971

Figura 1: Métricas obtenidas en el data set de clases balanceadas

Se obtuvieron mejores resultados en el test que en el validation, esto se puede deber a la diferencia de cantidad de muestras entre ambos sets, ya que el test set tiene bastante menos muestras que el validation set y puede que eso haya ocasionado que el modelo se confunda menos.



(a) Matriz de confusión de set de validation

(b) Matriz de confusión de set de test

Figura 2

Como se puede ver en las matrices de confusión, en el set de validation el modelo clasifico erroneamente algunos labels de clase 0, y esto en el set de test no ocurrio. Esto se puede deber al pequeño tamaño que tiene el set de test. Puede que algunas muestras presentes en el set de validation le generen algunos problemas a la hora de clasificarlo y justo ese tipo de muestras no estan presentes en el set de test. Al tener ambos sets muy pocas muestras la aleatoriedad entra en juego.

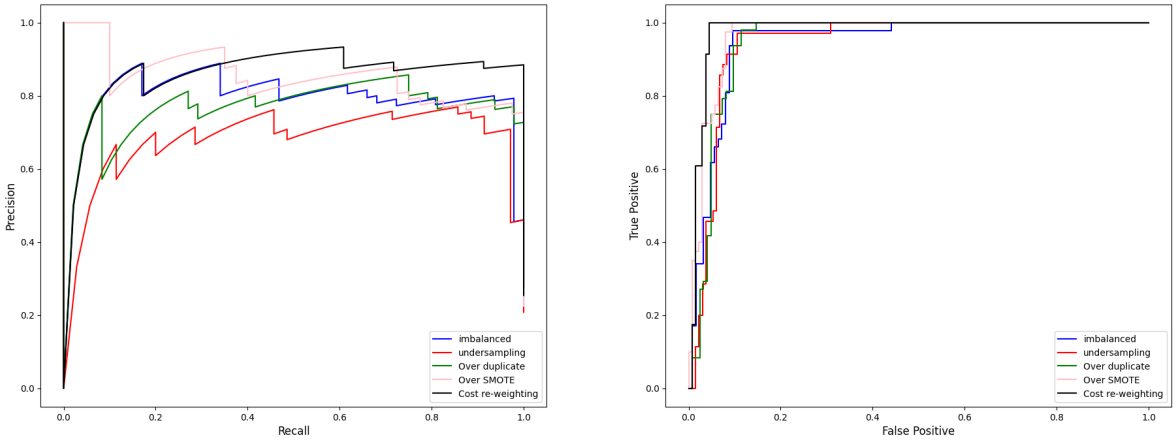
De igual manera se puede concluir que el modelo generalizo bien.

1.7.2. Clases des balanceadas

Al estudiar distintos métodos de re balanceo de clases, se consiguieron las siguientes métricas

Modelo	Accuracy	Precision	Recall	F-Score	AUC-ROC	AUC-PR
Sin rebalanceo	0.919	0.800	0.936	0.863	0.946	0.788
Undersampling	0.905	0.702	0.943	0.805	0.943	0.676
Oversampling duplicate	0.912	0.770	0.979	0.862	0.948	0.764
Oversampling SMOTE	0.927	0.755	1.000	0.860	0.968	0.854
Cost re-weighting	0.956	0.880	0.957	0.917	0.978	0.857

Figura 3: Metricas obtenidas en el dataset de clases desbalanceadas



(a) Las curvas PR de cada metodo

(b) Las curvas ROC de cada metodo

Figura 4

Como se puede ver, el método de Undersampling es menos certero en este caso, porque es tan grande la diferencia de muestras entre los diagnosticos positivos y los negativos, que el modelo se queda con muy pocas muestras para entrenar. Luego el el método de Oversampling duplicate tiene un mejor rendimiento, ya que es lo opuesto de Undersampling, en este caso el modelo si tiene suficientes muestras para entrenar y por eso tiene una mayor certeza. Se puede ver también que ovesrampling SMOTE es mejor que oversampling duplicate, y esto se debe a que al duplicar la información no se esta ganando nada mientras que con SMOTE al utilizar KNN se obtiene muestras nuevas parecidas pero no exactamente iguales, por lo que el modelo no esta aprendiendo de la misma información repetida y tiene menos probabilidades de overfitear. El mejor metodo se ve que es Cost re-weighting y esto era de esperar ya que es el unico modelo que no duplica información, remueve información o crea nueva a partir de KNN.

En conclusion, al momento de utilizar algunos de los metodos propuestos, en produccion, convendria utilizar Cost re-weighting ya que obtuvo las puntuaciones mas altas en la mayoria de las metricas medidas.

2. Predicción de Rendimiento de Jugadores de Basketball

Resumen

En este experimento se modelo la capacidad de predicción de la regresión logística con regularización L2, LDA y Random Forest para un problema de clasificación multi-clase. Se probaron distintos hiperparametros y se comparo a los 3 modelos en diversas métricas y se observo que el modelo que mejor performo fue el Random Forest.

2.1. Introducción

En este problema se busco predecir el rendimiento de jugadores de Basketball. La entrada de este problema de multi-clases son las estadísticas de los jugadores durante la temporada. Se estudio la capacidad de tres modelos distintos, regresión logística con regularización L2,

Análisis discriminante lineal (LDA) y Random Forest para obtener una predicción de 3 distintas clases en base al rendimiento del jugador, negative WAR (1), null WAR (2), positive WAR (3).

2.2. Métodos

2.2.1. Regresión Logística

Para problemas multiclase se emplea regresión logística *softmax*. En este caso, se calcula la probabilidad de cada clase k mediante la función *softmax*:

$$P(y = k|\mathbf{x}) = \frac{e^{\mathbf{w}_k^T \mathbf{x}+b_k}}{\sum_{j=1}^{K} e^{\mathbf{w}_j^T \mathbf{x}+b_j}}$$

donde K es el número de clases (en este caso, $K = 3$), $\mathbf{w}_k$ y b_k son los pesos y sesgos asociados a la clase k. La clase predicha es aquella con la mayor probabilidad.

En este trabajo, se implementó una regresión logística con regularización L2 para evitar el sobreajuste.

2.2.2. Análisis Discriminante Lineal (LDA) para Multiclase

Para un problema multiclase, LDA calcula las funciones discriminantes lineales para cada clase k:

$$\delta_k(\mathbf{x}) = \mathbf{x}^T \Sigma^{-1} \boldsymbol{\mu}_k - \frac{1}{2} \boldsymbol{\mu}_k^T \Sigma^{-1} \boldsymbol{\mu}_k + \log(\pi_k)$$

donde $\boldsymbol{\mu}_k$ es el vector de medias de la clase k, Σ es la matriz de covarianza común, y π_k es la probabilidad a priori de la clase k. La clase predicha es aquella que maximiza $\delta_k(\mathbf{x})$.

2.2.3. Bosque Aleatorio para Multiclase

El Bosque Aleatorio (*Random Forest*) es un método de aprendizaje por ensamblado que combina múltiples árboles de decisión para mejorar la precisión y reducir el sobreajuste. Cada árbol se entrena con una muestra bootstrap del conjunto de datos original, y en cada nodo del árbol se selecciona un subconjunto aleatorio de características para determinar la mejor división.

Para problemas multiclase, cada árbol predice la clase de una muestra, y la predicción final del bosque se obtiene mediante votación mayoritaria (para clasificación) o promediando las probabilidades (para predicciones probabilísticas).

2.2.4. Conjunto de datos e hiperparametros

Para este experimento se utilizaron los datasets, 'WAR class dev' y 'WAR class test'. Para entrenamiento se utilizo el 'dev' y se lo separo en entrenamiento y validación, con una proporción de 80 % y 20 % respectivamente. Ademas se normalizo por medio de la media y la desviación estándar. Tambien se removió los outlayers por medio de boxplots con Q1 = 0.25 y Q3 = 0.75. El hiperparametro utilizado para regularizaion L2 fue de 0.01, y para la regresión se utilizo un learing rate de 0.01 y 1000 iteraciones max. Para el Random Forest se utilizaron 3

arboles de profunidad 2, se eligieron estos hiperparametros ya que al trabajar con un dataset de dimensionalidad 3, mayor cantidad de arboles y mayor profundidad sobreajustaria, y cuando se probaron valores mas chicos como 2 arboles de profundidad 1 el modelo tenia muy malos resultados. Ademas, en base al analisas de la correlacion de las caracteristicas se decidio eliminar la columna 'mp' ya que tenia una correalacion de 1 con la columna 'poss' y tambien se elemino 'raptor total' porque tenia muy alta correlacion con 'war total'.

2.3. Resultados

Al comparar los distintos modelos, se obtuvieron las siguientes metricas en el set de validacion:

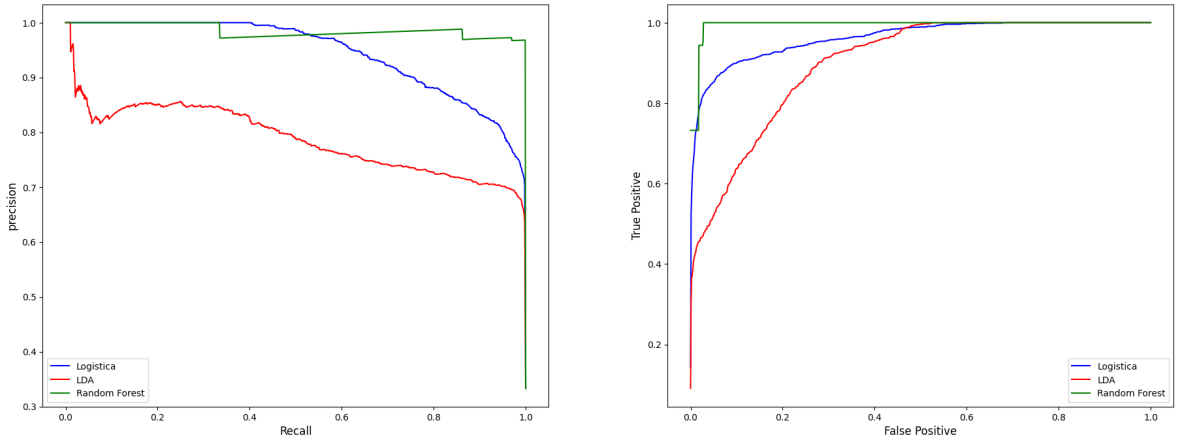
Modelo	Accuracy	Precision	Recall	F-Score	AUC-ROC	AUC-PR
Logistic	0.875	0.872	0.875	0.869	0.955	0.937
LDA	0.725	0.717	0.725	0.702	0.882	0.763
Random Forest	1.000	1.000	1.000	1.000	1.000	0.998

Figura 5: Métricas obtenidas en el set de validacion.

Como se puede ver el modelo LDA es el que tiene peor rendimiento, ya que LDA funciona mejor cuando las clases son linealmente separables y en este caso hay superposición entre ellas. La regresion tambien es un modelo lineal, pero es capaz de rendir bien incluso cuando las clases no son completamente linealmente separables, por eso tiene métricas mas altas que LDA. Por otro lado el Random Forest tiene un excelente rendimiento, ya que al ser un problema de baja dimensionalidad, es posible para el arbol separar a todas las clases y tener una certeza total.

Modelo	Accuracy	Precision	Recall	F-Score	AUC-ROC	AUC-PR
Logistic	0.889	0.886	0.889	0.884	0.963	0.944
LDA	0.742	0.736	0.742	0.725	0.901	0.787
Random Forest	0.970	0.973	0.970	0.971	0.995	0.985

Figura 6: Métricas obtenidas en el set de test.



(a) Las curvas PR de cada modelo

(b) Las curvas ROC de cada modelo

Figura 7

Cuando analizamos el set de test encontramos el mismo patrón, solo que ahora regresión y LDA tiene métricas mas altas, mientras que el Random Forest rinde un poco peor. Esto tiene sentido ya que ambos sets cumplen el mismo objetivo, comprobar el rendimiento del modelo, y si los datos fueron procesados correctamente y no hubo filtración de información del val o test, al train, entonces uno puede esperar resultados similares en ambos sets.

Es claro que para un entorno de producción de problemas de baja dimensionalidad, el Random Forest es el mejor modelo de predicción.