Mentoría Data Science aplicado a BCI

Consignas TP4: Introducción al Aprendizaje Automático

Aprendizaje automático

A) Entendimiento del problema:

a) El problema al cual nos enfrentamos es un problema de **clasificación supervisada multiclase** ya que se trata de una tarea de clasificación mutuamente excluyente entre más de dos clases. La clasificación multiclase parte del supuesto de que cada muestra está asignada a una sola etiqueta, y debido a que cada ejemplo se le asigno una única etiqueta, por mas que se tratara de un ejemplo impuro, se trata de un problema multiclase, donde, en nuestro problema en particular, las clases involucradas son las etiquetas 1, 2 y 99. Aunque bien se podría encarar el análisis como un problema de clasificación supervisada binaria, al considerar solo las dos clases que son relevantes para nuestro problema, es decir, las dos clases que corresponden a la recepción de un estímulo, la clase 1 y la clase 2, sin considerar la clase 99, la cual corresponde a momentos en los que el individuo no recibe un estímulo.

b) El análisis de regresión consiste en un conjunto de métodos de aprendizaje automático que nos permiten predecir una variable de resultado continua (y) en función del valor de una o varias variables predictoras (x). Brevemente, el objetivo del modelo de regresión es construir una ecuación matemática que defina (y) como una función de las variables (x). A continuación, esta ecuación se puede utilizar para predecir el resultado (y) sobre la base de nuevos valores de las variables predictoras (x). En nuestro problema en particular se podría plantear un problema de regresión utilizando como variable predictora (x) la señal cruda, es decir la señal de voltaje, y como variable a predecir (y) la frecuencia, producto de la transformada de Fourier de la señal cruda.

c) El aprendizaje no supervisado tiene lugar cuando no se dispone de salidas para el entrenamiento. Solo conocemos los datos de entrada, pero no existen datos de salida que correspondan a una determinada entrada. Por tanto, solo podemos describir la estructura de los datos, para intentar encontrar algún tipo de organización que simplifique el análisis. Por ello, para este problema se podría plantear un problema de aprendizaje no supervisado otorgándole como elementos de entrada los features, como podrían ser, las frecuencias con mayor potencia, o las amplitudes de las frecuencias de interés, resultados de la aplicación de la transformada de Fourier a la ventana de tiempo. De esta forma se podrá hacer un agrupamiento que implícitamente se deberían corresponder con las etiquetas correspondientes. Aunque como en nuestro problema en particular, como se cuentan con datos etiquetados, y no se busca generar una agrupación de los datos, no es de mucha utilidad aplicar aprendizaje no supervisado, ya que no sería útil hacer una división en clusters, aunque si nos sería de utilidad si por ejemplo se adquirieran nuevos datos sin etiquetar, allí podríamos agrupar dichos datos a partir de los ya etiquetados.

B) En el problema de clasificación supervisada:

a) Las métricas que podemos utilizar para evaluar el método de clasificación son:

Métricas de evaluación para la clasificación:

- **Accuracy (Exactitud):** Es el porcentaje total de elementos clasificados correctamente. Es la medida más directa de la calidad de los clasificadores. Es un valor entre 0 y 1. Cuanto más alto, mejor. Sin embargo, la métrica accuracy (exactitud) no funciona bien cuando las clases están desbalanceadas por lo que solo es una buena medida cuando las clases de variables de destino en los datos están casi equilibradas.
- **Recall, Sensibilidad o TPR (Tasa de True Positive):** Es el número de elementos identificados correctamente como positivos del total de positivos verdaderos.
- **Precisión:** Es el número de elementos identificados correctamente como positivo de un total de elementos identificados como positivos.
- **Especificidad o TNR (Tasa Negativa Verdadera**): Es el número de ítems correctamente identificados como negativos fuera del total de negativos.
- **Puntuación F1:** Esta métrica es la combinación de las métricas de Precisión y Recall''. La mejor puntuación F1 es igual a 1 y la peor a 0.

Para poder obtener una visión general de los resultados como parámetro de evaluación se escoge la métrica accuracy. Pero como en nuestro problema en particular los datos no se encuentran balanceados se adiciona la métrica Score F1 en la evaluación.

b) Las funciones de pérdida definen un objetivo con el que se evalúa el rendimiento del modelo y los parámetros aprendidos por el modelo se determinan minimizando una función de pérdida elegida. Las funciones de pérdida definen qué es y qué no es una buena predicción. La funcion de pérdida que se utilizara, dependerá del problema de clasificación del cual se esté tratando, y del modelo predictivo que se haya escogido. Para nuestro problema de clasificación en particular, como se trata de un problema multiclase, se podría utilizar la pérdida de entropía cruzada categórica, la cual es esencialmente la pérdida de entropía cruzada binaria expandida a múltiples clases. La pérdida de entropía cruzada aumenta a medida que la probabilidad predicha difiere de la etiqueta real. La fórmula de la pérdida de entropía cruzada categórica es la siguiente:

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

Aprendizaje Supervisado

A) Benchmarking y desarrollo del algoritmo desarrollador:

a) Para realizar el benchmarking se asignaron etiquetas de forma aleatoria a los features y se evaluó el modelo con las métricas escogida (accuracy y f1score).

b) Para obtener las métricas de benchmark se utilizó un promedio de los resultados de las métricas f1score, y accuracy, luego de realizar la asignación de etiquetas de forma aleatoria cinco veces, siendo los siguientes los resultados obtenidos.

ACCURACY= [0.33, 0.34, 0.33, 0.33, 0.32].

F1SCORE= [0.36, 0.37, 0.35, 0.35, 0.35].

PROMEDIO ACCURACY= [0.34]

PROMEDIO F1SCORE= [0.37]

B) Búsqueda a grandes rasgos:

a) Se evaluaron diferentes modelos sobre el conjunto de datasets obtenidos, que involucra los siguientes:

- dataset_time: conformado por el valor máximo, el valor mínimo, la media, y la moda de cada ventana de tiempo, correspondiente a cada canal.
- dataset_frec_max: conformado por la frecuencia con amplitud máxima de cada ventana de tiempo, correspondiente a cada canal.
- dataset_sxx: conformado por el logaritmo natural de la media, el máximo, y el cuantil 95, de valores de amplitud de cada ventana de tiempo, correspondiente a cada canal.
- dataset_freq_select: conformado por el promedio de amplitudes de las frecuencias cercanas a las frecuencias de interés (sean estas, 12.5, 16.5, 25.0, 33.0) de cada ventana de tiempo, correspondientes a cada canal.
- dataset_freq_unif: conformado por el promedio de las frecuencias selectivas por canal, del dataset '' dataset_freq_select''.

Los modelos que se evaluaron incluyen:

- Naive Bayes.
- SuperVectorMachine.
- SuperVectorMachin utilizando Kernels.
- RandomForest.
- Neural Network.
- K-Nearest Neighbor.
- XGBoost.
- DecisionTreeClassifier.

Los resultados del análisis inicial fueron los siguientes:

Datasets / Modelos	Metrica	dataset_time	dataset_frec_max	dataset_sxx	dataset_freq_select	dataset_freq_unif
Naive Bayes	F1Score	-	0.47	-	0.63	0.64
	Accuracy	-	0.61	-	0.62	0.62
SuperVectorMachine	F1Score	0.53	0.57	0.44	0.24	0.20
	Accuracy	0.59	0.55	0.42	0.35	0.34
SuperVectorMachine utilizando Kernels	F1Score	0.53	0.54	0.47	0.48	0.45
	Accuracy	0.58	0.59	0.45	0.48	0.45
RandomForest	F1Score	0.92	0.70	0.79	0.72	0.64
	Accuracy	0.92	0.71	0.79	0.74	0.66
Neural Network	F1Score	0.56	0.49	0.49	0.67	0.62
	Accuracy	0.63	0.61	0.61	0.69	0.66

K-Nearest Neighbor	F1Score	0.77	0.63	0.61	0.77	0.65
	Accuracy	0.77	0.64	0.60	0.79	0.66
Gradient boosting	F1Score	0.72	0.70	0.78	0.72	0.68
	Accuracy	0.74	0.73	0.78	0.74	0.71
DecisionTreeClassifier	F1Score	0.86	0.65	0.70	0.66	0.60
	Accuracy	0.86	0.65	0.69	0.67	0.61

Debido a que las clases se encontraban desbalanceadas, con aquellos modelos que lo permitían, se balancearon las clases para obtener así mejores resultados.

Si bien casi todos los modelos superan el desempeño del modelo de benchmark, excepto algunas excepciones, ciertos modelos poseen un mejor desempeño que otros modelos, como ser los modelos RandomForest, árboles de decisión, XGBoost y K-Nearest Neighbor. Naive Bayes, SuperVectorMachine, SuperVectorMachin utilizando Kernels y las redes neuronales de SKLearn no tuvieron un muy buen desempeño. A continuación, se analiza cada caso en particular:

- Naive Bayes: No se obtuvo un muy buen desempeño en general con este modelo, y con ciertos features, no clasificaba correctamente las clases 1 y 2, y en algunos casos directamente no las clasifica, por lo que dicho modelo se descartó.
- SuperVectorMachine: El desempeño del modelo no fue lo suficientemente bueno en ninguno de los casos, hasta en ciertos features no se superó el desempeño del benchmark. Al modificar los kernels utilizados mejora en cierta forma el desempeño del modelo, pero no es suficiente como para ser tenido en cuenta, ya que hay modelos con mejores desempeños. A diferencia de lo que ocurre en otros casos, en el caso de svm, con ciertos features, como ser las amplitudes de frecuencias selectivas, se pudo ver que si bien el score es bajo, clasificaba mejor las clases de interés, 1 y 2, y no así la clase 99, pero de todas formar los scores son relativamente bajos y no se considera útil la evaluación del modelo.
- RandomForest: Es el modelo que mejor desempeño presenta, por lo que, si bien no clasifica con la misma eficacia todas las etiquetas (siendo la clase 2 la peor clasificada), es el escogido para seguir con la evaluación.
- Neural Network: la red neuronal evaluada presenta un mal desempeño en general, y variable entre los features evaluados, por lo que no fue tenido en cuenta.
- K-Nearest Neighbor: Si bien el modelo no tuvo el mejor resultado de todos los modelos, se continuó con su análisis, ya que tiene un fundamento diferente al RandomForest, y se considera que optimizando hiperparámetros se pueden obtener buenos resultados.
- Gradient boosting: Si bien es uno de los modelos con mejor desempeño, se escoge utilizar RandomForest ya que tienen un fundamento similar por detrás, y este último presento un mejor desempeño.
- DecisionTreeClassifier: Este modelo también presentó un buen desempeño, pero como ocurre con el Gradient boosting, se escoge utilizar RandomForest ya que tienen un fundamento similar por detrás, siendo RandomForest una opción mejorada de los árboles de decisión.

En el caso de aquellos modelos en los que su fundamento se basa en árboles de decisión, es decir, RandomForest, XGBoost, y árboles de decisión propiamente dicho, se pudo observar que existía un overfitting al entrenar los modelos, esto se evidencia observando que existe un valor máximo para las métricas en la evaluación del conjunto de entrenamiento, y esta disminuía considerablemente en la evaluación con el conjunto de validación. Esto ocurre porque los árboles de decisión por definición overfitean el conjunto de entrenamiento.

Finalmente se conservaron los modelos de RandomForest y K-Nearest Neighbor para continuar el análisis de forma más exhaustiva y optimizar los modelos con la búsqueda de hiperparámetros adecuados, ya que, en general, fueron los que mejores métricas presentaron.

Con respecto a los features evaluados, fueron variables los resultados obtenidos, ya que para ciertos modelos se desempeñaban mejor ciertos features que otros, pero de acuerdo a los desempeños obtenidos en los modelos escogidos se continúa con el análisis del dataset_time para el modelo de RandomForest (a pesar de que para los features de la señal cruda no se encontró correlación previamente) y el dataset_freq_select para el modelo K-Nearest Neighbor. En ambos casos se balancean los datos para obtener resultados mas confiables.

Esta búsqueda otorgó como resultados finales, evaluados sobre el conjunto de test, las siguientes métricas:

-MODELO RANDOM FOREST:

F1SCORE= [0.89]

ACCURACY= [0.89]

-MODELO KNN:

F1SCORE= [0.81]

ACCURACY= [0.81]

Si bien estos resultados fueron aceptables, se considera que pueden ser mejorados, ya que, aunque el valor global de las métricas es considerablemente alto, las clases de interés no son tan bien clasificadas como lo es así la clase 99, por lo que se evaluó un modelo de XGBoosting el cual solo se consideraron las clases de interés (1 y 2), el cual otorgó los siguientes resultados:

- MODELO XGB:

F1SCORE= [0.80]

ACCURACY= [0.80]

Esto nos demuestra que no se obtuvieron mejores resultados que los ya obtenidos previamente, por lo que, el modelo que se escoge es RandomForest usando como features las transformaciones aplicadas a las ventanas de tiempo de la señal cruda, a pesar de que se contradice con lo observado en los Trabajos Prácticos anteriores (en los cuales no se encontró correlación previa), ya que es el modelo que mejor desempeño presenta, y aquel que clasifica mejor las clases.

El diseño del modelo es el siguiente:

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight='balanced',
                       criterion='entropy', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

El desempeño del modelo es el siguiente:

```
MODELO: RANDOM FOREST

F1Score= 0.8923456035411121
Accuracy= 0.8924455825864277

              precision    recall  f1-score   support

           1       0.81      0.83      0.82       140
           2       0.88      0.84      0.86       173
          99       0.92      0.93      0.93       468

    accuracy                           0.89       781
   macro avg       0.87      0.87      0.87       781
weighted avg       0.89      0.89      0.89       781
```

En un análisis futuro se podría evaluar realizar un análisis personalizado por paciente, ya que como se observó en el análisis previo, existía una alta variabilidad entres los resultados observados entre pacientes.