Resultados Analisis taller 2

Oscar Julian Rodriguez Cardenas
Introducción Machine Learning para economistas

June 4, 2022

1 Algoritmos escogidos

El tipo de problema elegido a resolver fué de clasificación, se tenía que clasificar a partir de variables como la edad, el genero, los niveles de presión en la sangre, los niveles de colesterol y el porcentaje de sodio y potasio que tipo de droga es la más adecuada para darle a una persona, escogí el algortimo de **Random Forest** y el algoritmo de **Gaussian Naive Bayes**.

2 Conjuntos de hiperparámetros

Para el algoritmo de Random Forest los hiperparámetros tenidos en cuenta fueron: El número de árboles o estimadores en el bosque donde habían mínimo 1 y máximo 20, maxfeatures que es el número de características a tener en cuenta para hacer la mejor partición, maxdepth que se refiere a la profundidad máxima que puede alcanzar cada árbol del bosque, minsamples que se refiere al número mínimo de muestras requeridas para estar en un nodo de hoja.

Para el algoritmo de Gaussian Naive Bayes el único hiperparámetro tuneable era varsmoothing que se refiere a agregar varianza artificialmente a los datos.

3 Mejores hiperparámetros de cada modelo.

1. **RANDOM FOREST**

	n_estimators	max_features	max_depth	min_samples_split	f1 score
0	1	auto	1	2	0.247085
1	1	auto	1	5	0.148458
2	1	auto	2	2	0.407957
3	1	auto	2	5	0.401925
4	1	auto	3	2	0.507979
...	...	...	...	...	...
315	20	sqrt	7	5	0.960616
316	20	sqrt	8	2	0.947783
317	20	sqrt	8	5	0.970347
318	20	sqrt	10	2	0.947783
319	20	sqrt	10	5	0.947783

320 rows × 5 columns

Podemos ver que para encontrar los mejores hiperparámetros se realizaron 320 posibles combinaciones de hiperparámetros (ver el código para ver la tabla completa), los parametros del mejor modelo fueron entonces: Número de arboles en el bosque (nestimators) fueron 7, número de características a tener en cuenta para hacer la mejor partición(maxfeatures) fué sqrt, la profundidad máxima de cada árbol(maxdepth) fué de 10 y finalmente el número mínimo de muestras requeridas para estar en un nodo de hoja (minsamples) fué de 5.Con el modelo inicial se obtenía una métrica f1 de 0.77 y después de optimizar los hiperparámetros se mejoró a 0.97

2. **GAUSSIAN NAIVE BAYES**

	var_smoothing	f1 score
0	1.000000e+00	0.620669
1	8.111308e-01	0.694779
2	6.579332e-01	0.710063
3	5.336699e-01	0.702380
4	4.328761e-01	0.734564
...	...	...
95	2.310130e-09	0.815624
96	1.873817e-09	0.815624
97	1.519911e-09	0.815624
98	1.232847e-09	0.815624
99	1.000000e-09	0.815624

100 rows × 2 columns

Podemos ver que para encontrar el mejor hiperparámetro se realizaron 100 iteraciones (ver el código para ver la tabla completa), el parametro del mejor modelo fué entonces: la varianza artificial de los datos (varsmoothing) fué de 0.12328467394420659.5.Con el modelo inicial se obtenía una métrica f1 de 0.8 y después de optimizar los hiperparámetros se mejoró a 0.95.

Aunque con el modelo de Random Forest se obtuvó una mejor métrica f1, prefiero quedarme con el modelo de Gaussian Naive Bayes pues si se observa cuando se estan entrenando los modelos con distintos hiperparámetros mediante validación cruzada, el modelo de Gaussian Naive Bayes tiene menos varianza en sus resultados, lo que se podría traducir en menos sobreajuste y una mejor generalización de los datos

4 Comparación de algoritmos

El clasificador Gaussian naive Bayes tiene como característica principal que asume que el efecto de una característica particular en una clase es independiente de otras características, cuando las variables realmente son independentes tiene un mejor desempeño que otros modelos de clasificación como en este caso que obtuvo un mejor desempeño que Random Forest tal vez debido a esto, además este algoritmo funciona bastante bien con entradas categóricas, sin embargo hay una desventaja de este algoritmo si la variable categórica tiene una categoría en el conjunto de datos de prueba, que no se observó en el conjunto de datos de entrenamiento, el modelo asignará una probabilidad de 0 y no podrá hacer una predicción.El algoritmo de Random Forest ya sabemos que son varios árboles de decisión, una de las ventajas de este modelo es que nos puede mostrar la importancia de las variables para predecir otra variable, al igual Gaussian Naive Bayes funciona bastante bien con datos categóricos, sin embargo una desventaja de este algoritmo es la pérdida de interpretación de los resultados es una especie de caja negra como las redes neuronales artificiales, una ventaja de este algoritmo sobre Gaussian Naive Bayes es que al no basarse plenamente sobre probabilidades le permite hacer predicciones sobre valores no vistos durante su entrenamiento.La principal diferencia entre estos dos algoritmos es que Gaussian Naive Bayes se basa totalmente en probabilidades condicionales para predecir mientras que Random Forest busca que variables son las más importantes para predecir y busca un umbral para cada variable.