Presentación Ensembles Práctica 4 Entrega y comentarios

Docente: José Manuel Aroca

UNIVERSIDAD DE BURGOS

Minería de Datos
Curso 2022/2023

Qué vamos a utilizar

Entrega:

1. Solo entregar projecto de Knime
2. Explicaciones sobre la forma de tratamiento de datos iniciales para el muestreo.
3. Algoritmos evaluados y combinados dentro del ensemble.(introducir algunos que puedan empeorar resultados y sacar conclusiones)
4. Resultados a modo de tabla con los ganadores/perdedores.
5. ¿Cuál es la mejor combinación y por qué?
6. ¿Cuál es la peor combinación y por qué?

Para el correcto funcionamiento de un ensemble, cada uno de los clasificadores que lo conforman se debe entrenar con un subconjunto de datos diferente, pues de otra forma, se obtendrían k modelos exactamente iguales. Para obtener esos k subconjuntos de datos, se debe tener en cuenta lo siguiente:

• Para el conjunto “labeled”: se obtendrá un muestreo del mismo tamaño (si contiene 10 instancias, el subconjunto contendrá 10 instancias). Sin embargo, el muestreo será con reemplazamiento, es decir, algunas de las instancias pueden estar repetidas.

▷ Muestreo con reemplazo versus sin reemplazo en 2023 → STATOLOGOS®

• Para el conjunto “unlabeled”: se obtendrá un submuestreo del tamaño definido por el parámetro unlabeled_sample_frac (valor entre 0 y 1). Por ejemplo, si el parámetro toma el valor 0.75 (valor que se debe fijar por defecto), y el conjunto contiene 100 instancias, la submuestra contendrá 75. En este caso, el submuestreo debe realizarse sin reemplazamiento.

https://deepnote.com/@a_mas/Funciones-de-Muestreo-en-Python-5e3efaeb-d12c-472c-96b7-bb34e377b3e7

Muestreo en Python - Analytics Lane

Una vez se han generado ambos subconjuntos, se deben combinar para obtener el conjunto de datos de entrenamiento de su clasificador correspondiente (instancia de SelfTrainingClassifier). Para este clasificador, se deben asignar los parámetros threshold y max_iter, que se habían definido previamente como parámetros del ensemble (st_threshold, st_max_iter).

```
X = df[['User1', 'User2']]
y_train = df['Class'].values

enc = OneHotEncoder(sparse=False)
enc.fit(X[['User1']])

# using the same encoder assuming that the A1 whether as User1 or User2 are the same
X_train = np.concatenate((enc.transform(X[['User1']]), enc.transform(X[['User2']])),

# without weights
lp_model = LabelSpreading(gamma=0.25, max_iter=20)
lp_model.fit(X_train, y_train)
predicted_labels = lp_model.transduction_[unlabeled_set]
print(predicted_labels)
```

Debido a que contamos con k modelos entrenados, ¿Cuál de las k predicciones escogemos como predicción final? La idea será combinar todas las predicciones con la estrategia conocida como Majority Voting. Una vez se obtienen las k predicciones, la predicción final será la más común (si en un ensemble de tamaño 10, 6 predicen que el agua no es potable, y 4 que sí, la predicción final del ensemble será que el agua no es potable).

sklearn.ensemble.VotingClassifier — scikit-learn 1.2.2 documentation

Ensemble methods: majority voting example | Kaggle

[1.11. Ensemble methods — scikit-learn 1.2.2 documentation]

https://medium.com/analytics-vidhya/ensemble-models-bagging-boosting-c33706db0b0b

https://forum.knime.com/t/custom-ensemble-machine-learning-algorithm-for-classification-problem/31566/3