

Bitácora de experimentos

Dataset

- heart.csv

Experimento 1

Fecha y hora: 15 / julio / 2021

Especificaciones

- **Especificación 1:** Sin hiper parámetros.
- **Especificación 2:** test_size = 0.25 → De 1,025 registros, 768 fueron para entrenamiento y 257 para la evaluación.

Algoritmos

Algoritmo: Clústeres por vecinos más cercanos

Configuración:

- `KNN = KNeighborsClassifier()`
- `KNN.fit(X_train, y_train)`
- `y_pred_knn = KNN.predict(X_test)`

Resultado:

- Precisión del modelo en el subconjunto de entrenamiento: 0.99
- Precisión del modelo en el subconjunto de prueba: 0.96

Algoritmo: Clasificador ingenuo de Bayes

Configuración:

- NaiveBayes = GaussianNB()
- NaiveBayes.fit(X_train, y_train)
- y_pred_nb = NaiveBayes.predict(X_test)

Resultado:

- Precisión del modelo en el subconjunto de entrenamiento: 1.0
- Precisión del modelo en el subconjunto de prueba: 1.0

Algoritmo: Regresión

Configuración:

- LogisticRegression = LogisticRegression()
- LogisticRegression.fit(X_train, y_train)
- y_pred_lr = LogisticRegression.predict(X_test)

Resultado:

- Precisión del modelo en el subconjunto de entrenamiento: 1.0
- Precisión del modelo en el subconjunto de prueba: 1.0

Algoritmo: Máquina de Soporte Vectorial

Configuración:

- SVC = SVC()
- SVC.fit(X_train, y_train)
- y_pred_svc = SVC.predict(X_test)

Resultado:

- Precisión del modelo en el subconjunto de entrenamiento: 1.0
- Precisión del modelo en el subconjunto de prueba: 1.0

Algoritmo: Bosque Aleatorio

Configuración:

- RFC = RandomForestClassifier()
- RFC.fit(X_train, y_train)
- y_pred_rfc = RFC.predict(X_test)

Resultado:

- Precisión del modelo en el subconjunto de entrenamiento: 1.0
- Precisión del modelo en el subconjunto de prueba: 1.0

Algoritmo: Red Neuronal

Configuración:

- ANN = MLPClassifier()
- ANN.fit(X_train, y_train)
- y_pred_ann = ANN.predict(X_test)

Resultado:

- Precisión del modelo en el subconjunto de entrenamiento: 1.0
- Precisión del modelo en el subconjunto de prueba: 1.0

Conclusión

Los mejores modelos de clasificación son el clasificador ingenuo de Bayes, la regresión, la máquina de soporte vectorial, el bosque aleatorio y la red neuronal; sin embargo, me temo que he caído en un Overfitting.

Experimento 2

Fecha y hora: 16 / julio / 2021

Especificaciones

- **Especificación 1:** Con hiper parámetros.
- **Especificación 2:** `test_size = 0.4` → De 1,025 registros, 615 fueron para entrenamiento y 410 para la evaluación.

Algoritmos

Algoritmo: Clústeres por vecinos más cercanos

Configuración:

- (`weights = 'distance'`, `n_neighbors = 8`, `metric = 'euclidean'`)

Resultado:

- Precisión del modelo en el subconjunto de entrenamiento: 1.0
- Precisión del modelo en el subconjunto de prueba: 0.98

Algoritmo: Clasificador ingenuo de Bayes

Configuración:

- NaiveBayes = GaussianNB()
- NaiveBayes.fit(X_train, y_train)
- y_pred_nb = NaiveBayes.predict(X_test)

Resultado:

- Accuracy 100%
- Sensibilidad 100%
- Especificidad 100%

Algoritmo: Regresión

Configuración:

- (penalty = 'none', C = 31622776.60168379)

Resultado:

- Precisión del modelo en el subconjunto de entrenamiento: 1.0
- Precisión del modelo en el subconjunto de prueba: 1.0

Algoritmo: Máquina de Soporte Vectorial

Configuración:

- (gamma = 0.1, C = 43939.70560760786)

Resultado:

- Accuracy 100%
- Sensibilidad 100%
- Especificidad 100%

Algoritmo: Bosque Aleatorio

Configuración:

- (n_estimators = 325, max_features = 'log2', max_depth = 10, criterion = 'entropy')

Resultado:

- Accuracy 100%
- Sensibilidad 100%
- Especificidad 100%

Algoritmo: Red Neuronal

Configuración:

- (hidden_layer_sizes = (10, 30, 10), alpha = 0.1, activation = 'relu')

Resultado:

- Accuracy 100%
- Sensibilidad 100%
- Especificidad 100%

Conclusión

Creo igualmente que he vuelto a caer en el Overfitting, pero ya se implementaron hiper parámetros... tal vez la BBDD es muy pequeña. Los mejores modelos de clasificación son el clasificador ingenuo de Bayes, la regresión, la máquina de soporte vectorial, el bosque aleatorio y la red neuronal.