Benemérita Universidad Autónoma de Puebla - FCC

Minería de Datos

Avance II. Implementación y experimentación con clasificadores

SEMANA DEL 20 - 24 DE ABRIL 2020

Emiliano Carrillo Moncayo

Introducción y Objetivos

Esta semana se planteó como objetivo escoger y experimentar con los 3 modelos de clasificadores que ocuparé para mi ensamble final. Para esto, se ocuparon las técnicas de preprocesamiento implementadas en la sección pasada. Objetivamente, la extracción de variables y procesamiento del archivo de texto, la estandarización de datos por medio del método StandardScaler, y la aplicación del PCA.

Se tomó la decisión de ocupar PCA como método para cada modelo de clasificador ya que, tras experimentar con esto, se observó que optimizaba todos los casos. De igual forma empecé a experimentar con el nuevo dataset con 385 atributos. Para este dataset en específico observé que la cantidad de componentes óptima para el PCA eran 12 así que la experimentación a continuación muestra los resultados de los clasificadores tras reducir la dimensionalidad del dataset a 12 componentes principales.

Experimentos y Desarrollo

Decidí ocupar los siguientes métodos de clasificación supervisados para hacer mi experimentación: K-Nearest-Neighbors, Naive Bayes, y una Red Neuronal.

Para todos los experimentos expuestos a continuación se dividió el dataset en sets de entrenamiento y pruebas, y se ocuparon los mismos sets para entrenar y probar cada uno de éstos.

K-Nearest-Neighbors

Para el modelo KNN experimenté con el parámetro de cantidad de vecinos (valor K) para ver cuál valor era el más apropiado para este dataset en específico. Para ello construí un loop que me calculó el error promedio tras hacer la evaluación para cada valor k de 1 a 40. La gráfica es la siguiente:

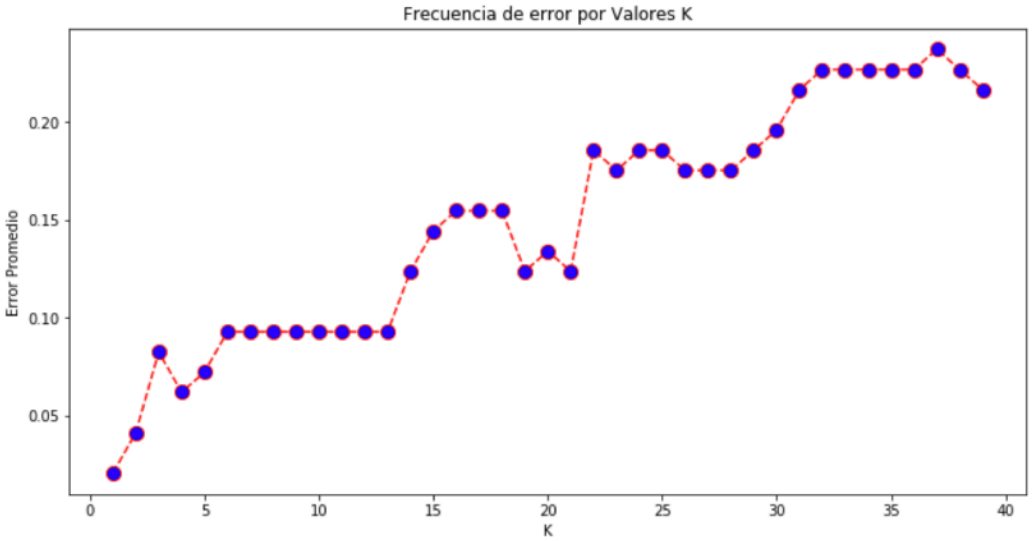

FIG 1. FRECUENCIA DE ERROR PARA DISTINTOS VALORES K

Tras varios experimentos observé que con 5 vecinos cercanos obtenía resultados de predicción más elevados. Al final obtuve un puntaje de predicción del 93% con KNN.

```
******* CLASIFICADOR KNN *******

Matriz de confusión:
[[11  1  0  0  0  0]
 [ 0 40  0  0  0  1]
 [ 0  1  9  0  2  0]
 [ 0  0  0  4  0  0]
 [ 0  0  0  0  6  2]
 [ 0  0  0  0  0 20]]

Reporte de clasificación:
              precision    recall  f1-score   support

           0       1.00      0.92      0.96        12
           1       0.95      0.98      0.96        41
           2       1.00      0.75      0.86        12
           3       1.00      1.00      1.00         4
           4       0.75      0.75      0.75         8
           5       0.87      1.00      0.93        20

    accuracy                           0.93        97
   macro avg       0.93      0.90      0.91        97
weighted avg       0.93      0.93      0.93        97


Puntaje de precisión:
0.9278350515463918
```

FIG 2. RESULTADOS DEL CLASIFICADOR KNN

Naive Bayes

Para el clasificador de Naive Bayes se ocupó el modelo gausiano por la naturaleza y forma del dataset. Originalmente estaba ocupando un algoritmo Multinominal y conseguía resultados extraños.

Esto pasó porque mientras que el algoritmo multinominal se ocupa para atributos discretos, el Gausiano espera que tu dataset siga una distribución normal. Cosa que se consiguió tras escalar los datos en la sección pasada de procesamiento.

Estos son los resultados que se obtuvieron con NaiveBayes. Mientras que Bayes tuvo un buen resultado de precisión, no fue el mejor con los parámetros que les colocamos a todos los clasificadores. Fue el que tuvo una menor taza de predicción.

```
******* CLASIFICADOR NAIVE BAYES *******

Matriz de confusión:
[[ 9  1  0  0  0  2]
 [ 1 39  0  0  0  1]
 [ 2  2  8  0  0  0]
 [ 0  0  0  4  0  0]
 [ 0  1  0  0  5  2]
 [ 0  1  0  0  0 19]]

Reporte de clasificación:
              precision    recall  f1-score   support

           0       0.75      0.75      0.75        12
           1       0.89      0.95      0.92        41
           2       1.00      0.67      0.80        12
           3       1.00      1.00      1.00         4
           4       1.00      0.62      0.77         8
           5       0.79      0.95      0.86        20

    accuracy                           0.87        97
   macro avg       0.90      0.82      0.85        97
weighted avg       0.88      0.87      0.86        97


Puntaje de precisión:
0.865979381443299
```

FIG 3. RESULTADOS DEL CLASIFICADOR NAIVE BAYES

Neural Network (Multi Layer Perceptron)

Para el tercer y último clasificador se ocupó el MLPClassifier (Multi Layer Perceptron) el cual es una red neuronal.

Para la experimentación y obtención de los mejores parámetros del modelo se corrió un Grid Search el cual permitió probar con distintas combinaciones de parámetros y automáticamente seleccionar la mejor para el modelo.

```
parameter_space = {
    'hidden_layer_sizes': [(50,50,50), (50,100,50), (100,)],
    'activation': ['tanh', 'relu'],
    'solver': ['sgd', 'adam'],
    'alpha': [0.0001, 0.05],
    'learning_rate': ['constant','adaptive'],
}
```

FIG 4. ESPACIO DE PARÁMETROS CON LOS QUE SE PROBÓ LAS COMBINACIONES

Por último me arrojó la combinación de parámetro óptima para mi modelo al correrlo por 300 iteraciones.Los parámetros óptimos fueron:

```
{
'activation': 'tanh',
'alpha': 0.05,
'hidden_layer_sizes': (50, 50, 50),
'learning_rate': ‘constant',
'solver': ‘adam'
}
```

Los resultados obtenidos con dichos valores de parámetros me dieron un porcentaje de predicción de 98% siendo el más alto de los tres modelos seleccionados.

```
******* CLASIFICADOR NEURAL NET*******

Matriz de confusión:
[[11  1  0  0  0  0]
 [ 0 41  0  0  0  0]
 [ 0  0 12  0  0  0]
 [ 0  0  0  4  0  0]
 [ 0  0  0  0  8  0]
 [ 0  0  0  0  0 20]]

Reporte de clasificación:
              precision    recall  f1-score   support

           0       1.00      0.92      0.96        12
           1       0.98      1.00      0.99        41
           2       1.00      1.00      1.00        12
           3       1.00      1.00      1.00         4
           4       1.00      1.00      1.00         8
           5       1.00      1.00      1.00        20

    accuracy                           0.99        97
   macro avg       1.00      0.99      0.99        97
weighted avg       0.99      0.99      0.99        97


Puntaje de precisión:
0.9896907216494846
```

FIG 5. RESULTADOS DEL CLASIFICADOR NEURAL NET