6/4/2021

Proyecto Final: Agentes Basados en Conocimiento

Sistemas inteligentes

Tecnológico de Monterrey

Miguel Ángel Muñoz Vázquez - A01423629
Aarón Pérez Ontiveros – A01422524
ITESM CAMPUS CUERNAVACA

Introducción

Los algoritmos para la clasificación de datos son: supervisados, que parte de un conjunto de datos ya clasificado y que sus atributos sirven para caracterizar dicha clasificación; y los no supervisados, que parte de un conjunto de datos no clasificado y que dicha clasificación depende de un procedimiento estadístico. Los algoritmos utilizados a continuación para la clasificación de datos fueron implementados con la librería scikit-learn utilizando el lenguaje de programación Python 3.9.4. Los algoritmos seleccionados son los siguientes:

MultinomialNaiveBayes: es una variante del algoritmo de Naive Bayes, es de aprendizaje supervisado, se utiliza para el análisis de textos.

DecisionTreeClassifier: es un algoritmo de aprendizaje supervisado que su complejidad depende del número de atributos de entrada (suele ser más rápido que una red neuronal).

RandomForestClassifier: es un algoritmo de aprendizaje supervisado, se compone de arboles de decisión y su predicción está dada por la creación aleatoria de los conjuntos de datos en arboles de decisión.

MLPClassifier: Multi-layer Perceptron, es un algoritmo de aprendizaje supervisado. Consta de múltiples redes neuronales que asignan un peso a los datos para entrenar el modelo.

KNeighborsClassifier: es un algoritmo de aprendizaje supervisado. Memoriza las instancias de entrenamiento como su base de conocimiento y para predecir las clases.

A continuación, se explica el procedimiento de clasificación de los datos. El conjunto de datos consta de dos archivos: el primero con el nombre 'Gold-Ingles.csv' contiene los identificadores de cada instancia y dos atributos más que corresponden a la clasificación, uno para el rango de edad y otro para el género; el otro archivo 'English.txt' contiene los identificadores de cada instancia, un atributo que especifica el lenguaje del texto y por último el texto (inglés en esta práctica)

Descripción del conjunto de datos (género)

Un problema clásico que se puede encontrar en muchos ámbitos diferentes es la de detectar si un texto fue escrito por una mujer o por un hombre, ejemplo de caso real en donde se pudiera necesitar esta detección es la investigación antropológica de textos antiguos, donde es importante saber el género de quien lo escribió como uno de los primeros pasos, para delimitar posibilidades y posteriormente aplicar demás técnicas para obtener más información de ellos, como lo es la identidad de quien lo escribió (personajes históricos sobre todo).

Para los experimentos realizados en este proyecto se tomó en cuenta una base de datos, donde para cada instancia se tiene el identificador de esta, el idioma y el texto correspondiente. Por un lado, el identificador consiste en una cadena alfanumérica de 32 caracteres; por ejemplo: *0526eb9cfcee11c0036f3fa6d11158d5*. Por otro lado, el idioma esta codificado en la representación ISO 639-1, que está formado de 2 caracteres, para el caso de nuestros datos siempre es *en* (de inglés). Finalmente, el texto tiene un formato arbitrario que está delimitado por el carácter de nueva línea *\n*.

Usando el texto, después de cierto preprocesamiento se pretende construir un modelo que tenga la capacidad de predecir el género (*male* o *female*)

Preprocesamiento

El procesamiento de los datos comienza normalizando los textos para eliminar el conjunto de caracteres que no son letras y eliminando aquellas que no pertenecen a un lema que las represente. A partir de este resultado se obtiene la descripción de los datos que se van a procesar como el número de palabras existente por clase, cantidad de textos por clase y el vocabulario resultante del procesamiento anterior.

Para esta clasificación se procesaron los datos para filtrar los datos de la clase de 'genero' junto con sus características.

Las métricas, que se calcularon de la base de datos para conocer más a detalle los datos, se muestran en la siguiente tabla:

Métrica	**Valor**
Número de instancias	5000
Longitud promedio de los textos por clase	male: 2216.11 | female: 2519.98
Número de palabras por clase	male: 764091 | female: 905415
Vocabulario	42232

Clasificación y Evaluación

♣ A continuación se muestra la evaluación con la clasificación de los datos con el filtro de frecuencia

♠ Nota : Todos los nombres de los atributos de las filas y columnas están al revés. El nombre de las filas es el de las columnas y el nombre de las columnas es el de las filas.

♦ DecisionTreeClassifier -> Entrenamiento: 30% de los datos

	female predict	male predicted
female real	1161	1008
male real	635	696

	accuracy	recall	fscore
female	0.53526971	0.64643653	0.58562421
male	0.5229151	0.4084507	0.45864909
exactitude	0.53057143		

- DecisionTreeClassifier -> Entrenamiento: 60% de los datos

	female predic	male predicted
female real	643	520
male real	401	436

	accuracy	recall	fscore
female	0.55288048	0.61590038	0.58269144
male	0.520908	0.45606695	0.48633575
exactitude	0.5395		

- KNeighborsClassifier: 30% de los datos

	female predicted	male predicted
female real	652	566
male real	1144	1138

	accuracy	recall	fscore
female	0.53530378	0.36302895	0.43264764
male	0.49868536	0.66784038	0.57099849
exactitude	0.51142857		

- KNeighborsClassifier: 60% de los datos

	female predicted	male predicted
female real	400	341
male real	644	615

	accuracy	recall	fscore
female	0.5398111	0.3831418	0.4481793
male	0.4884829	0.6433054	0.5553047
exactitude	0.5075		

- MLPClassifier: 30% de los datos

	female predicte	male predicted
female real	954	810
male real	842	894

	accuracy	recall	fscore
female	0.54081633	0.5311804	0.53595506
male	0.51497696	0.52464789	0.51976744
exactitude	0.528		

- MLPClassifier: 60% de los datos

	female predicted	male predicted
female real	559	463
male real	485	493

	accuracy	recall	fscore
female	0.54696673	0.53544061	0.5411423
male	0.50408998	0.51569038	0.5098242
exactitude	0.526		

- MultinomialNaiveBayes: 30% de los datos

	female predicted	male predicted
female real	1007	829
male real	789	875

	accuracy	recall	fscore
female	0.54847495	0.56069042	0.55451542
male	0.52584135	0.51349765	0.5195962
exactitude	0.53771429		

- MultinomialNaiveBayes: 60% de los datos

	female predicted	male predicted
female real	544	419
male real	500	537

	accuracy	recall	fscore
female	0.56490135	0.5210728	0.54210264
male	0.51783992	0.56171548	0.5388861
exactitude	0.5405		

♦ RandomForestClassifier: 30% de los datos

	female predicted	male predicted
female real	1202	920
male real	594	784

	accuracy	recall	fscore
female	0.5664467	0.669265	0.6135784
male	0.5689405	0.4600939	0.5087605
exactitude	0.5674286		

♦ RandomForestClassifier: 60% de los datos

	female predicted	male predicted
female real	723	547
male real	321	409

	accuracy	recall	fscore
female	0.56929134	0.69252874	0.62489196
male	0.56027397	0.42782427	0.485172
exactitude	0.566		

♣ A continuación se muestra la evaluación con la clasificación de los datos con el filtro de TF-IDF

♠ Nota : Todos los nombres de los atributos de las filas y columnas están al revés. El nombre de las filas es el de las columnas y el nombre de las columnas es el de las filas.

- DecisionTreeClassifier -> Entrenamiento: 30% de los datos

	female predicte	male predicted
female real	1178	924
male real	633	765

	accuracy	recall	fscore
female	0.56041865	0.65046935	0.60209558
male	0.5472103	0.45293073	0.49562682
exactitude	0.55514286		

- DecisionTreeClassifier -> Entrenamiento: 60% de los datos

	female predicted	male predicted
female real	754	614
male real	275	357

	accuracy	recall	fscore
female	0.551169591	0.73275024	0.62911973
male	0.564873418	0.3676622	0.44541485
exactitude	0.5555		

- KNeighborsClassifier: 30% de los datos

	female predicted	male predicted
female real	427	361
male real	1384	1328

	accuracy	recall	fscore
female	0.54187817	0.23578134	0.32858792
male	0.48967552	0.78626406	0.6034992
exactitude	0.50142857		

- KNeighborsClassifier: 60% de los datos

	female predicted	male predicted
female real	330	249
male real	699	722

	accuracy	recall	fscore
female	0.56994819	0.32069971	0.41044776
male	0.50809289	0.74356334	0.60367893
exactitude	0.526		

- MLPClassifier: 30% de los datos

	female predicted	male predicted
female real	741	558
male real	1070	1131

	accuracy	recall	fscore
female	0.5704388	0.40916621	0.47652733
male	0.51385734	0.669627	0.581491
exactitude	0.53485714		

- MLPClassifier: 60% de los datos

	female predicted	male predicted
female real	595	506
male real	434	465

	accuracy	recall	fscore
female	0.5404178	0.57823129	0.55868545
male	0.51724138	0.47888774	0.4973262
exactitude	0.53		

- MultinomialNaiveBayes: 30% de los datos

	female predicted	male predicted
female real	897	714
male real	914	975

	accuracy	recall	fscore
female	0.55679702	0.49530646	0.52425482
male	0.51614611	0.57726465	0.54499721
exactitude	0.53485714		

- MultinomialNaiveBayes: 60% de los datos

	female predicted	male predicted
female real	563	439
male real	466	532

	accuracy	recall	fscore
female	0.56187625	0.54713314	0.5544067
male	0.53306613	0.54788877	0.54037583
exactitude	0.5475		

- RandomForestClassifier: 30% de los datos

	female predicted	male predicted
female real	999	767
male real	812	922

	accuracy	recall	fscore
female	0.56568516	0.55162893	0.55856863
male	0.53171857	0.54588514	0.53870874
exactitude	0.54885714		

- RandomForestClassifier: 60% de los datos

	female predicted	male predicted
female real	751	586
male real	278	385

	accuracy	recall	fscore
female	0.56170531	0.72983479	0.63482671
male	0.58069382	0.39649846	0.47123623
exactitude	0.568		

Descripción del conjunto de datos (edad)

Otra forma de clasificar los datos de la base de datos a la que tenemos acceso es por medio de la edad. La clasificación se hace por rango de edades (10s, 20s y 30s). Se puede recabar mucha información en las redes sociales donde todas las personas escriben publicaciones de cierto tipo y dependiendo de la edad se expresan a su manera. En este caso se desconoce la fuente de los textos, pero leyéndolos se puede apreciar el tipo de lenguaje y por los temas se puede inferir si se trata de una puberto, un adolescente o un adulto joven.

Preprocesamiento

El procesamiento de los datos comienza normalizando los textos para eliminar el conjunto de caracteres que no son letras y eliminando aquellas que no pertenecen a un lema que las represente. A partir de este resultado se obtiene la descripción de los datos que se van a procesar como el numero de palabras existente por clase, cantidad de textos por clase y el vocabulario resultante del procesamiento anterior.

Para esta clasificación se procesaron los datos para filtrar los datos de la clase de 'edad' junto con sus características.

Las métricas, que se calcularon de la base de datos para conocer más a detalle los datos, se muestran en la siguiente tabla:

Métrica	**Valor**
Número de instancias	5000
Longitud promedio de los textos por clase	10s: 2635.2 | 20s:2079.09 | 30s:2521.29
Número de palabras por clase	10s:128914 | 20s:525648 | 30s:1014944
Vocabulario	42232

Con estas métricas conocemos la cantidad de datos existentes por cada clase que vamos a trabajar, el número de instancias y el vocabulario.

Clasificación y Evaluación

♣ A continuación se muestra la evaluación con la clasificación de los datos con el filtro de frecuencia

♠ Nota : Todos los nombres de los atributos de las filas y columnas están al revés. El nombre de las filas es el de las columnas y el nombre de las columnas es el de las filas.

♦ DecisionTree -> Entrenamiento: 30% de los datos

	10s predicted	20s predicted	30s predicted
10s real	7	20	42
20s real	69	470	423
30s real	172	762	1535

	accuracy	recall	fscore
10s	0.10144928	0.02822581	0.04416404
20s	0.48856549	0.37539936	0.42457091
30s	0.62170919	0.7675	0.68695458
exactitude	0.57485714		

- DecisionTree -> Entrenamiento: 60% de los datos

	10s predicted	20s predicted	30s predicted
10s real	2	6	11
20s real	33	264	224
30s real	104	448	908

	accuracy	recall	fscore
10s	0.1052632	0.0143885	0.0253165
20s	0.5067179	0.367688	0.4261501
30s	0.6219178	0.7944007	0.6976566
exactitude	0.587		

- KNeighborClassifier -> Entrenamiento: 30% de los datos

AO	10s predicted	20s predicted	30s predicted
10s real	7	36	62
20s real	101	653	787
30s real	140	563	1151

	accuracy	recall	fscore
10s	0.06666667	0.02822581	0.03966006
20s	0.42375081	0.5215655	0.46759757
30s	0.62081985	0.5755	0.5973015
exactitude	0.51742857		

- DecisionTree -> Entrenamiento: 60% de los datos

AO	10s predicted	20s predicted	30s predicted
10s real	10	34	53
20s real	53	369	453
30s real	76	315	637

	accuracy	recall	fscore
10s	0.1030928	0.0719424	0.0847458
20s	0.4217143	0.5139276	0.4632768
30s	0.6196498	0.5573053	0.5868263
exactitude	0.508		

- MLPClassifier -> Entrenamiento: 30% de los datos

AO	10s predicted	20s predicted	30s predicted
10s real	0	0	0
20s real	94	529	628
30s real	154	723	1372

	accuracy	recall	fscore
10s			
20s	0.42286171	0.42252396	0.42269277
30s	0.61004891	0.686	0.64579901
exactitude	0.54314286		

- MLPClassifier -> Entrenamiento: 60% de los datos

AO	10s predicted	20s predicted	30s predicted
10s real	0	0	0
20s real	54	319	386
30s real	85	399	757

	accuracy	recall	fscore
10s			
20s	0.42028986	0.44428969	0.43195667
30s	0.60999194	0.66229221	0.63506711
exactitude	0.538		

- MultinomialNaiveBayes -> Entrenamiento: 30% de los datos

AO	10s predicted	20s predicted	30s predicted
10s real	2	19	15
20s real	41	297	234
30s real	205	936	1751

	accuracy	recall	fscore
10s	0.05555556	0.00806452	0.01408451
20s	0.51923077	0.23722045	0.32565789
30s	0.60546335	0.8755	0.71586263
exactitude	0.58571429		

- MultinomialNaiveBayes -> Entrenamiento: 60% de los datos

AO	10s predicted	20s predicted	30s predicted
10s real	3	23	20
20s real	34	243	236
30s real	102	452	887

	accuracy	recall	fscore
10s	0.06521739	0.02158273	0.03243243
20s	0.47368421	0.33844011	0.39480097
30s	0.61554476	0.776028	0.68653251
exactitude	0.5665		

- RandomForest -> Entrenamiento: 30% de los datos

AO	10s predicted	20s predicted	30s predictec
10s real	0	0	0
20s real	25	269	167
30s real	223	983	1833

	accuracy	recall	fscore
10s			
20s	0.5835141	0.21485623	0.31406888
30s	0.60315893	0.9165	0.7275253
exactitude	0.60057143		

- RandomForest -> Entrenamiento: 60% de los datos

AO	10s predicted	20s predicted	30s predicted
10s real	0	0	0
20s real	9	104	52
30s real	130	614	1091

	accuracy	recall	fscore
10s			
20s	0.63030303	0.1448468	0.23556059
30s	0.59455041	0.95450569	0.73270651
exactitude	0.5975		

♣ A continuación se muestra la evaluación con la clasificación de los datos con el filtro de TF-IDF

♠ Nota : Todos los nombres de los atributos de las filas y columnas están al revés. El nombre de las filas es el de las columnas y el nombre de las columnas es el de las filas.

♦ DecisionTree -> Entrenamiento: 30% de los datos

	10s predicted	20s predicted	30s predicted
10s real	6	15	54
20s real	66	551	562
30s real	167	691	1388

	accuracy	recall	fscore
10s	0.08	0.0251046	0.03821656
20s	0.46734521	0.43834527	0.45238095
30s	0.61798753	0.69261477	0.65317647
exactitude	0.55571429		

- DecisionTree -> Entrenamiento: 60% de los datos

	10s predicted	20s predicted	30s predicted
10s real	0	8	18
20s real	26	295	237
30s real	95	456	865

	accuracy	recall	fscore
10s	0	0	
20s	0.52867384	0.3886693	0.44798785
30s	0.61087571	0.77232143	0.68217666
exactitude	0.58		

- KNeighborClassifier -> Entrenamiento: 30% de los datos

	10s predicted	20s predicted	30s predicted
10s real	52	451	539
20s real	69	372	562
30s real	118	434	903

	accuracy	recall	fscore
10s	0.04990403	0.21757322	0.08118657
20s	0.37088734	0.29594272	0.32920354
30s	0.62061856	0.4505988	0.52211622
exactitude	0.37914286		

- KNeighborClassifier -> Entrenamiento: 60% de los datos

	10s predicted	20s predicted	30s predicted
10s real	18	188	175
20s real	50	321	476
30s real	53	250	469

	accuracy	recall	fscore
10s	0.04724409	0.14876033	0.07171315
20s	0.37898465	0.4229249	0.39975093
30s	0.60751295	0.41875	0.49577167
exactitude	0.404		

- MLPClassifier -> Entrenamiento: 30% de los datos

	10s predicted	20s predicted	30s predicted
10s real	0	0	0
20s real	89	556	653
30s real	150	701	1351

	accuracy	recall	fscore
10s		0	
20s	0.42835131	0.44232299	0.43522505
30s	0.61353315	0.6741517	0.6424156
exactitude	0.54485714		

- MLPClassifier -> Entrenamiento: 60% de los datos

	10s predicted	20s predicted	30s predicted
10s real	0	0	1
20s real	44	323	357
30s real	77	436	762

	accuracy	recall	fscore
10s	0	0	
20s	0.4461326	0.42555995	0.43560351
30s	0.59764706	0.68035714	0.63632568
exactitude	0.5425		

- MultinomialNaiveBayes -> Entrenamiento: 30% de los datos

	10s predicted	20s predicted	30s predicted
10s real	37	264	286
20s real	56	326	403
30s real	146	667	1315

	accuracy	recall	fscore
10s	0.06303237	0.15481172	0.08958838
20s	0.41528662	0.25934765	0.31929481
30s	0.61795113	0.65618762	0.63649564
exactitude	0.47942857		

- MultinomialNaiveBayes -> Entrenamiento: 60% de los datos

	10s predicted	20s predicted	30s predicted
10s real	22	163	187
20s real	26	220	230
30s real	73	376	703

	accuracy	recall	fscore
10s	0.05913978	0.18181818	0.08924949
20s	0.46218487	0.28985507	0.3562753
30s	0.61024306	0.62767857	0.61883803
exactitude	0.4725		

- RandomForest -> Entrenamiento: 30% de los datos

	10s predicted	20s predicted	30s predicted
10s real	0	0	0
20s real	1	19	12
30s real	238	1238	1992

	accuracy	recall	fscore
10s		0	
20s	0.59375	0.01511535	0.02948022
30s	0.57439446	0.99401198	0.72807018
exactitude	0.57457143		

- RandomForest -> Entrenamiento: 60% de los datos

	10s predicted	20s predicted	30s predicted
10s real	0	0	0
20s real	0	15	4
30s real	121	744	1116

	accuracy	recall	fscore
10s		0	
20s	0.78947368	0.01976285	0.03856041
30s	0.56335184	0.99642857	0.71976782
exactitude	0.5655		

Conclusión

Pudimos observar a primera vista que para la característica por frecuencia simple, RandomForestClassifier con 30% de los datos como entrenamiento fue el mejor algoritmo con una exactitud de 0.57 para los datos de género. De manera similar con 30% de los datos como entrenamiento, RandomForestClassifier fue el mejor algoritmo con exactitud de 0.60 para los datos de edad. Por otro lado, DecisionTree con entrenamiento de 60% fue el que peor rindió para los datos de edad con una exactitud de 0.50; mientras que KNeighborsClassifier con 60% de los datos fue el peor para los datos de género. Probablemente random forest fue el mejor con pocos datos de entrenamiento porque con ello no se presentó problema de overfitting. Decision Tree fue malo porque presentó el problema de overfitting para edades, ya que teníamos más clases. KNeighborsClassifier fue malo probablemente porque se trataban de sólo dos clases. En general se podría decir que Random Forest es la mejor opción en general.

Con la caracteristica de frecuencia inversa sucede que el algoritmo de random forest obtiene mejores predicciones.

Bibliografía

Russell, S., Norvig P. (2004). *Inteligencia Artificial: un enfoque moderno.*

Russell, S., Norvig P. (2016). *Artificial Intelligence: a modern approach.*

Clasificación supervisada y no supervisada en ArcGIS | El blog de franz. (2021). Retrieved 9 June 2021, from https://acolita.com/clasificacion-supervisada-no-supervisada-en-arcgis/
1.9. Naive Bayes — scikit-learn 0.24.2 documentation. (2021). Retrieved 9 June 2021, from https://scikit-learn.org/stable/modules/naive_bayes.html

Navlani, A. (2021). Retrieved 9 June 2021, from https://www.datacamp.com/community/tutorials/decision-tree-classification-python

Navlani, A. (2021). Retrieved 9 June 2021, from https://www.datacamp.com/community/tutorials/random-forests-classifier-python

Robinson, S. (2021). Introduction to Neural Networks with Scikit-Learn. Retrieved 9 June 2021, from https://stackabuse.com/introduction-to-neural-networks-with-scikit-learn

1.6. Nearest Neighbors — scikit-learn 0.24.2 documentation. (2021). Retrieved 9 June 2021, from https://scikit-learn.org/stable/modules/neighbors.html