Proyecto Análisis de Datos

Parte 2: Reducción de dimensionalidad e información no estructurada

José Manuel Bustos Muñoz
Santiago Martínez De La Riva

Con la base de datos que hayáis elegido en la práctica 1 (GiveMeSomeCredit / BikeSharing-Dataset) realizar lo siguiente:

1. Detectar las variables más informativas utilizando alguno de los métodos de Selección de Características (Feature Selection) por filtrado vistos en clase.

El método que hemos seleccionado para hacer el estudio de las variables que más nos aportan sobre la variable target seleccionada "cnt", es el método visto en clase llamado Mutual Information. Aún así, para verificar los resultados hemos comprobado dichos resultados con los obtenidos por el algoritmo "f_classif", y hemos podido comprobar como los resultados de las variables que más información aportan del target son similares.

Debido a que elegimos como target la variable "cnt", creemos que no debemos de tener en cuenta las features "casual" y "registered", ya que como vimos en la primera parte del proyecto están fuertemente correlacionadas con el target, a parte que la variable "cnt" es el resultado de la suma de estas. Y es por estos motivos son por los que hemos decidido quitarlas del cojunto de features a analizar.

Además al trabajar con la variable "cnt" como target y ser una variable contínua, la hemos discretizado y hemos creado 4 clases diferentes:

- clase 0: (-infinito, 250]
- clase 1: (250, 500]
- clase 2: (500, 750]
- clase 3: (750, +infinito]

Como consecuencia de esto hemos creado una nueva variable "cnt_class", que es la que finalmente hemos seleccionado como variable target.

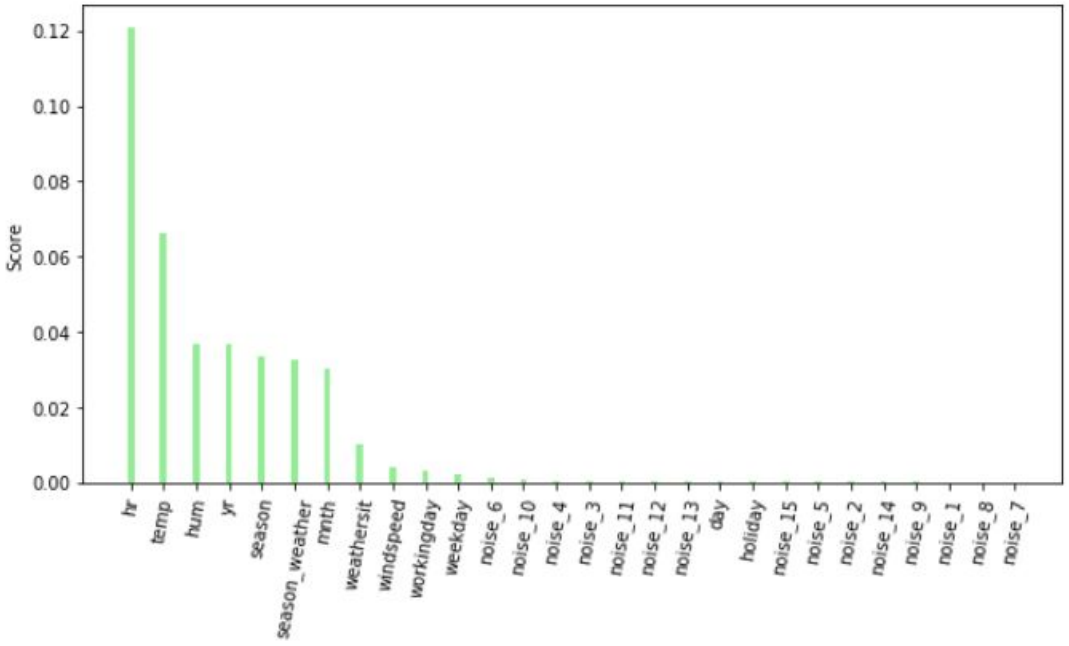
En el dataframe sobre el cual hemos trabajado, hemos definido como categóricas aquellas variables o features discretas, y sobre las variables continuas hemos calculado los binsx correspondientes, dentro de la parte del código correspondiente al método de "mutual_information", a través de una fórmula basada en el número de valores de cada variable contínua.

Una vez tenemos el dataframe como queremos, seleccionadas el conjunto de variables a analizar y el target, hemos ejecutado los dos métodos, y como ya habíamos visto en el análisis previo durante la primera parte del proyecto, vemos como las variables que aportan más información a la variable target, las que tienen más peso, son las features:

["hr", "temp", "hum", "yr", "season", "season_weathersit", "mnth", "weathersit"]

Resultados obtenidos después de aplicar el método de MI:

```
hr          score=0.120636
temp        score=0.066054
hum         score=0.037019
yr          score=0.036812
season      score=0.033679
season_weather    score=0.032705
mnth        score=0.030185
weathersit          score=0.010334
windspeed           score=0.004232
workingday          score=0.003422
weekday             score=0.002509
noise_6             score=0.001299
noise_10            score=0.000936
noise_4             score=0.000700
noise_3             score=0.000605
noise_11            score=0.000584
noise_12            score=0.000534
noise_13            score=0.000522
day         score=0.000517
holiday             score=0.000488
noise_15            score=0.000465
noise_5             score=0.000436
noise_2             score=0.000415
noise_14            score=0.000414
noise_9             score=0.000370
noise_1             score=0.000250
noise_8             score=0.000211
noise_7             score=0.000148
```



ranking_variables_clasificacion - mutual_info

2. Realizar ese mismo análisis utilizando ahora alguno de los métodos de Selección de Características por wrapping vistos en clase.

¿Obtienes lo mismo en los dos casos? En vista de los resultados obtenidos y los que obtuviste en la práctica 1:

Resultados obtenidos aplicando la estrategia 3 de Wrapping:

```
[ 0.02524094  0.04111697  0.03981221  0.14924883  0.00399696  0.02738659
  0.02040441  0.01723389  0.07771286  0.06931366  0.03075553  0.02804129
  0.03102636  0.02921216  0.02928626  0.02943998  0.02809275  0.02863518
  0.02908204  0.02915324  0.02994856  0.02846924  0.02826384  0.02981379
  0.02964904  0.02970452  0.02942594  0.03053297]

Elegidos:
['yr' 'mnth' 'hr' 'temp' 'hum']
Score de my_model en train: 0.862
Score de my_model en test : 0.807
```

Como resultado de aplicar la estrategia 3 basada en entrenar un modelo auxiliar que esté diseñado también para dar la importancia de los atributos, y seleccionar usando esta información los atributos que usaremos en nuestro modelo, hemos obtenido el siguiente conjunto de variables: ['yr', 'mnth', 'hr', 'temp', 'hum'].

Si nos fijamos con las variables de más puntuación aportadas con el algoritmo de filtrado de MI, podemos ver que las variables "hr", "temp", "hum" y "yr", eran de las que más scoring tenían por lo que parece verificar que para ambos métodos las variables seleccionadas son las que más información nos aportan sobre la variable target elegida, lo que de la misma forma se corresponde con los análisis preliminares hechos en la primera parte del proyecto, en donde claramente veíamos en esa exploración como las horas cercanas a la entrada al trabajo 7-9 y salida del mismo 8-9, había un incremento sustancial del uso del servicio de renting.

También a partir de la variable "temp", podemos corroborar que hay una correlación entre la buena temperatura y el uso incremental del servicio de renting. También veíamos como la variable "mnth" nos aportaba información relevante respecto la variable target, pues los meses cercanos a las estaciones con mejores temperaturas o periodos vacacionales había un incremento del uso de las bicicletas.

- ¿Hay alguna variable que no sea informativa?

Si nos fijamos la variable "day", parece no ser muy informativa, al contrario de lo que analizamos nosotros en nuestro primer análisis en el preprocesado de

datos. Pensabamos que a través de esta variable a primeros de mes cuando los usuarios pueden tener más dinero, harían menos uso del servicio de renting, y usarían otros medios de transporte más caros, como el taxi, el metro o el autobús, pero vemos ahora después del estudio realizado como el score de esta variable es muy baja, y no nos aporta información sobre la variable target, luego la podemos descartar sin ninguna duda.

También parece que la varaible "weekday" no aporta mucha información sobre la variable target, cómo si podíamos pensar en un principio. Es decir, que el servicio de renting estuviera relacionado con el día de la semana y que el uso fuera mayor los fines de semana que entre semana, pero podemos comprobar como el score de esta variable es también baja por lo que no nos aporta información alguna.

- ¿Hay alguna variable que sea informativa pero redundante?

Entendemos que al seleccionar como target la variable “cnt”, las variables “casual” y “registered”, son redundantes, pues es la suma de ambas. Otra estrategía que podíamos haber seguido en lugar de eliminar estas, sería haber analizado como target los usuarios casuales y los registrados, pero nuestra intención era establecer qué variables nos aportaban más información para tratar de inferir en la medida de los posible bajo qué circunstancias, o mejor dicho, cuándo se hace un mayor uso del servicio de reting, con el fin como explicabamos anteriormente de poder mejorar el servicio, preveyendo en el futuro esos momentos de máximo uso.

- ¿Qué variables quitarías en vista a estos resultados de análisis posteriores? ¿Cuáles dejarías?.

Las variables que quitaríamos en función de la variable target seleccionada, serían aquellas que menos score nos han dado en los métodos aplicados, a saber: "day", "weekday", "workingday" y "holiday", “registered” y “casual”. Estas dos últimas variables sabemos que son importantes para estudios de inferencia separados pero no para el caso de estudio sobre el que nos estamos centrando.

Por otro lado el conjunto de variables que mantendríamos para poder analizar con más profundidad serían: "temp", "yr", "hum", "hr", "season", "wheathersit", "month", "windspeed".

3. Visualiza la proyección de tu base de datos en los primeros componentes principales obtenidos por PCA, junto a la proyección de las variables originales (tal y como hemos visto en los ejemplos en clase). Realizaremos dos gráficos diferentes:

- **Gráfico 1:** Eje horizontal: proyección sobre la primera componente. Eje vertical: proyección sobre la segunda componente.

- **Gráfico 2:** Eje horizontal: proyección sobre la segunda componente. Eje vertical: proyección sobre la tercera componente.

¿Qué puedes decir en base a estos gráficos sobre la correlación entre las diferentes variables originales?.

Principal Component Analysis (PCA) es un método estadístico que permite simplificar la complejidad de espacios muestrales con muchas dimensiones a la vez que conserva su información.

El proceso de PCA identifica aquellas direcciones en las que la varianza es mayor. Tanto la proporción de varianza explicada como la proporción de varianza explicada acumulada son dos valores de gran utilidad a la hora de decidir el número de componentes principales a utilizar en los análisis posteriores. Si se calculan todas las componentes principales de un set de datos, entonces, aunque transformada, se está almacenando toda la información presente en los datos originales. El sumatorio de la proporción de varianza explicada acumulada de todas las componentes es siempre 1.

Se usa la proporción acumulada de varianza para determinar la cantidad total de varianza que explican los componentes principales. Se conservan los componentes principales que explican un nivel aceptable de varianza. El nivel aceptable depende de la aplicación específica. Para propósitos descriptivos, es posible que solo se necesite explicar el 80% de la varianza. Sin embargo, si se desea realizar otros análisis con los datos, se recomienda que los componentes principales expliquen por lo menos el 90% de la varianza. Con los valores propios de la varianza también se puede determinar el número de componentes principales. Conservar los componentes principales con los valores propios más grandes. Por ejemplo, según el criterio de Kaiser, se usan solo los componentes principales con valores propios que son mayores que 1.

Viendo los datos de las varianzas explicadas y la acumulada, y la gráfica donde se muestran, se puede observar que con los 11-12 primeros componentes principales se obtendría prácticamente toda la varianza acumulada, la totalidad de la información. Por tanto bastaría con ese número de dimensiones. Mirando por los valores propios se aprecia como los dos primeros componentes principales tienen una mayor varianza y por lo tanto recogen mayor parte de información que el resto, y la varianza va disminuyendo en los siguientes

componentes principales. Superior a 1 sólo tendrían valor propio los 7 primeros componentes principales, y el octavo prácticamente cercano a 1, así que podrían seleccionarse esos bajo otro criterio.

En las gráficas de influencias se muestran visualmente los resultados para los dos pares de componentes, identificando con las flechas y su posición las influencias de cada atributo en relación a los componentes principales del eje x y el eje y de la gráfica.

Para el componente 1 hay bastantes variables con influencia positiva, las que más las variables de season, month y season_weather. También temp, casual y registered tienen buena influencia positiva en este componente.

En el componente 2 las variables con mayor influencia positiva son casual, registered, hr, temp y windspeed. En cuanto a influencia negativa destacan las variables de hum con la mayor influencia, y weathersit.

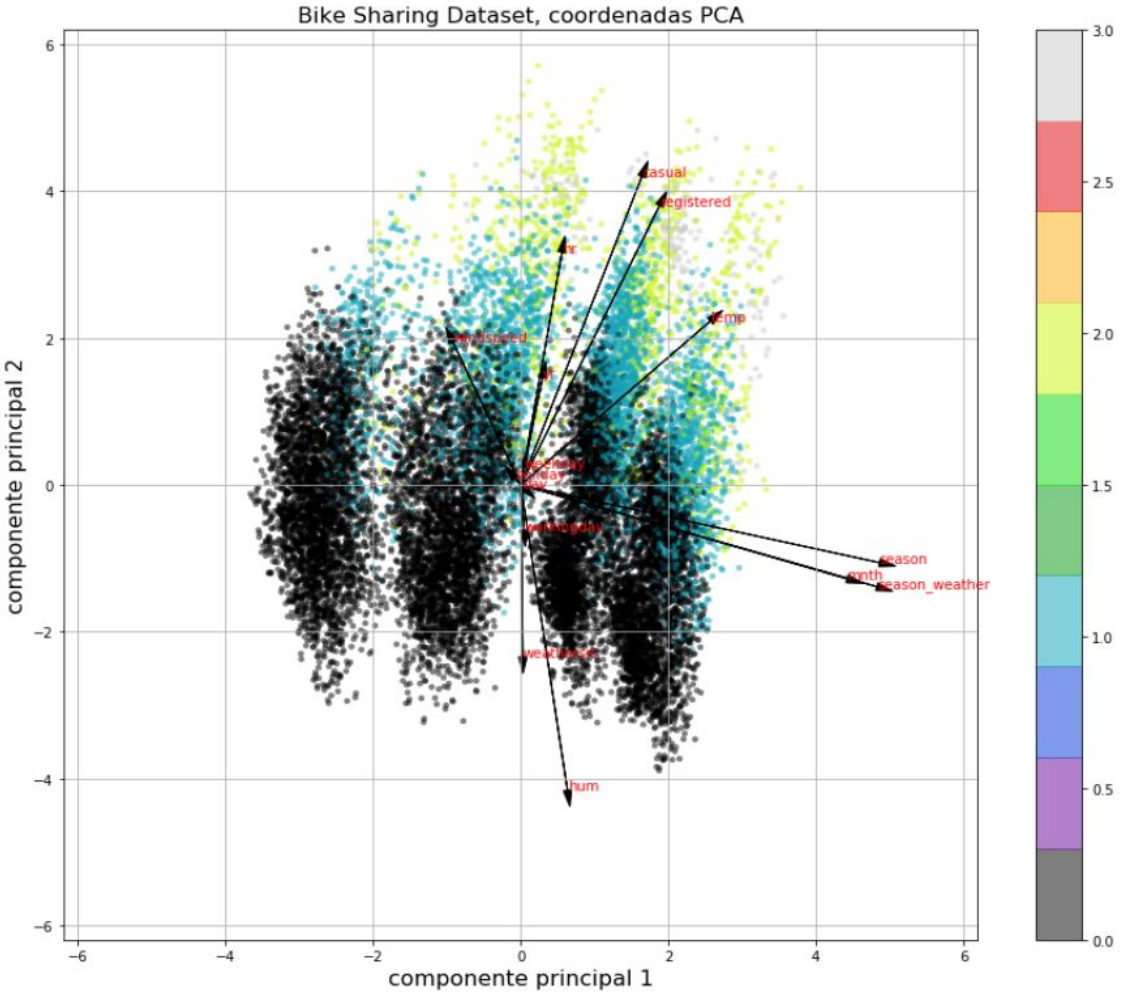
Por último, para el componente 3 las variables con mayor influencia positiva son holiday y casual. Y las de influencia negativa son workingday, weekday y registered.

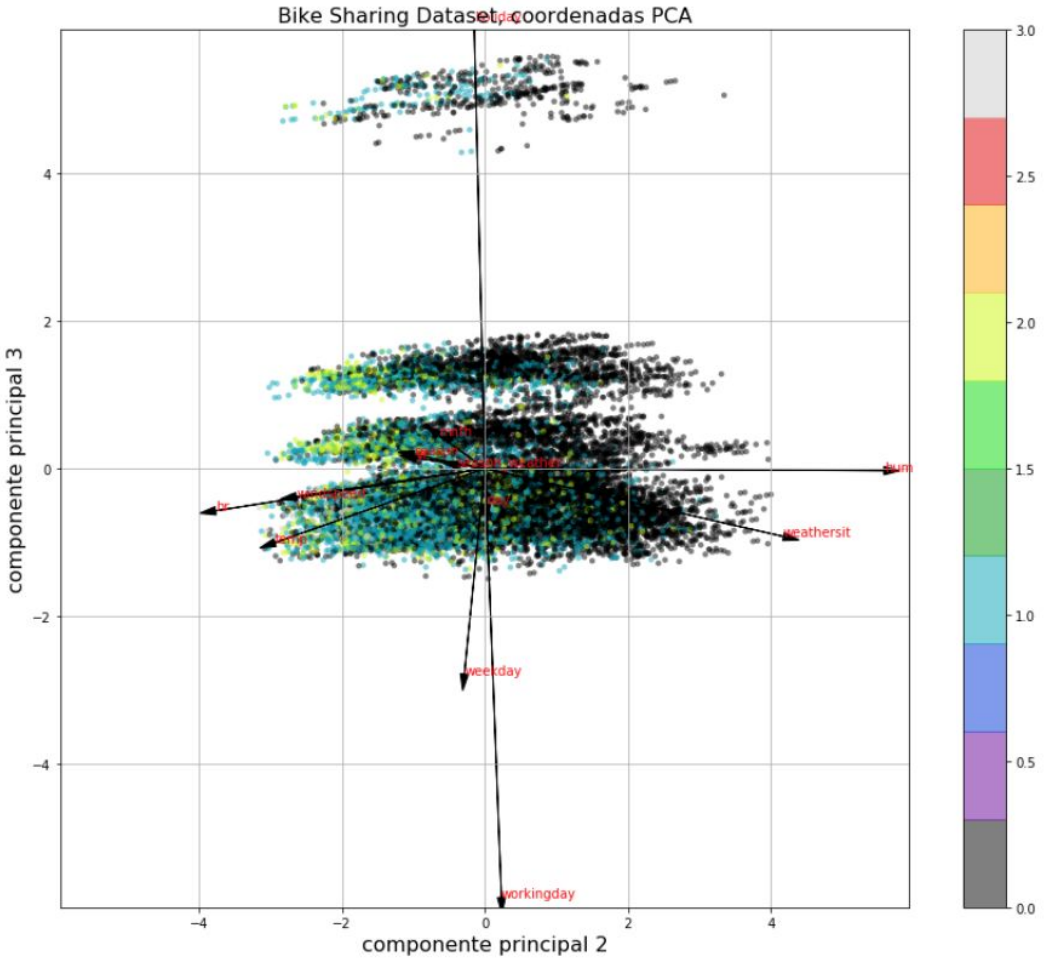
También se aprecia como variables que están de algún modo correlacionadas tienen casi la misma influencia para el mismo componente principal. Por ejemplo en el componente 3 que tienen una influencia más cercana las variables de holiday y casual, en contraposicion de workingday y registered, puede deberse a que en día de vacaciones es cuando más se suelen producir usos casuales, y en los días laborables la mayoría de usos son registrados.

O como variables que pueden considerarse contrarias tienen la influencia opuesta. Por ejemplo para el componente 3 la variable holiday tiene una gran influencia positiva, y la variable workingday practicamente la misma influencia pero negativa. Esto tiene lógica ya que cuando un día pertenece a una de esas categorías no pertenece a la otra.

Podemos ver en ambos gráficos tanto para la componente 1 y 2, como para la 2 y la 3, que las variables que están fuertemente relacionados van en la misma dirección. Así en el primer gráfico vemos como en la misma dirección tenemos varios grupos de variables:

- Casual y registered.
- season, mnth y season_weather.
- weekday, holiday y day.
- wheathersit y más o menos en la misma dirección hum.
- Y por otro lado temp.


Bike Sharing Dataset, coordenadas PCA
componente principal 1
componente principal 2
casual
registered
hr
temp
windspeed
season
season_weather
hum



¿Te podría ayudar la proyección PCA para realizar una selección de variables? En caso de que sí: ¿cómo? (sólo arguméntalo). En caso de que no: ¿por qué?.

Creemos que si sería un método que puede ayudar para la selección de las variables adecuadas, sobretodo para reducir algo la dimensionalidad y poder quedarnos con un espacio más simple sin perder la información importante para analizar los datos.
Nos podría ayudar a realizar una selección de variables, ya que las variables que estén en la misma dirección, si quiere decir que están fuertemente correlacionadas y que nos pueden dar el mismo tipo de información, nos permitiría seleccionar alguna de ellas. Por ejemplo, para el caso del grupo de season, mnth y season_weather, podríamos quedarnos simplemente con season o con mnth dependiendo de la granularidad o profundidad con la que queramos realizar el estudio.

Además las variables que están muy en el epicentro quiere decir que aportan muy poca información sobre la variable target, o sea que son practicamente independientes y podrías desecharlas. Es decir, las variables: weekday, holiday, workingday y day. Esto cuadra bastante con los resultados obtenidos en los apartados anteriores.

4. Implementa una función sencilla que detecte outliers usando las proyecciones PCA. Ayuda: tomando los datos de entrenamiento, calcula la proyección PCA en el número de componentes dado como argumento a la función, calcula el valor promedio de las componentes, y detecta qué puntos se alejan "demasiado" de ese promedio. Una opción sencilla es, una vez calculadas las proyecciones PCA de cada punto, calcular la distancia en ese espacio de todos los puntos al promedio, y considerar como outlier el P% de puntos que tengan distancia mayor (P estaría dado como argumento a nuestra función detectora de outliers)

Se ha creado una función para calcular los posibles valores extremos para cada atributo. Se le pasa a la función el valor % para calcular los percentiles que delimiten los valores válidos y los outliers.

```
def detectar_outliers(x, p):
    diccionario_ = {}
    atributos = x.keys()
    outlier_counts = defaultdict(lambda: 0)

    for atributo in atributos:
        values = x[atributo]

        # Calcular los percentiles
        P1 = np.percentile(values,p)
        P3 = np.percentile(values,100-p)

        # rango
        step = 1.5*(P3-P1)

        # conseguir indices de los outliers
        value_min = P1 - step
        value_max = P3 + step
        ioutliers = (values < value_min) | (values > value_max)

        # conseguir outliers
        outliers = x[ioutliers]
        diccionario_[atributo] = (value_min,value_max)

        # Ver los outliers detectados
        print ("Puntos considerados outliers para el atributo '{}':".format(atributo))
        display(outliers)

        # añadir outliers al diccionario
        for i in outliers.index.values:
            outlier_counts[i] += 1

    return outliers
```

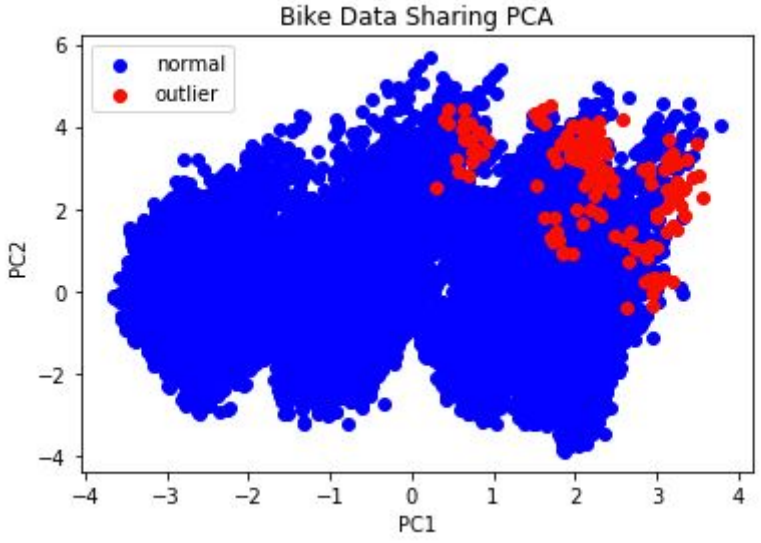
Al aplicar la función al dataframe se obtienen los registros outliers para cada uno de los atributos, por ejemplo para “windspeed”.

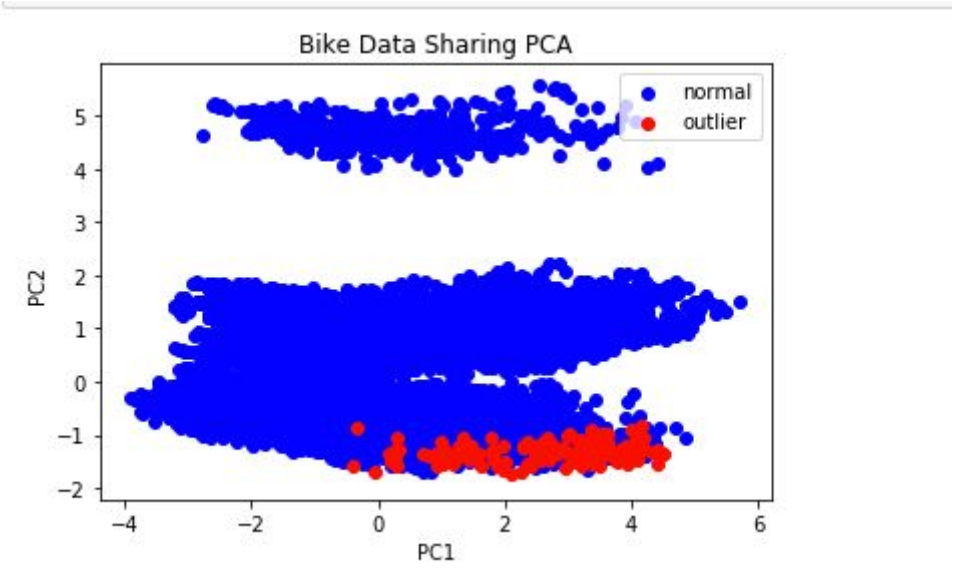
Puntos considerados outliers para el atributo 'windspeed':

	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	hum	windspeed	season_weather	day	casual	registered
1007	1	0	2	15	0	1	1	1	0.56	0.21	0.6567	1	14	19	71
1009	1	0	2	17	0	1	1	1	0.46	0.33	0.6119	1	14	25	218
1010	1	0	2	18	0	1	1	1	0.40	0.40	0.6119	1	14	11	194
1014	1	0	2	22	0	1	1	1	0.34	0.46	0.6567	1	14	1	44
1017	1	0	2	1	0	2	1	1	0.30	0.42	0.7761	1	15	0	5
1018	1	0	2	2	0	2	1	1	0.28	0.41	0.6866	1	15	1	2
1119	1	0	2	9	0	6	0	1	0.40	0.16	0.6567	1	19	18	37
1123	1	0	2	13	0	6	0	1	0.44	0.16	0.6119	1	19	52	103
1124	1	0	2	14	0	6	0	1	0.46	0.15	0.6567	1	19	102	94

5. Repite la visualización que realizaste en el apartado 3 marcando en colores diferentes los puntos que serían outliers y los puntos que no serían outliers (según la función que has implementado en el apartado 4). ¿Coinciden los puntos que detectas como outliers con aquellos que considerabas outliers en la práctica 1?

Al aplicar la función de cálculo de outliers al dataframe y obtener los registros que pueden ser considerados extremos, se procede a pintar de nuevo las dos gráficas con los primeros componentes principales, dibujando de un color distinto los registros outliers.





APARTADOS OPCIONALES:

La entrega solo de los apartados 1 a 5 es suficiente para optar al 10 en esta práctica. Se proponen además los siguientes dos apartados opcionales que ayudarán a subir la nota en caso de que la calificación obtenida en los apartados 1 a 5 no sea 10:

6. Toma el notebook "text_analytics_2classes.ipynb" y complétalo añadiendo un modelo para la clasificación automática en las dos categorías (CreditCard/Mortgage).

Evalúa el score de dicho modelo en training y en test. ¿Qué scores tienes? ¿hay sobreajuste o el sistema está generalizando bien?

7. Toma el notebook del apartado 6 y amplía ahora las clases de 2 a 5 (CreditCard / DebtCollection / CreditReporting/ Mortgage / Other).

Construye y evalúa ahora un clasificador en este nuevo problema de 5 clases. ¿Qué scores obtienes en training y en test? De las 5 clases, ¿qué pares de clases confunde más entre ellas?