meteoregresion

October 16, 2020

Predicción humedad a las 3 pm a partír de los features calculados entre las 8:55 am y 9:04 am por método de regresión

[3]:
```
import pandas as pd
from sklearn import preprocessing
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LinearRegression
from math import sqrt

import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

Cargamos los datos

[4]:
```
data = pd.read_csv('meteo/diario.csv')
data.columns
data.shape
```

[4]: (1095, 11)

[7]:
```
data.head(10)
```

[7]:
```
   number  air_pressure_9am  air_temp_9am  avg_wind_direction_9am  \
0       0        918.060000     74.822000              271.100000
1       1        917.347688     71.403843              101.935179
2       2        923.040000     60.638000               51.000000
3       3        920.502751     70.138895              198.832133
4       4        921.160000     44.294000              277.800000
5       5        915.300000     78.404000              182.800000
6       6        915.598868     70.043304              177.875407
7       7        918.070000     51.710000              242.400000
8       8        920.080000     80.582000               40.700000
9       9        915.010000     47.498000              163.100000

   avg_wind_speed_9am  max_wind_direction_9am  max_wind_speed_9am  \
```

```
0           2.080354          295.400000          2.863283
1           2.443009          140.471548          3.533324
2          17.067852           63.700000         22.100967
3           4.337363          211.203341          5.190045
4           1.856660          136.500000          2.863283
5           9.932014          189.000000         10.983375
6           3.745587          186.606696          4.589632
7           2.527742          271.600000          3.646212
8           4.518619           63.000000          5.883152
9           4.943637          195.900000          6.576604

   rain_accumulation_9am  rain_duration_9am  relative_humidity_9am  \
0                   0.00                0.0              42.420000
1                   0.00                0.0              24.328697
2                   0.00               20.0               8.900000
3                   0.00                0.0              12.189102
4                   8.90            14730.0              92.410000
5                   0.02              170.0              35.130000
6                   0.00                0.0              10.657422
7                   0.00                0.0              80.470000
8                   0.00                0.0              29.580000
9                   0.00                0.0              88.600000

   relative_humidity_3pm
0              36.160000
1              19.426597
2              14.460000
3              12.742547
4              76.740000
5              33.930000
6              21.385657
7              74.920000
8              24.030000
9              68.050000
```

Comenzamos a revisar la información y a limpiar información inservible, también revisamos que variables nos interesan

[8]:
```
data[data.isnull().any(axis=1)]
```

[8]:
```
     number  air_pressure_9am  air_temp_9am  avg_wind_direction_9am  \
16       16        917.890000           NaN              169.200000
111     111        915.290000     58.820000              182.600000
177     177        915.900000           NaN              183.300000
262     262        923.596607     58.380598               47.737753
277     277        920.480000     62.600000              194.400000
334     334        916.230000     75.740000              149.100000
358     358        917.440000     58.514000               55.100000
```

361	361	920.444946	65.801845	49.823346
381	381	918.480000	66.542000	90.900000
409	409	NaN	67.853833	65.880616
517	517	920.570000	53.600000	100.100000
519	519	916.250000	55.670000	176.400000
546	546	NaN	42.746000	251.100000
620	620	921.200000	56.786000	192.300000
625	625	912.400000	50.774000	171.600000
656	656	920.830000	66.344000	NaN
670	670	910.920000	48.362000	156.500000
672	672	922.448945	72.863773	NaN
705	705	911.900000	59.072000	199.800000
731	731	922.970166	51.391847	33.810942
737	737	917.895130	76.804690	104.771020
788	788	917.923442	73.249717	42.101739
840	840	918.043767	NaN	181.774042
848	848	915.250000	37.562000	246.500000
861	861	919.065408	NaN	172.303728
869	869	NaN	45.104000	259.000000
998	998	914.140000	71.240000	NaN
1031	1031	922.669195	NaN	47.946284
1035	1035	919.670000	77.576000	171.800000
1063	1063	917.300185	65.790001	NaN
1066	1066	919.564869	73.726732	68.704694

	avg_wind_speed_9am	max_wind_direction_9am	max_wind_speed_9am \
16	2.192201	196.800000	2.930391
111	15.613841	189.000000	NaN
177	4.719943	189.900000	5.346287
262	10.636273	67.145843	13.671423
277	2.751436	NaN	3.869906
334	2.751436	187.500000	4.183078
358	10.021491	NaN	12.705819
361	21.520177	61.886944	25.549112
381	3.467257	89.400000	4.406772
409	4.328594	78.570923	5.216734
517	4.697574	NaN	6.285801
519	6.666081	188.200000	NaN
546	12.929513	274.400000	17.604718
620	9.551734	201.400000	11.005745
625	NaN	181.400000	4.831790
656	15.457255	189.400000	16.486248
670	NaN	177.500000	16.128337
672	3.682370	214.196160	4.849450
705	1.275056	239.500000	1.834291
731	NaN	59.290089	11.111555
737	1.632705	97.178763	NaN

788	4.132698	64.284969	5.345258
840	0.964376	185.618601	1.570007
848	11.587349	258.700000	NaN
861	2.639600	193.058141	3.326949
869	3.265932	275.000000	4.026492
998	1.722444	232.900000	2.326418
1031	7.969686	65.770066	10.262337
1035	6.554234	191.000000	8.164831
1063	1.879553	222.498226	2.692862
1066	3.551777	102.571616	4.861315

	rain_accumulation_9am	rain_duration_9am	relative_humidity_9am \
16	0.000	0.000000	48.990000
111	0.000	0.000000	21.500000
177	0.000	0.000000	29.260000
262	0.000	NaN	17.990876
277	0.000	0.000000	52.580000
334	NaN	1480.000000	31.880000
358	0.000	0.000000	13.880000
361	NaN	40.364018	12.278715
381	NaN	0.000000	20.640000
409	0.000	0.000000	18.487385
517	4.712	14842.000000	79.880000
519	0.000	0.000000	72.550000
546	14.627	7825.000000	87.870000
620	NaN	0.000000	59.790000
625	0.000	0.000000	86.840000
656	0.000	0.000000	23.770000
670	4.970	10560.000000	80.560000
672	0.000	0.000000	16.753670
705	NaN	0.000000	77.630000
731	0.000	4.735034	34.807753
737	0.000	0.000000	13.771311
788	0.000	NaN	6.939692
840	0.000	0.000000	11.911222
848	3.171	2891.000000	91.000000
861	0.000	0.000000	12.497839
869	0.000	80.000000	85.270000
998	0.000	0.000000	24.200000
1031	0.000	0.000000	18.920805
1035	0.000	NaN	56.860000
1063	0.000	0.000000	14.972668
1066	NaN	0.000000	11.657314

	relative_humidity_3pm
16	51.190000
111	29.690000

```
177                  46.500000
262                  16.461685
277                  54.030000
334                  32.900000
358                  25.930000
361                   7.618649
381                  14.350000
409                  20.356594
517                  84.530000
519                  74.390000
546                  70.770000
620                  77.750000
625                  64.740000
656                  51.630000
670                  88.220000
672                  17.804720
705                  59.130000
731                  18.418179
737                  16.792455
788                  18.793825
840                  18.154358
848                  90.780000
861                  13.438518
869                  90.260000
998                  41.380000
1031                 19.641841
1035                 50.650000
1063                 20.966267
1066                 17.331823
```

Comenzamos borrando los valores nulos y eliminamos la columna number que no nos serviría para el ejercicio…

[9]:
```python
data = data.dropna()
del data['number']
```

[10]:
```python
data.head(20)
```

[10]:
```
    air_pressure_9am  air_temp_9am  avg_wind_direction_9am  \
0         918.060000     74.822000              271.100000
1         917.347688     71.403843              101.935179
2         923.040000     60.638000               51.000000
3         920.502751     70.138895              198.832133
4         921.160000     44.294000              277.800000
5         915.300000     78.404000              182.800000
6         915.598868     70.043304              177.875407
7         918.070000     51.710000              242.400000
8         920.080000     80.582000               40.700000
```

9	915.010000	47.498000	163.100000
10	919.650000	77.036000	70.600000
11	915.640000	45.716000	241.600000
12	917.390000	49.784000	204.100000
13	920.820000	62.438000	213.600000
14	911.000000	86.432000	202.900000
15	922.383131	70.865263	36.174175
17	916.915255	77.018961	234.539345
18	918.800000	67.082000	176.100000
19	922.040000	68.576000	58.300000
20	919.992262	62.964383	54.799094

	avg_wind_speed_9am	max_wind_direction_9am	max_wind_speed_9am	\
0	2.080354	295.400000	2.863283	
1	2.443009	140.471548	3.533324	
2	17.067852	63.700000	22.100967	
3	4.337363	211.203341	5.190045	
4	1.856660	136.500000	2.863283	
5	9.932014	189.000000	10.983375	
6	3.745587	186.606696	4.589632	
7	2.527742	271.600000	3.646212	
8	4.518619	63.000000	5.883152	
9	4.943637	195.900000	6.576604	
10	3.825167	85.500000	4.764682	
11	5.860783	265.800000	8.030615	
12	1.275056	211.800000	2.013246	
13	2.617220	165.700000	3.310671	
14	1.207948	162.900000	1.677705	
15	1.847278	58.428632	2.529142	
17	2.274725	229.474199	2.906513	
18	4.876529	183.400000	5.569981	
19	9.551734	81.900000	12.571603	
20	12.680436	74.254223	15.452306	

	rain_accumulation_9am	rain_duration_9am	relative_humidity_9am	\
0	0.00	0.0	42.420000	
1	0.00	0.0	24.328697	
2	0.00	20.0	8.900000	
3	0.00	0.0	12.189102	
4	8.90	14730.0	92.410000	
5	0.02	170.0	35.130000	
6	0.00	0.0	10.657422	
7	0.00	0.0	80.470000	
8	0.00	0.0	29.580000	
9	0.00	0.0	88.600000	
10	0.00	0.0	22.070000	
11	0.55	1770.0	90.560000	

```
12                     0.00                 0.0              73.150000
13                     0.00                 0.0              43.640000
14                     0.00                 0.0              15.190000
15                     0.00                 0.0              12.110889
17                     0.00                 0.0              21.031462
18                     0.00                 0.0              18.900000
19                     0.00                 0.0               7.540000
20                     0.00                 0.0              18.809518

    relative_humidity_3pm
0                36.160000
1                19.426597
2                14.460000
3                12.742547
4                76.740000
5                33.930000
6                21.385657
7                74.920000
8                24.030000
9                68.050000
10               32.130000
11               79.090000
12               58.430000
13               27.990000
14               24.370000
15               14.801706
17               20.755683
18               45.870000
19                7.740000
20               14.649909
```

Ahora vamos a seleccionar cuales columnas van a ser nuestros features y cuales serían nuestro target, en este caso nuestro target sería relative_humidity_3pm y el resto de columnas serían nuestros features, por otro lado, haciendo un análisis visual, vemos que las columnas rain_acumulation_9am y rain_duration_9am tiene casi todos lo valores en 0 lo que hace pensar que durante el tiempo de estudio no hubo mucha lluvia y estas características no está influyendo de manera relevante en el cálculo…

[11]:
```
features =␣
 ↪['air_temp_9am','avg_wind_direction_9am','avg_wind_speed_9am','max_wind_direction_9am','max
target = ['relative_humidity_3pm']
x = data[features]
y = data[target]
```

Vemos como quedan las features…

[12]:
```
x.head(10)
```

[12]:
```
   air_temp_9am  avg_wind_direction_9am  avg_wind_speed_9am  \
0     74.822000              271.100000            2.080354
1     71.403843              101.935179            2.443009
2     60.638000               51.000000           17.067852
3     70.138895              198.832133            4.337363
4     44.294000              277.800000            1.856660
5     78.404000              182.800000            9.932014
6     70.043304              177.875407            3.745587
7     51.710000              242.400000            2.527742
8     80.582000               40.700000            4.518619
9     47.498000              163.100000            4.943637

   max_wind_direction_9am  max_wind_speed_9am  relative_humidity_9am
0              295.400000            2.863283              42.420000
1              140.471548            3.533324              24.328697
2               63.700000           22.100967               8.900000
3              211.203341            5.190045              12.189102
4              136.500000            2.863283              92.410000
5              189.000000           10.983375              35.130000
6              186.606696            4.589632              10.657422
7              271.600000            3.646212              80.470000
8               63.000000            5.883152              29.580000
9              195.900000            6.576604              88.600000
```

Ahora el target…

[13]:
```
y.head(10)
```

[13]:
```
   relative_humidity_3pm
0              36.160000
1              19.426597
2              14.460000
3              12.742547
4              76.740000
5              33.930000
6              21.385657
7              74.920000
8              24.030000
9              68.050000
```

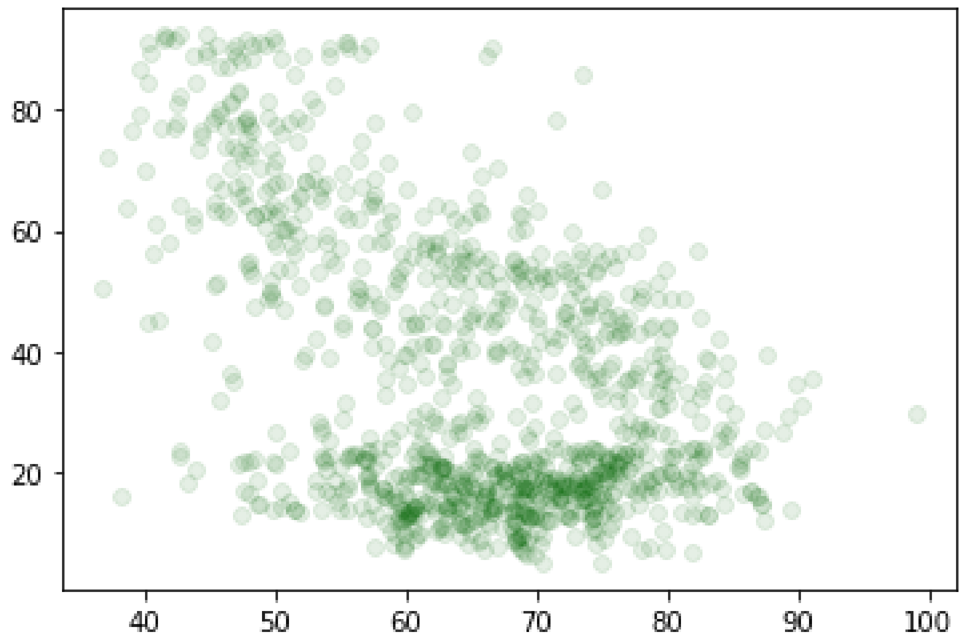
Se realiza una revisión visual de las features respecto al target con el fin de verificar si se ve alguna correlación entre ellas. Primero con la presión del aire a las 9am (este se elimina después de hacer pruebas ya que elevaba bastante el error cuadrático medio)…

[14]:
```
#plt.scatter(x['air_pressure_9am'],y,color="darkgreen", label="Data", alpha=.1)
```

Ahora con la temperatura del aire a las 9am…

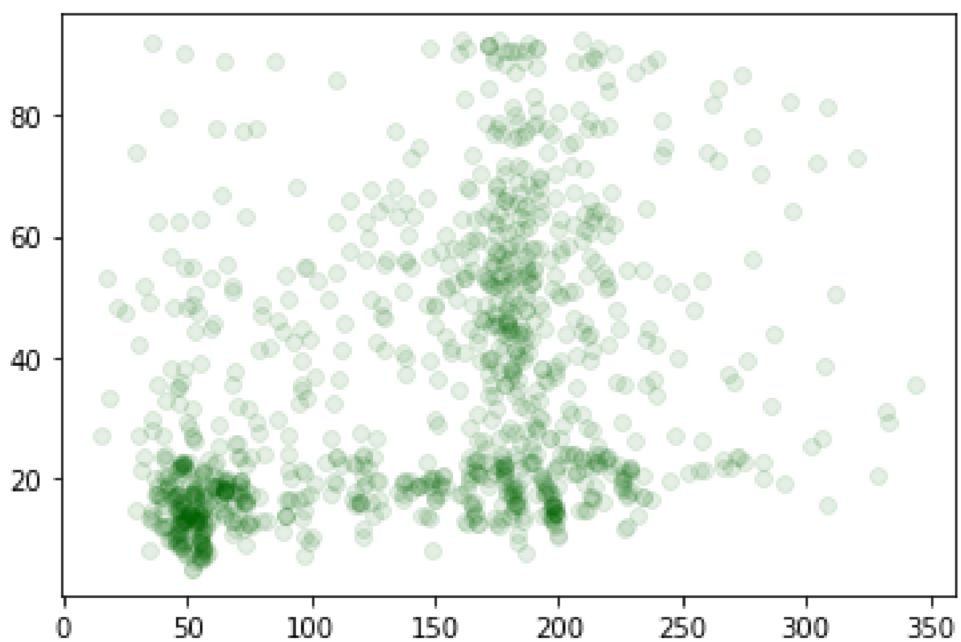
[15]:
```
plt.scatter(x['air_temp_9am'],y,color="darkgreen", label="Data", alpha=.1)
```

[15]: <matplotlib.collections.PathCollection at 0x227c93cb460>



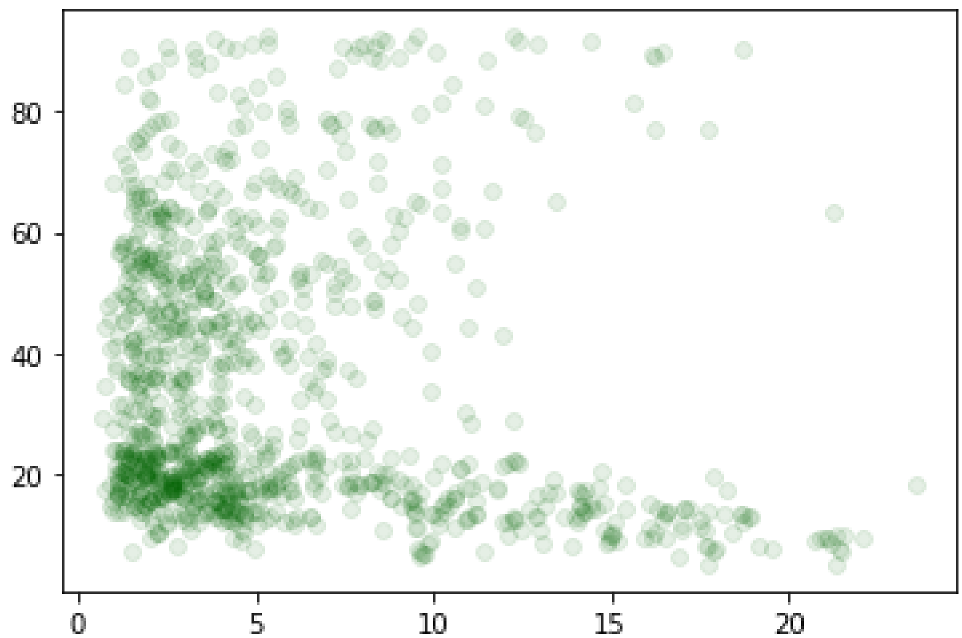
[16]:
```
plt.scatter(x['avg_wind_direction_9am'],y,color="darkgreen", label="Data",␣
 ↪alpha=.1)
```

[16]: <matplotlib.collections.PathCollection at 0x227c9466eb0>



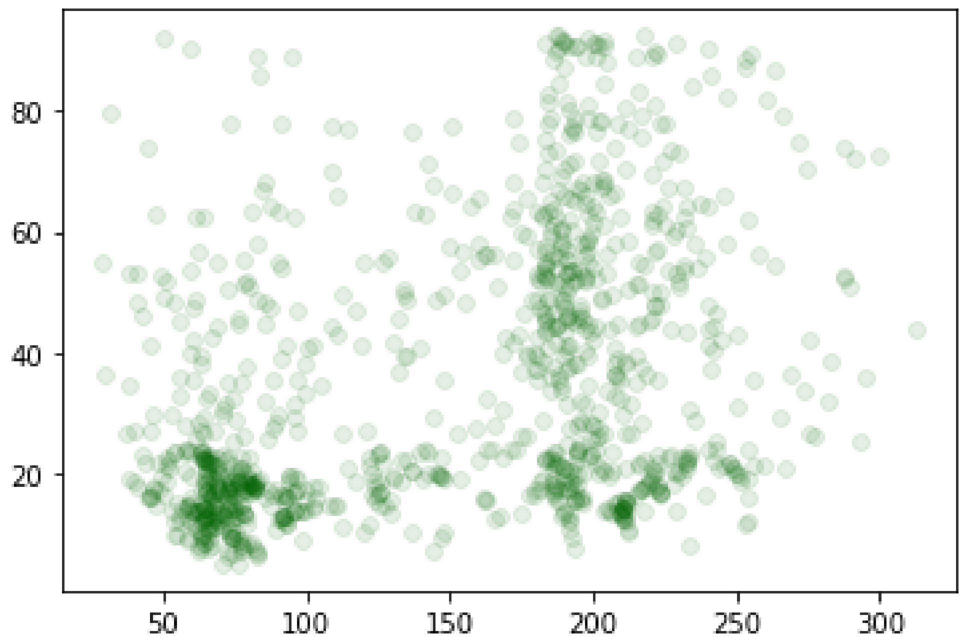
[17]:
```
plt.scatter(x['avg_wind_speed_9am'],y,color="darkgreen", label="Data", alpha=.1)
```

[17]: <matplotlib.collections.PathCollection at 0x227c94c5ee0>



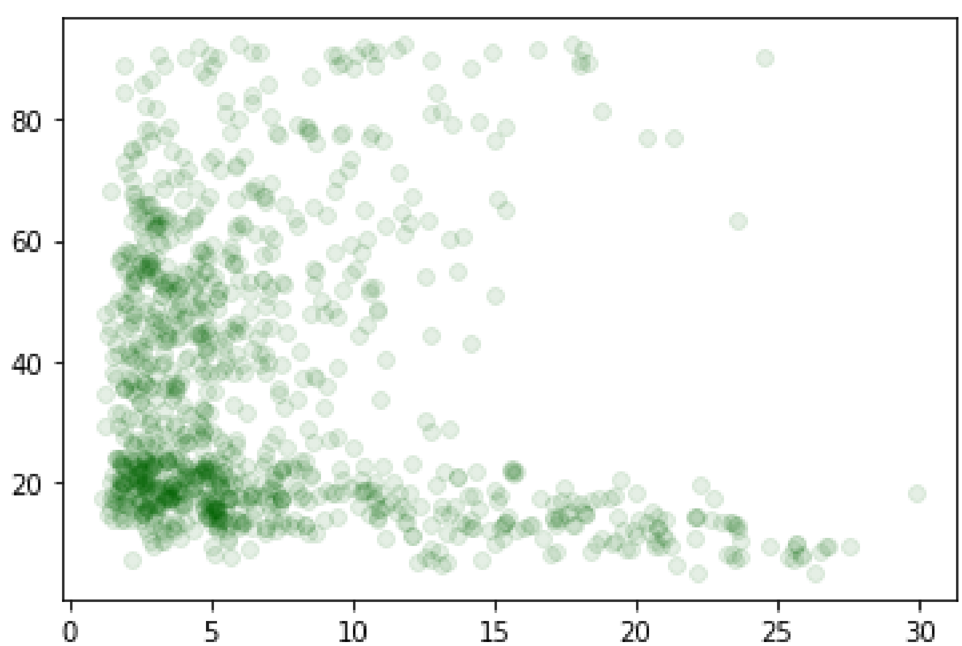
[18]:
```
plt.scatter(x['max_wind_direction_9am'],y,color="darkgreen", label="Data",␣
↪alpha=.1)
```

[18]: <matplotlib.collections.PathCollection at 0x227c9517880>



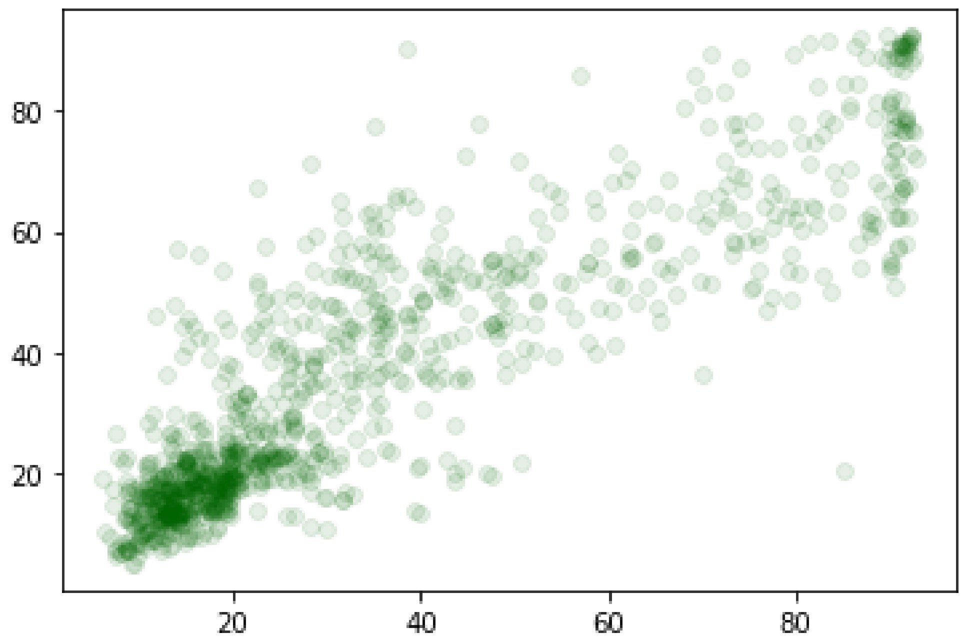

```
[19]: plt.scatter(x['max_wind_speed_9am'],y,color="darkgreen", label="Data", alpha=.1)
```

```
[19]: <matplotlib.collections.PathCollection at 0x227c9567580>
```



[20]:
```
plt.scatter(x['relative_humidity_9am'],y,color="darkgreen", label="Data",␣
↪alpha=.1)
```

[20]: <matplotlib.collections.PathCollection at 0x227c95b6760>



Separamos los datos de entrenamiento y de prueba, manejando una relación de 70% de entrenamiento y 30% para testing

[21]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30,␣
↪random_state=324)
```

Debido a que el rango de valores entre las features es un poco diferente, podemos utilizar un escalador para homogeneizar los datos

[22]:
```
scale = preprocessing.StandardScaler()
scale.fit(x_train)
x_train = scale.transform(x_train)
```

Comenzamos con el entrenamiento del modelo, al cual le pasamos los datos de entrenamiento…

[23]:
```
regressor = LinearRegression()
regressor.fit(x_train, y_train)
```

[23]: LinearRegression()

Ahora vamos al testing y le pasamos los datos de testeo…

[24]:
```
x_test=scale.transform(x_test)
y_prediction = regressor.predict(x_test)
```

Calculamos en error cuadrático medio para ver que tan precisa fue la predicción...

[25]:
```
RMSE = sqrt(mean_squared_error(y_true=y_test, y_pred = y_prediction))
regressor.score(x_test, y_test)
```

[25]: 0.814604054492224

[26]:
```
print ( RMSE )
```

9.569374524572197

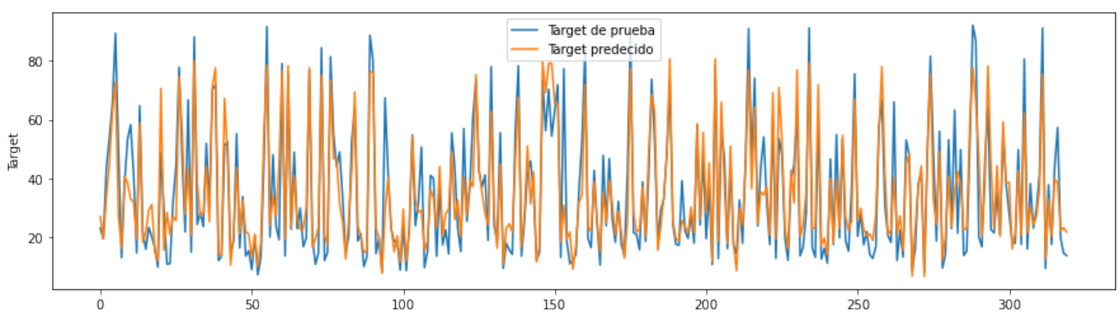
[27]:
```
y_test.shape
```

[27]: (320, 1)

[28]:
```
y_prediction.shape
```

[28]: (320, 1)

[29]:
```
n = len(y_test)
t = np.array(range(n))
plt.figure(figsize=(15, 4))
plt.plot(t, y_test, label="Target de prueba")
plt.plot(t, y_prediction, label="Target predecido")
plt.legend()
plt.xlabel("n")
plt.ylabel("Target")
plt.show()
```



[ ]: