Asignación 2

David Nogales Pérez

6 de noviembre de 2021

Problema 8: Viajando por EE.UU.

La siguiente tabla muestra las distancias (en millas) entre Baltimore y otras 12 ciudades de EE.UU., juntamente con el precio del billete de avion (en dolares) entre ellas.

Ciudad	Distancia	Tarifa
Atlanta	576	178
Boston	370	138
Chicago	612	94
Dallas	1216	278
Detroit	409	158
Denver	1502	258
Miami	946	198
NewOrleans	998	188
NewYork	189	98
Orlando	787	179
Pittsburgh	210	138
St.Louis	737	98

Tabla 1: Distancias entre ciudades y precios de billetes de avión.

(a) Plantear y resolver numéricamente el problema de predecir la Tarifa con la Distancia usando la rutina **LinearRegression()**. Hacer un gráfico con los datos y la solución obtenida.

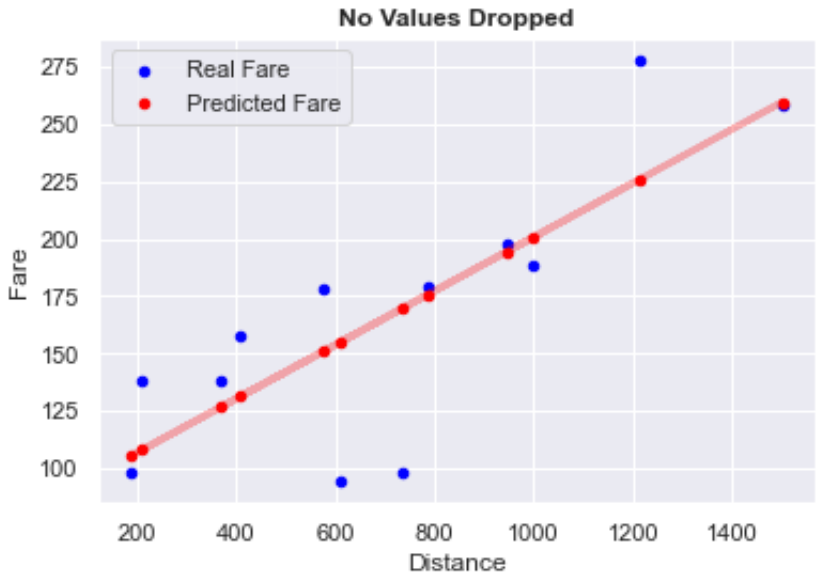

Figura 1: Gráfica de los datos y la solución obtenida (Código en Anexo I).

En la figura 1 se puede observar el resultado de entrenar el modelo de regresión lineal con los datos de la tabla 1 (en azul) y predecir los costos de los vuelos (en rojo).

(b) Observar que algunas ciudades tienen tarifas anormalmente bajas para la distancia en la que se encuentran. Diseñar una manera de reducir la influencia de estos casos y recalcular la solución.

Con el objetivo de reducir la influencia de los casos extremos[1] en nuestro conjunto de datos se plantea el siguiente procedimiento:

1. Encontrar valores, siguiendo diferentes criterios, que se encuentren en el siguiente rango de cuantiles: [0.0,0.10][0.90,1.00]
2. Eliminar/Modificar dichos valores.

Los criterios para encontrar los valores son:

- Por Tarifa.
- Por $millas/\$$.
- Por Tarifa y $millas/\$$.

Adicionalmente se plantea también modificar los valores extremos.

Eliminar valores por tarifa

Se encuentran las ciudades cuyas tarifas que se encuentran en los cuantiles fijados anteriormente y se las eliminan del conjunto de datos.

Eliminar valores por millas/$

Se encuentran las ciudades cuyo ratio $\frac{distancia}{tarifa}$ se encuentra en los cuantiles fijados anteriormente y se las eliminan del conjunto de datos.

Eliminar valores por tarifa y millas/$

Este método simplemente elimina la intersección de las ciudades que se eliminan con los 2 métodos vistos anteriormente.

Modificando tarifas

Se substituyen las tarifas que se encuentran en los cuantiles fijados anteriormente de la siguiente forma:

$$\begin{aligned} Tarifa_q &= mediaDolaresPorMilla * Distancia_q \\ mediaDolaresPorMilla &= \frac{1}{mediaMillasPorDollar} \\ mediaMillasPorDollar &= \frac{1}{n}\sum_{i=1}^{n}\frac{distancia_i}{tarifa_i} \end{aligned}$$

[1] En el Notebook adjunto se incluye también un estudio adicional de detección de outliers que no se incluye en este documento.

Resultados

Los resultados de los métodos escritos anteriormente se pueden observar numéricamente en la tabla 2 y gráficamente en la figura 2.
El método que obtiene menos error (MSE y MAE) y que tiene el R^2 mas alto es el que solo toma en cuenta eliminar los valores extremos de Tarifa (Drop by Fare). Aunque cabe remarcar que mediante este criterio eliminamos casi la mitad del conjunto de datos.

	MSE	MAE	R2	Dropped Cities	Coeff	Intercept
No Values Dropped	1192.4	25.72	0.63	None	0.12	83.27
Drop by Fare	62.39	6.56	0.87	Chicago, NewYork, St.Louis, Dallas, Denver	0.07	123.09
Drop by Miles per Dollar	360.04	15.46	0.82	NewYork, Pittsburgh, Chicago, St.Louis	0.11	103.56
Drop by Fare and Miles per Dollar	329.66	14.45	0.84	St.Louis, NewYork, Chicago	0.11	108.29
Input Fare	867.55	27.0	0.84	None	0.18	55.76

Tabla 2: Resultados obtenidos mediante diferentes métodos (Código en Anexo I).

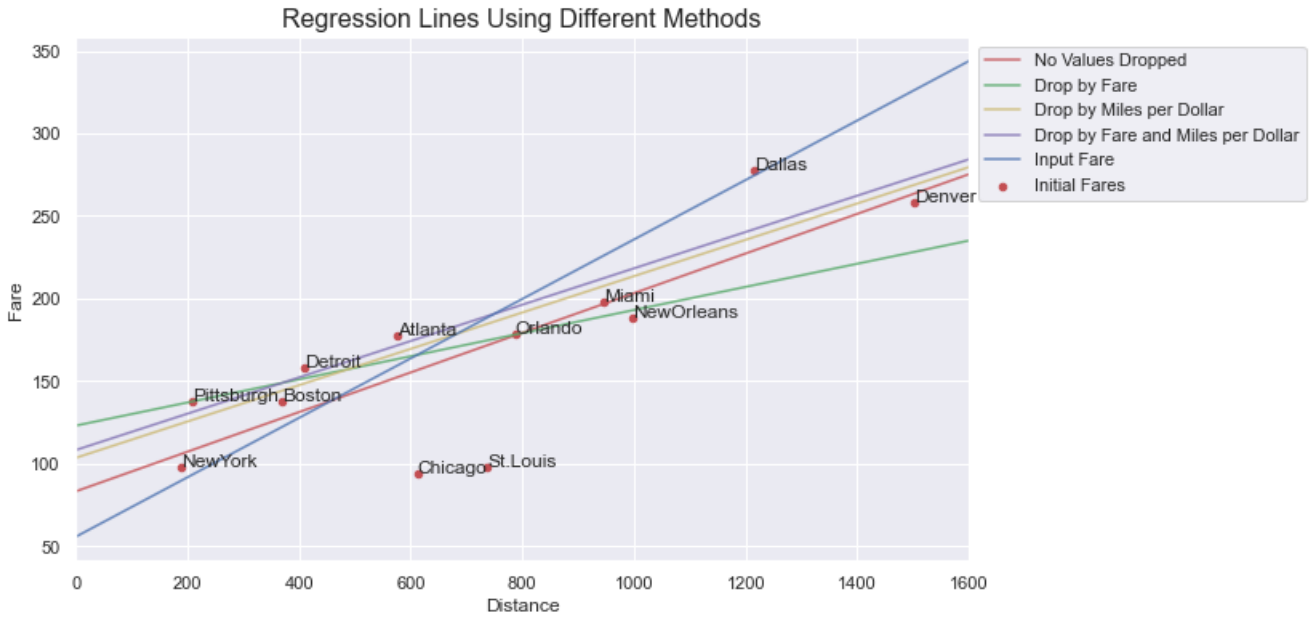

Figura 2: Comparación de rectas de regresión obtenidas por medio de los diferentes métodos.

Observando los resultados podemos concluir que con cualquier método propuesto anteriormente rebajamos el impacto de los valores extremos en nuestra regresión.

Cabe aclarar que por ejemplo el método de modificar los valores puede que no sea adecuado ya que para alguna distancia dada puede que el modelo genere una tarifa excesivamente alta.

Anexo I

Por simplicidad solo se adjunta el código utilizado para entrenar el modelo, generar la gráfica de predicción y guardar los resultados. El resto del código puede ser visto en el Notebook adjunto.

```
def train_and_get_results(new_data_df, dropped_cities, results, method, ax=None,
                          print_info=False, line_color='red', draw_points=False):
    #Training model
    X = new_data_df.Distance.array.reshape(-1,1)
    y = new_data_df.Fare.array.reshape(-1,1)
    regressor_linear = LinearRegression()
    regressor_linear.fit(X,y)
    y_predicted = regressor_linear.predict(X)
    y_predicted = y_predicted.reshape(1,-1)[0]
    new_data_df["Prediction"] = y_predicted
    #For plotting prediction
    if ax!=None:
        if draw_points:
            new_data_df.plot(kind="scatter",x="Distance",y="Fare",c="blue",ax=ax)
            new_data_df.plot(kind="scatter",x="Distance",y="Prediction",ylabel="Fare",
                             ax=ax,c=line_color)
        ax.legend(['Real Fare','Predicted Fare'])
        ax.plot(X, y_predicted, color=line_color, linewidth=3,alpha=0.3)
        ax.set_title(method,fontweight ="bold")
    #Getting different metrics
    mse = mean_squared_error(y, y_predicted)
    mae = mean_absolute_error(y, y_predicted)
    r2 = r2_score(y, y_predicted)
    if(print_info):
        print('----------------',method,'----------------')
        print("Mean squared error: %.2f" % mse)
        print("Mean absolute error: %.2f" % mae)
        print("Coefficient of determination: %.2f" % r2)
        print('----------------------------------------')
    #Saving results
    results[method]= {  'MSE':round(mse,2),'MAE':round(mae,2),'R2':round(r2,2),
                        'Dropped Cities':dropped_cities,
                        'Coeff':round(regressor_linear.coef_[0][0],2),
                        'Intercept':round(regressor_linear.intercept_[0],2)}
```