Sebastian Bocul

1.Raport z analizy i modelowania zestawu danych Boston Housing

2. Wstęp:  
Celem ćwiczenia jest analiza i modelowanie zestawu danych Boston Housing, który zawiera informacje różnych domów w mieście Boston. Moją motywacją była ciekawość oraz zainteresowanie, w jaki sposób możemy wykorzystać regresje liniową w świecie biznesu.

3. Opis danych:

Zbiór danych zawiera 506 wierszy oraz 14 kolumn:

crim - wskaźnik przestępczości na mieszkańca według miasta.

zn - część gruntów mieszkalnych przeznaczonych na działki o powierzchni ponad

25 000 m².

indus - udział niehandlowych akrów na miasto.

chas - zmienna fikcyjna Charles River.

nox - stężenie tlenków azotu.

rm - średnia liczba pokoi na mieszkanie.

age - odsetek jednostek zajmowanych przez właścicieli wybudowanych przed 1940 r.

dis - średnia ważona odległości do pięciu bostońskich centrów zatrudnienia.

rad - wskaźnik dostępności do autostrad radialnych.

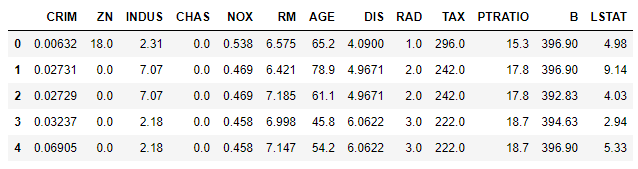
tax - pełna stawka podatku od nieruchomości za 10 000 USD.

ptratio - stosunek uczniów do nauczycieli według miasta.

b - 1000 (Bk - 0,63) ^ 2, gdzie Bk to odsetek czarnych według miasta.

lstat - niższy status populacji (procent).

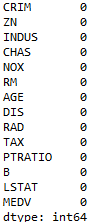
medv - średnia wartość domów zajmowanych przez właściciela w tysiącach dolarów.



Rysunek przedstawiający fragment danych. Brak kolumny MEDV - kolumna ta zostaje stworzona przez nas.

4.Opis procesu przygotowywania danych do analizy:  
Przed rozpoczęciem procesu analizy danych należy sprawdzić, czy dane są poprawne. W naszym przypadku sprawdzamy, czy nasz zbiór nie zawiera brakujących danych, komendą:

print(boston.isnull().sum())

wynik programu:  


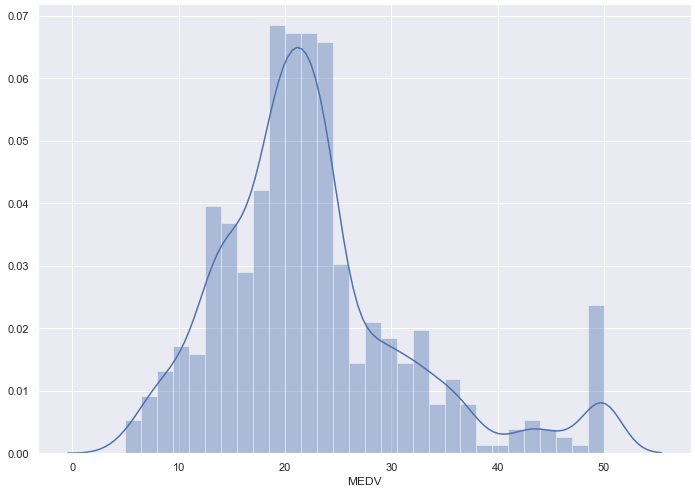
5. Analiza danych:  
Analiza danych została przeprowadzona na podstawie wykresu oraz koleracji danych.

Wykres Kod:  
sns.set(rc={'figure.figsize':(11.7,8.27)})

sns.distplot(boston['MEDV'], bins=30)

plt.show()

Po wykonaniu powyższych instrukcji zostanie ukazany nam wykres przedstawiający rozkład MEDV.

Jak widzimy wartości niektórych słupków odbiegają od ogólnego rozkładu danych.

Koleracja danych kod:

correlation\_matrix = boston.corr().round(2)

sns.heatmap(data=correlation\_matrix, annot=True

Wyniki koleracji zostaną zaokrąglone do 2 miejsc po przecinku.



Koleracja danych

Z powyżej przedstawionego obrazu możemy wywnioskować które wartości najbardziej oddziałują na MEDV oraz najmniej. RM ma wartość 0,7 co oznacza, że jest duża pozytywną zależność między RM a MEDV. LSTAT – oddziałuje najbardziej negatywnie (-0,74).

Przyjrzyjmy się wartością RM oraz LSTAT z bliska:

plt.figure(figsize=(20, 5))

features = ['LSTAT', 'RM']

target = boston['MEDV']

for i, col in enumerate(features):

plt.subplot(1, len(features) , i+1)

x = boston[col]

y = target

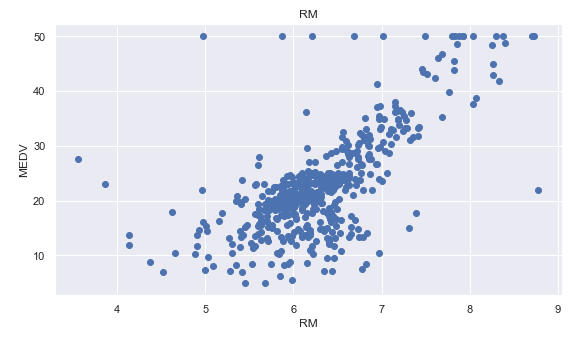
plt.scatter(x, y, marker='o')

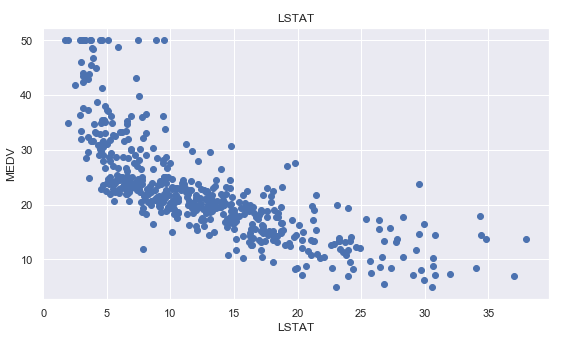
plt.title(col)

plt.xlabel(col)

plt.ylabel('MEDV')

Po wykonaniu powyższego kodu dostaniemy wykresy zależności (RM - MEDV, LSTAT - MEDV).





Wnioski:

Jak widzimy z wykresu zależności MEDV – RM możemy wywnioskować iż cena mieszkania idzie zależnie od ilości pokojów, które posiada te mieszkanie(więcej pokoi = większa cena).

Drugi wykres MEDV-LSTAT(niższy status populacji (procent)) mówi nam o tym iż czym niższy status populacji tym cena mieszkania też jest niższa.

6. Modelowanie danych

Do modelacji naszych danych zastosujemy regresje liniowa

Wykorzystamy do tego metodę najmniejszych kwadratów, która jest najstarszą i najłatwiejszą metodą jednak posiada wady(np. Małą odporność na dane odstające) oraz RMSE(Root Mean Squared Error)czyli średniej kwadratowej błędów.

Kod:

Przygotowanie danych:

lin\_model = LinearRegression()

lin\_model.fit(X\_train, Y\_train)

Modelowanie danych:

y\_train\_predict = lin\_model.predict(X\_train)

rmse = (np.sqrt(mean\_squared\_error(Y\_train, y\_train\_predict)))

r2 = r2\_score(Y\_train, y\_train\_predict)

print("Wydajnosc modelu dla danych trenujacych")

print("--------------------------------------")

print('RMSE is {}'.format(rmse))

print('R2 score is {}'.format(r2))

print("\n")

y\_test\_predict = lin\_model.predict(X\_test)

rmse = (np.sqrt(mean\_squared\_error(Y\_test, y\_test\_predict)))

r2 = r2\_score(Y\_test, y\_test\_predict)

print("Wydajnosc modelu dla danych testowych")

print("--------------------------------------")

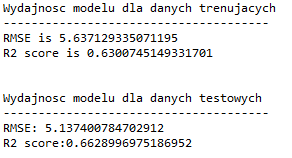
print('RMSE: {}'.format(rmse))

print('R2 score:{}'.format(r2))

plt.scatter(Y\_test, y\_test\_predict)

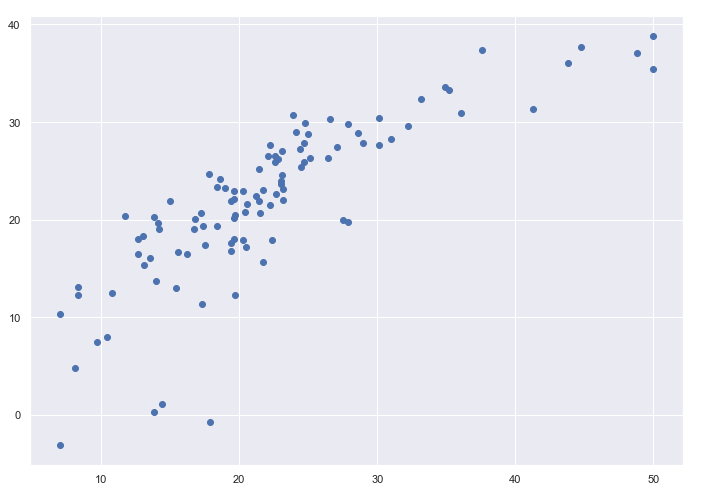
plt.show()

Wynik kodu:



Nasz zbiór został podzielony na treningowy i testowy.

A następnie została sprawdzona wydajność obydwu zbiorów.



Wynikiem naszej regresji liniowej jest wykres(idealna regresja liniowa powinna przedstawiać linię prostą) przestawiający przewidzianą wartość cenową dla poszczególnych domów.

7. Wnioski końcowe:  
Jak widzimy na powyższym przykładzie, analiza i modelowanie możemy zastosować w wielu przykładach życia codziennego nie tylko w fizyce i matematyce. Dla osób zaczynających przygodę w tym temacie może być na początku niezrozumiałe i zagmatwane jednak po krótkim czasie wszystko staje sie przejrzyste i logiczne.