

Класификация и Логистична регресия

Д-р инж. Огнян Кабранов
Дипл. инж. Даниела Цветкова

2019 година

Здравейте отново !!!!

Ще говорим аз основните проблеми, които машинното обучение решава

Регресия - предсказване стойностите на непрекъснатата функция. Ще разгледаме **линейна регресия**.

Класификация - предсказване принадлежност към клас. Ще разгледаме **логистична регресия**.

Линейна регресия

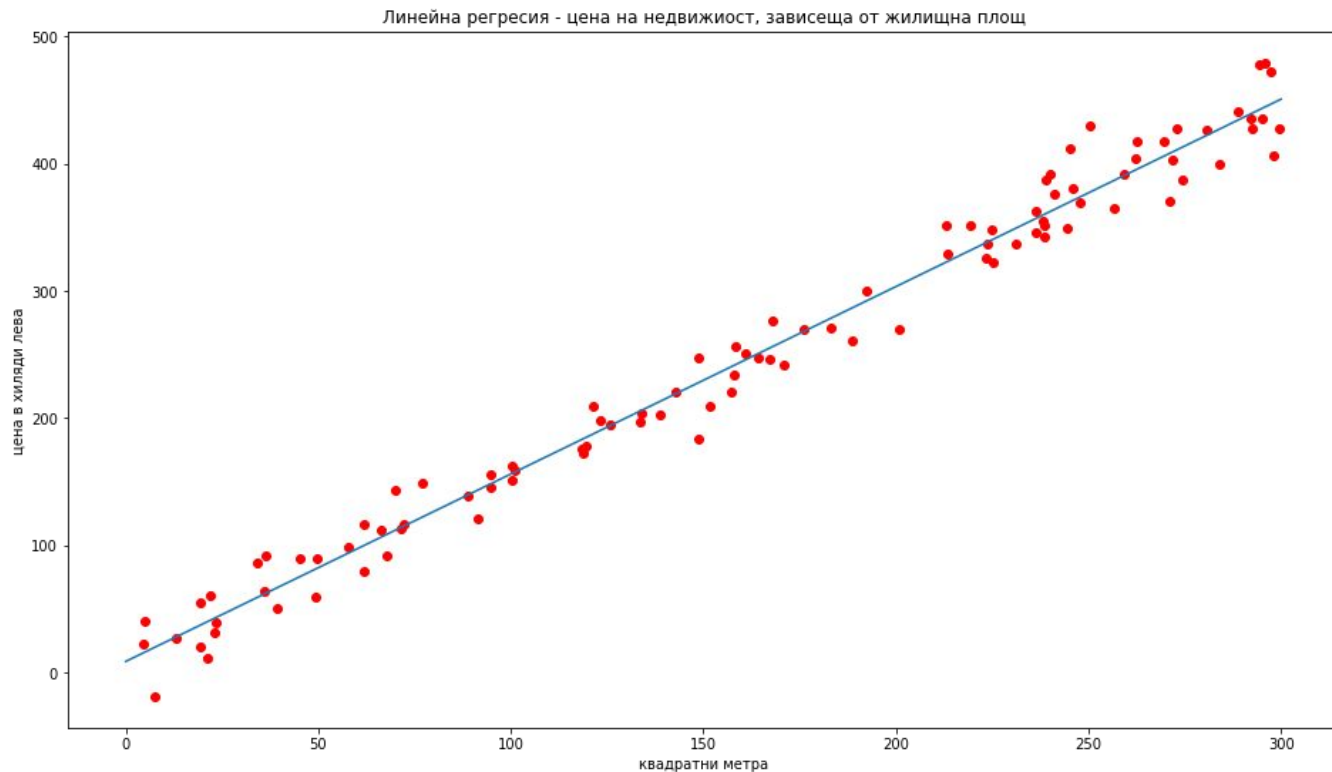
Моделиране на наблюдения с линейна функция

Пример: цена на недвижимост, в зависимост от жилищна площ.

Нека приемем че имаме 100 набора от данни (площ и цена).

Искаме да обучим нашата система да предсказва цената.

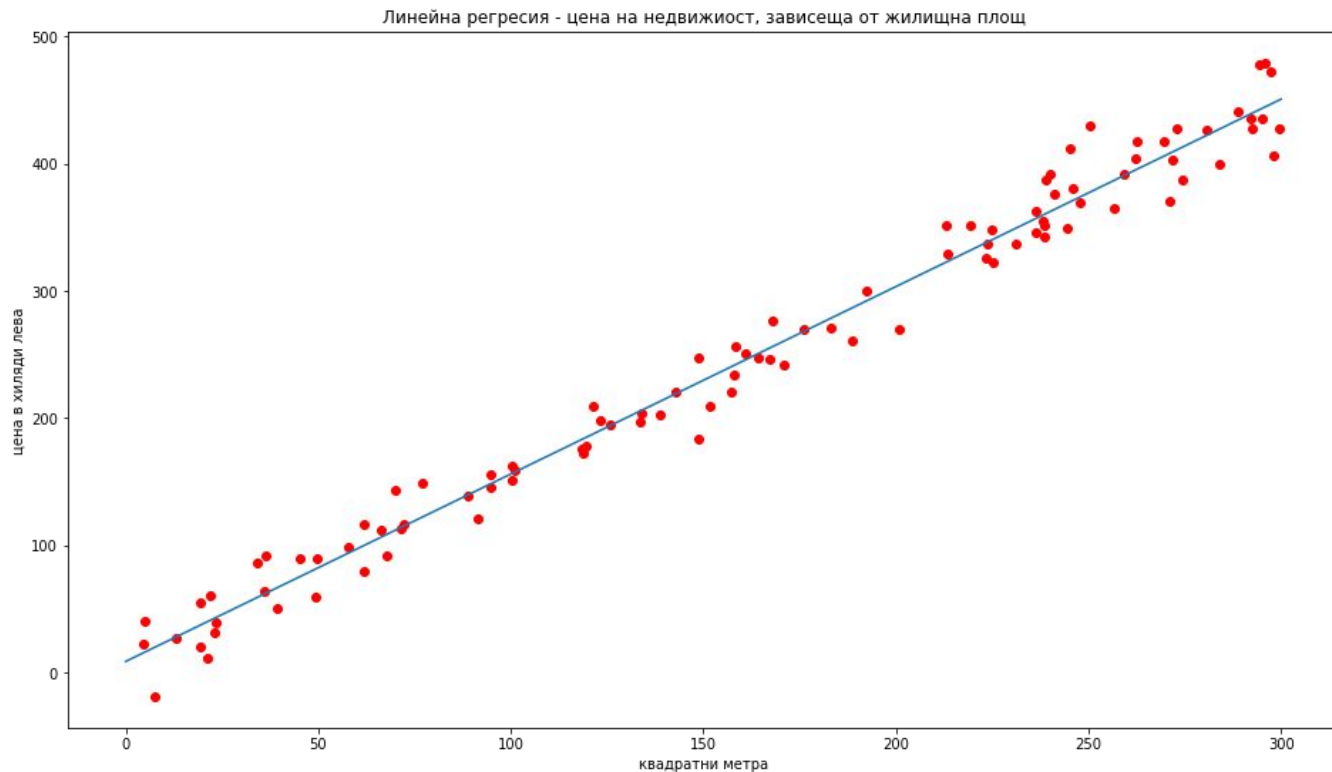
Линейна регресия



Всички
наблюдения са
обобщени от
линейна функция.

$$y = a \cdot x + b$$

Линейна регресия



Или с други думи
нашият
предиктивен
модел за пазарни
цени на
недвижимост е
линейна
регресия

Линейна регресия и Python

- Няма да навлизаме в математическите детайли на линейната регресия, това ще бъде изучавано в университета.
- Само ще спомена, че регресията може да бъде в повече от едно измерение, например площ, брой спални, възраст на имота итн.
- Тук ще представим малък пример, който ще илюстрира колко е лесно да се използва.

Линейна регресия и Python

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

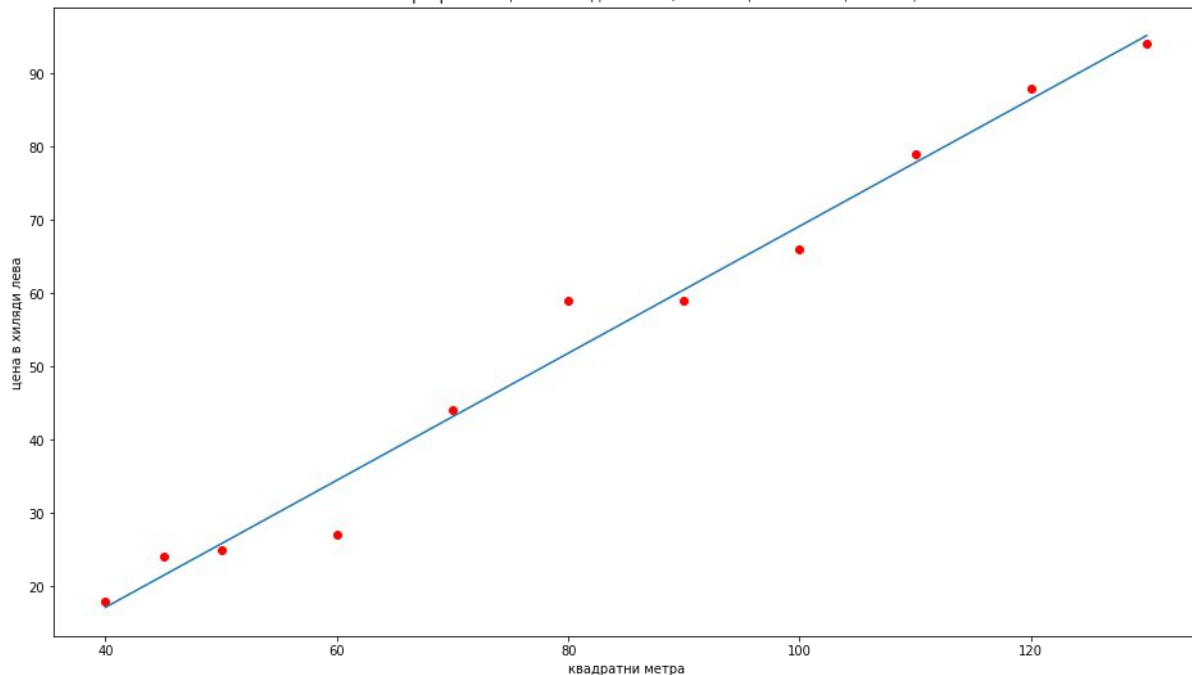
x = np.array([[40], [45], [50], [60], [70], [80], [90], [100], [110], [120], [130]])
y = np.array([[15], [20], [25], [27], [42], [54], [59], [70], [79], [88], [94]])

model = LinearRegression()
model.fit(np.asarray(x), np.asarray(y))

x_new = np.linspace(40, 130, 100)
y_new = model.predict(x_new[:, np.newaxis])
plt.figure(figsize=(16, 9))
ax = plt.axes()
ax.scatter(x, y, color='r')
ax.plot(x_new, y_new)
plt.show()
```


Линейна регресия и Python

Линейна регресия - цена на недвижност, зависеща от жилищна площ



Всичко това може
да се случи не
само в едно, но и
в много
измерения

Логическа регресия

Апроксимация на наблюдения с отговор ДА или НЕ (логически отговор).

Пример: Колко часа трябва да учим да издържим изпита по информатика ?

Задание

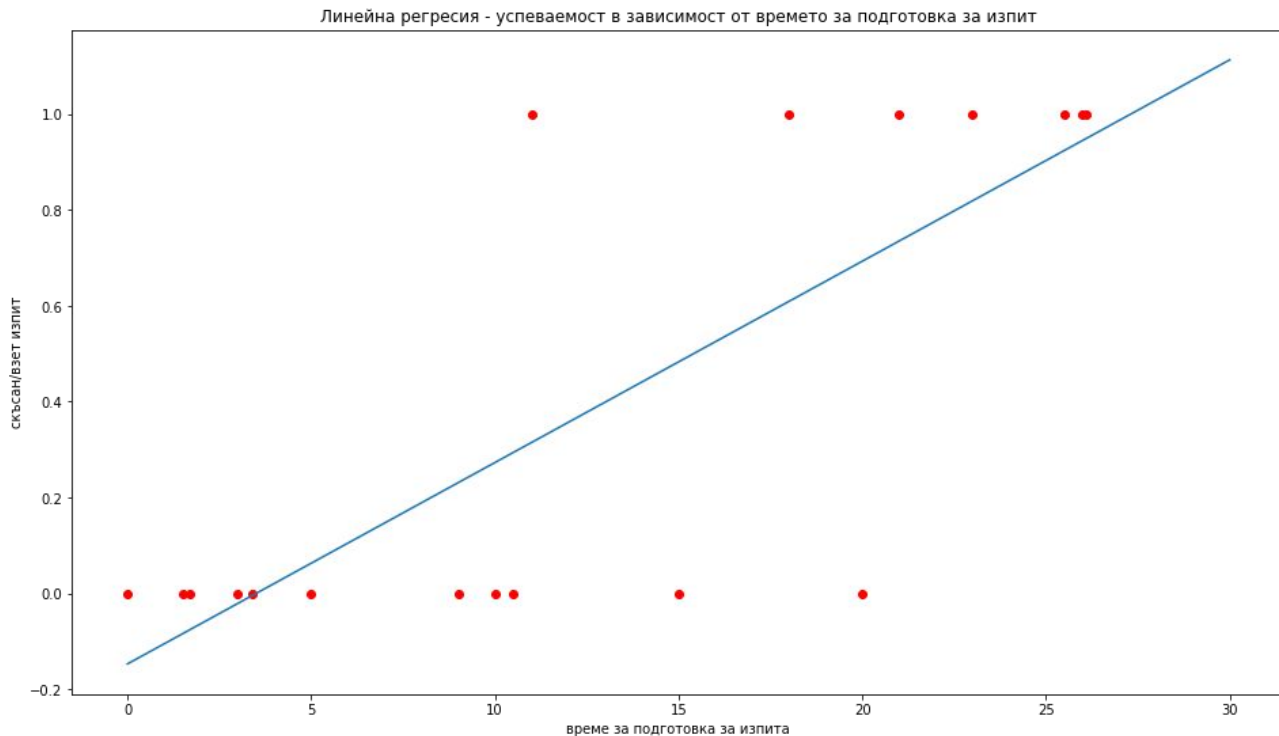
Наблюдения върху студенти,

Часове за подготовка: 0.0, 1.5, 1.7, 3.0, 3.4, 5.0, 9.0, 10.0, 10.5, 11.0, 15.0, 18.0, 20.0, 21.0, 23.0, 25.5, 26.0, 26.1

успеваемост: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1

Пример: Колко часа трябва да учим да издържим изпита по информатика ?

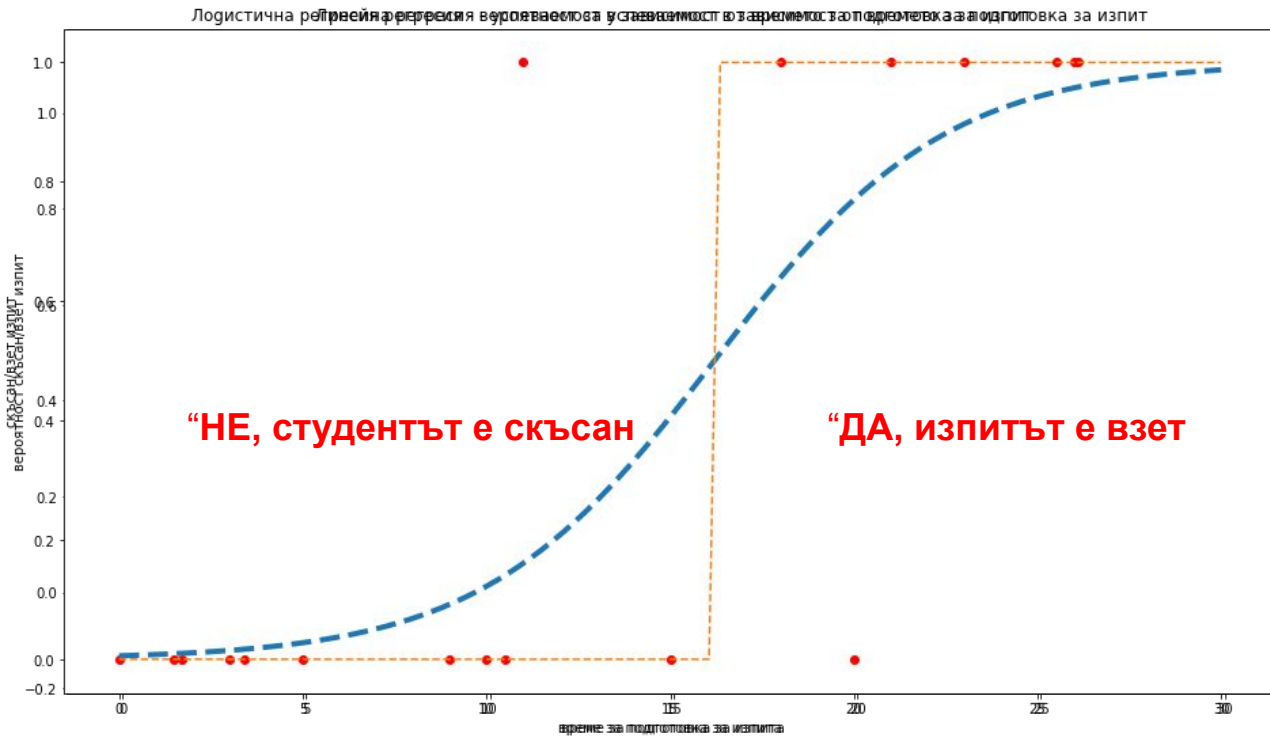
Модел с линейна регресия



**Този
предиктивен
модел не е
подходящ.**

$$y = a.x + b$$

Модел с логистична регресия



Сигмоидна функция
е много по акуратно
моделира
действителността :

$$y = a.x + b$$

$$p = \frac{1}{1 + e^{-y}}$$

```
# create a logistic regression model
logisticModel = LogisticRegression(C=1e5)
logisticModel.fit(np.asarray(x), np.asarray(y))

# predict y from the data
x_new = np.linspace(0, 30, 100)
y_new = logisticModel.predict_proba(x_new[:, np.newaxis])

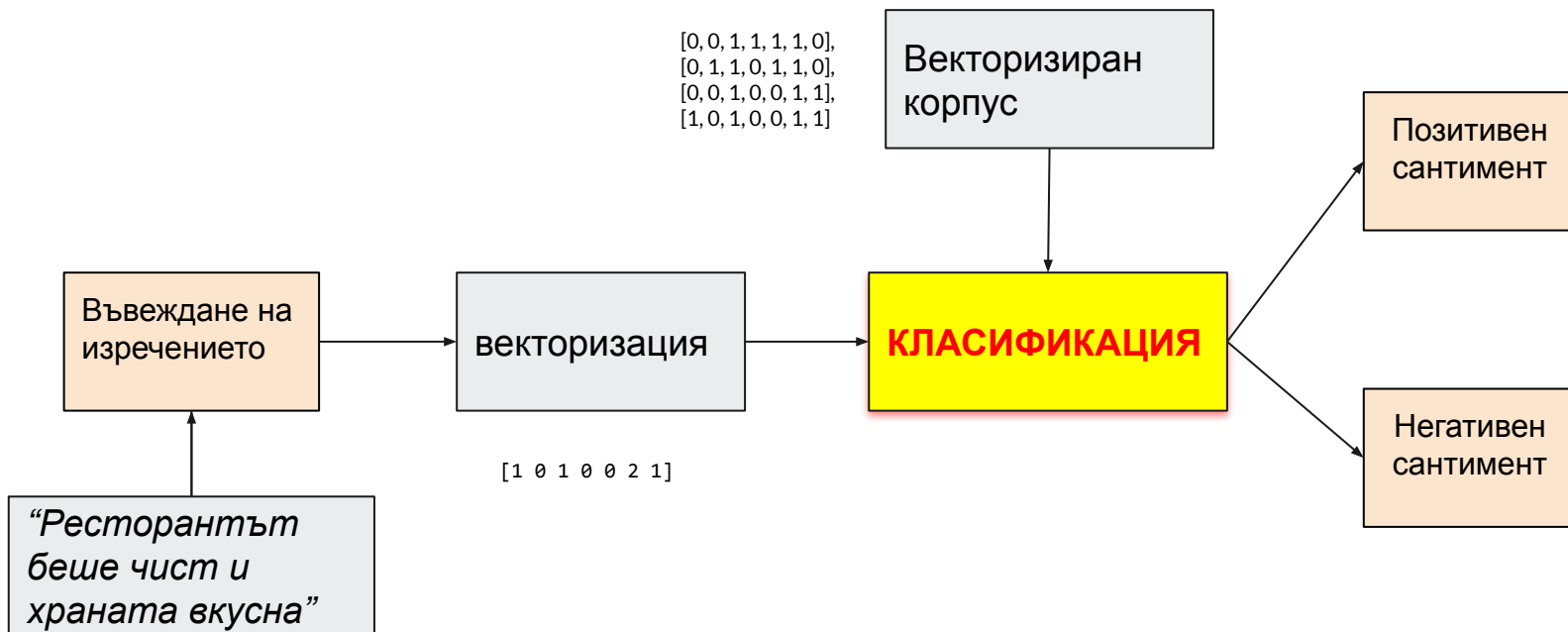
# plot the results
plt.figure(figsize=(16, 9))
ax = plt.axes()
ax.scatter(x, y, color='r')
ax.plot(x_new, y_new[:, 1], '--', linewidth=4)
ax.plot(x_new, logisticModel.predict(x_new[:, np.newaxis]), '--')
ax.set_xlabel('време за подготовка за изпита')
ax.set_ylabel('вероятност скъсан/взет изпит')
plt.title('Логистична регресия регресия - вероятност за успеваемост в зависимост  
от времето за подготовка за изпит')
#ax.axis('tight')

plt.show()
```

Следващия път

Ще говорим как да използваме всичко това за класификация когато анализираме на сантименти.

Да дефинираме проблема за разпознаване на насторения/сентименти



БЛАГОДАРЯ И ДО НОВИ СРЕЩИ ?

Thank you! Danke ! Merci !



Литература

Линейна регресия :

<https://debian.fmi.uni-sofia.bg/~boev/pages/ca039040-a56c-102c-87f2-000461a13bbe/files/04-linreg.pdf>

Логистична регресия :

<http://ufldl.stanford.edu/tutorial/supervised/LogisticRegression/>

Много добър източник от университета във Станфорд, Калифорния

<http://cs229.stanford.edu/notes/cs229-notes1.pdf>