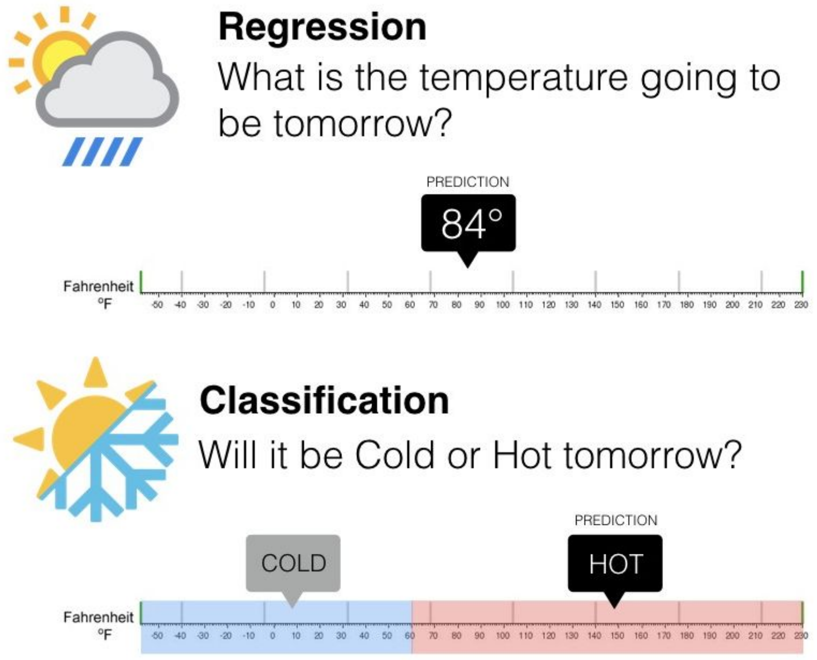
CS410: Principles and Techniques of Data Science

Module 9: Linear Models

https://drive.google.com/drive/folders/1vc2ZGZ_Pe1UQsnE_9kUtGL1fRwfq6eui?usp=sharing

Introduction

The most common supervised learning tasks are regression (predicting values) and classification (predicting classes).



Introduction

In this module we will look at the Linear Regression model

We will discuss two very different ways to train it:

1. Using a direct “closed-form” equation
2. Using an iterative optimization approach called Gradient Descent (GD)

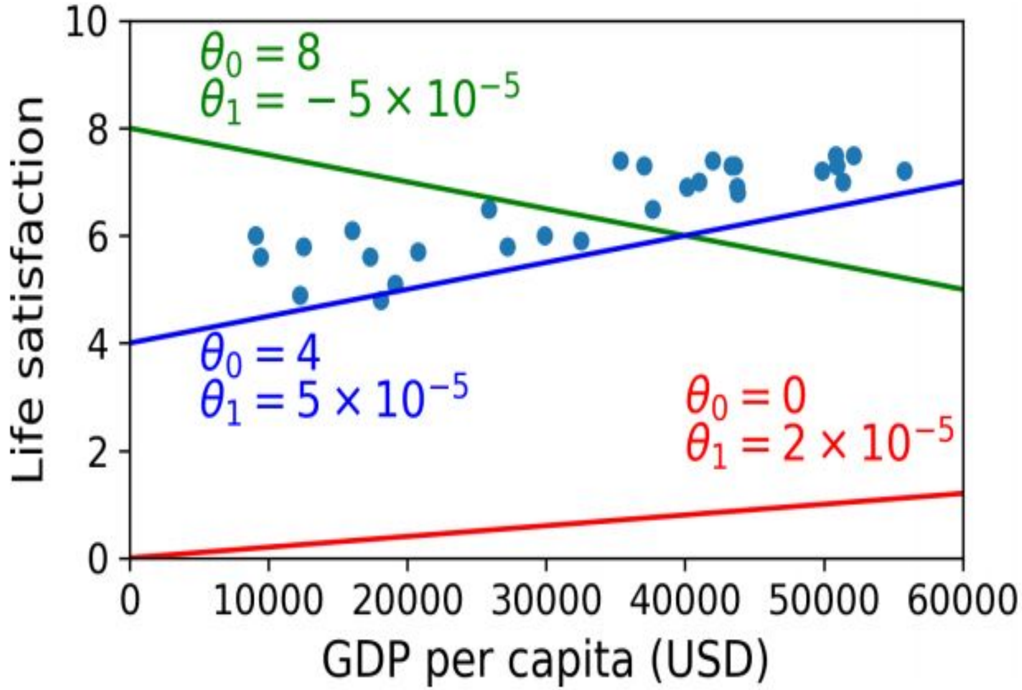
Polynomial Regression is a more complex model that can fit nonlinear datasets

Commonly used models for classification: Logistic Regression.

Linear Regression

Simple regression model of life satisfaction

GDP (x1)	LS (y)
10000	5.8
11000	5.9
30000	6
50000	7.5
60000	7.6
55000	?



$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots\ldots\ldots + \theta_n x_n$$

In this equation: $\hat{y}$ is the predicted value

n is the number of features

x_i is the ith feature value.

θ_j is the jth model parameter (including the bias term θ_0 and the feature weights $\theta_1, \theta_2, \cdots, \theta n$).

Linear Regression Training

Training a model = setting its parameters so that the model best fits the training set.

Most common performance measure of a regression model is the Mean Square Error (RMSE) => find the value of θ that minimizes the MSE.

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots\ldots\ldots + \theta_n x_n$$

$$MSE(X, h_\theta) = \frac{1}{m} \sum_{i=1}^{m} (\theta^T x^{(i)} - y^{(i)})^2$$

The Normal Equation

To find the value of θ that minimizes the cost function, there is a closed-form solution
= a mathematical equation that gives the result directly => Normal Equation

$$\hat{\theta} = (X^T X)^{-1} X^T y$$

In this equation:

$\hat{\theta}$ is the value of θ that minimizes the cost function.

y is the vector of target values containing $y^{(1)}$ to $y^{(m)}$.

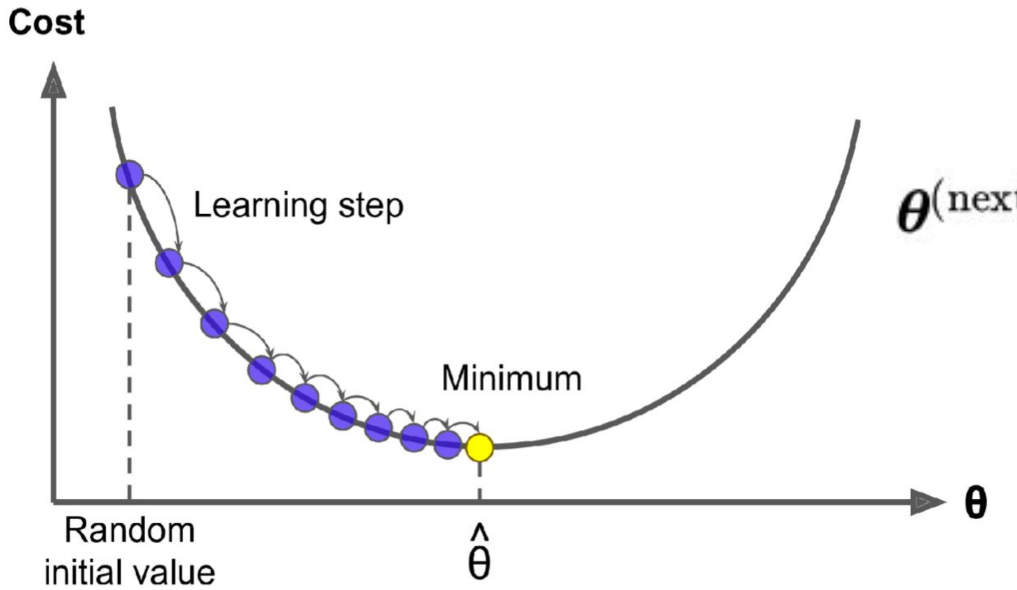
To program:

1. Use NumPy package and use the inv( ) function from NumPy linear algebra module (np.linalg)
2. Use Scikit-learn linear regression

Gradient Descent

An optimization algorithm capable of finding optimal solutions to a wide range of problems.
Tweak parameters iteratively in order to minimize a cost function.

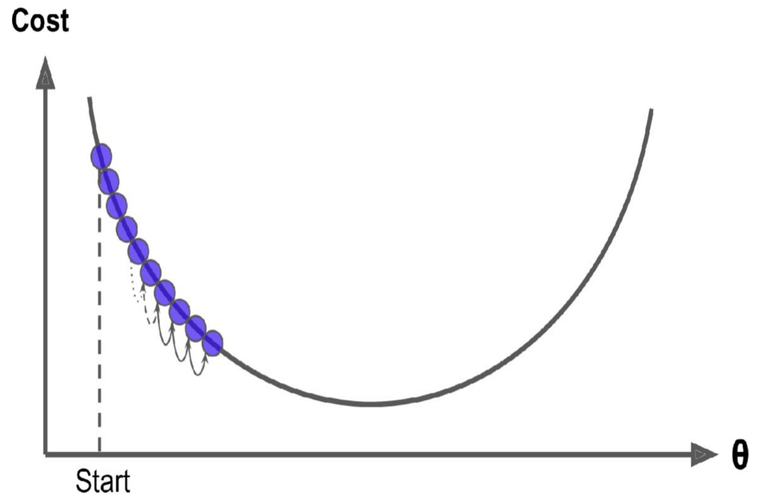
You start by filling θ with random values (this is called random initialization).
Then you improve it gradually, taking one baby step at a time, each step attempting to decrease the cost function (e.g., the MSE), until the algorithm converges to a minimum



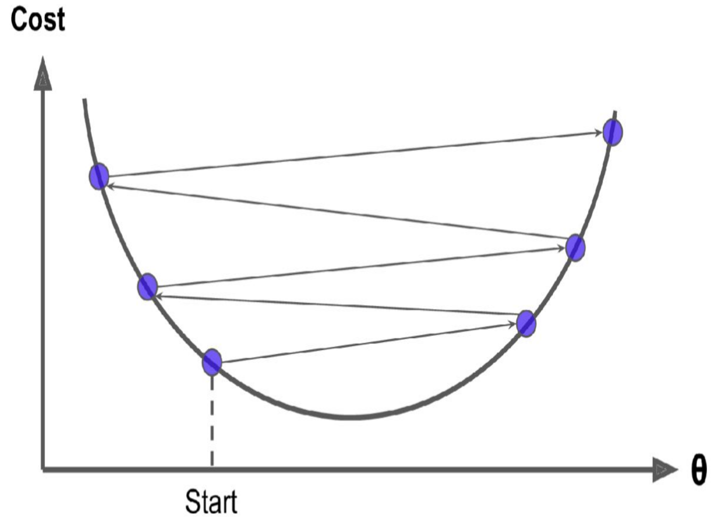
$$\boldsymbol{\theta}^{(\text{next step})} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \, \text{MSE}(\boldsymbol{\theta})$$

Gradient Descent: Learning Rate

Learning rate is too small=> the algorithm will have to go through many iterations to converge, which will take a long time



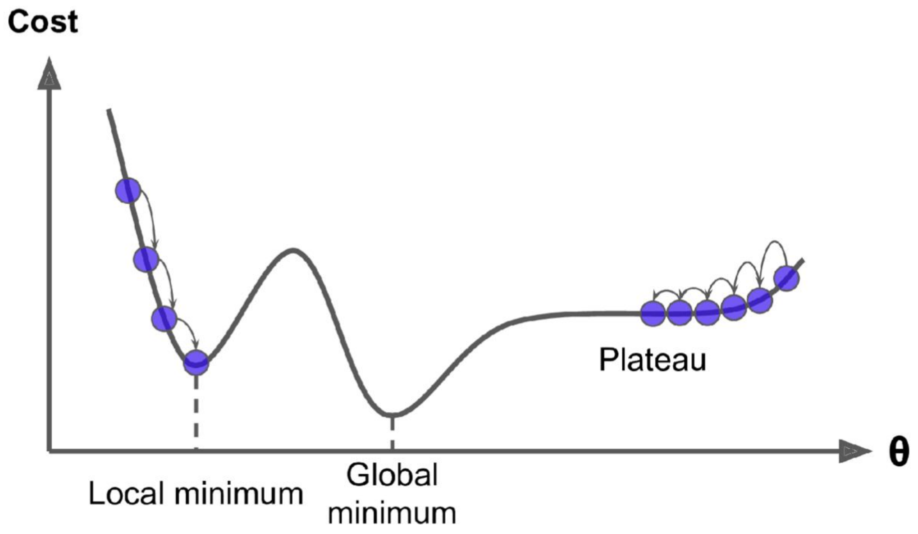
Learning rate is too high => you might jump across the valley and end up on the other side missing the minima



Challenges of Gradient Descent

Two main challenges with Gradient Descent:

1. If the random initialization starts the algorithm on the left, then it will converge to a local minimum, which is not as good as the global minimum.
2. If it starts on the right, then it will take a very long time to cross the plateau. And if you stop too early, you will never reach the global minimum.

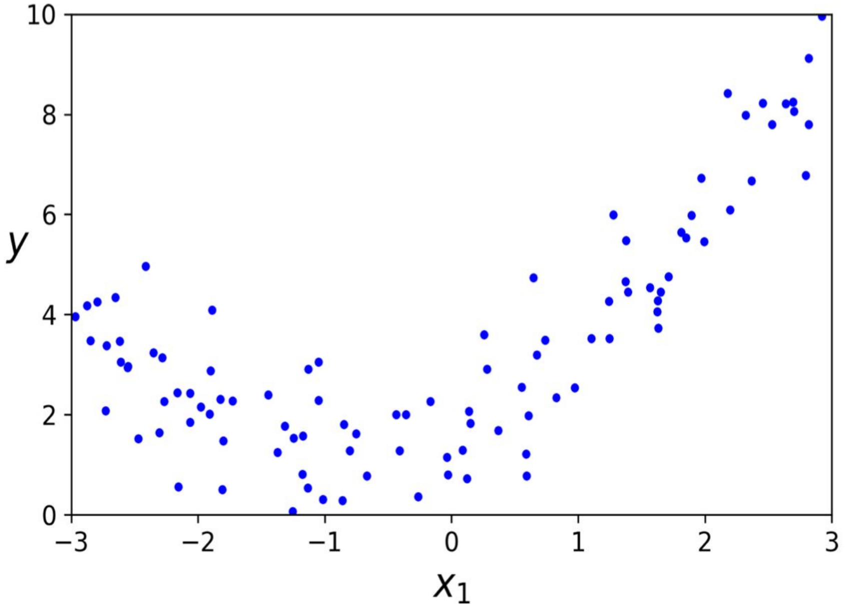


Polynomial Regression

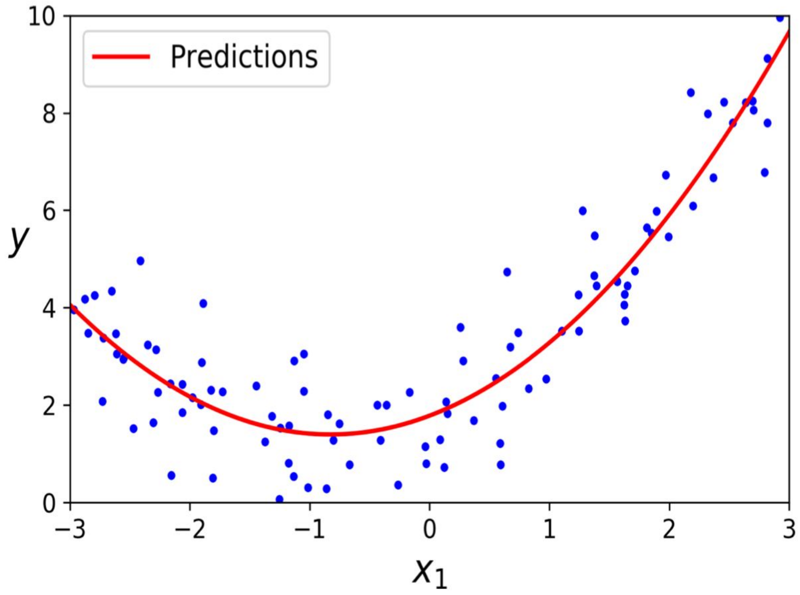
What if your data is more complex than a straight line?

-> Add powers of each feature as new features

A straight line will never fit this data properly



Adding the square (second-degree polynomial) of each feature in the training set as a new feature and then fitting a linear regression to this extended training set



Logistic Regression

Logistic regression => ML algorithm for classification.

Commonly used to estimate the probability that an instance belongs to a particular class (e.g., what is the probability that this email is spam?).

If the estimated probability is greater than 50% => positive class (labeled “1”)

otherwise it predicts negative class (labeled “0”).

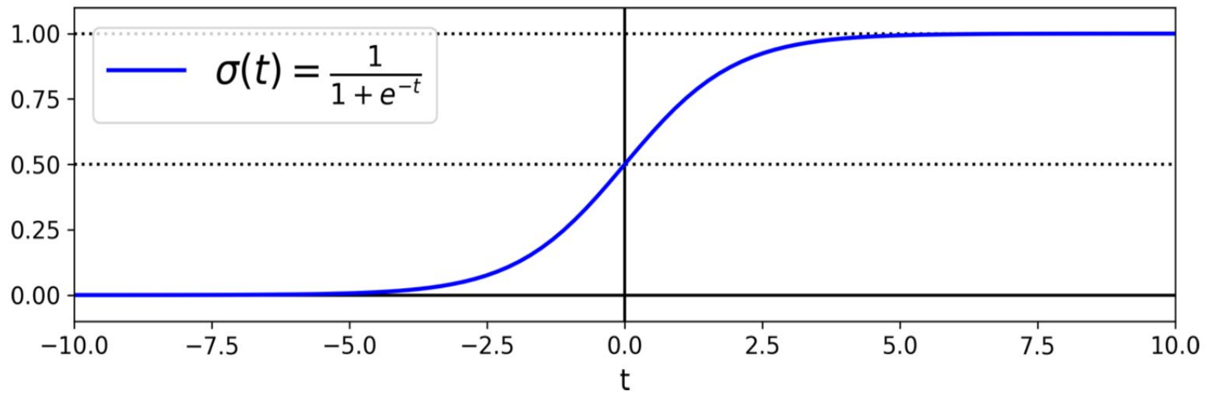
This makes it a **binary classifier**.

How does Logistic Regression work?

Just like a Linear Regression model, a Logistic Regression model computes a weighted sum of the input features, but instead of outputting the result directly like the Linear Regression model does, it outputs the logistic of this result.

Logistic Regression model estimated probability

$$\hat{y} = \sigma\boxed{(x^T\theta)} \Leftarrow \text{Linear regression}$$



Logistic Regression model prediction

$$\hat{y} = \begin{cases} 0 \; if \; \hat{p} < 0.5 \\ 1 \; if \; \hat{p} \geq 0.5 \end{cases}$$

Feature Engineering

- Our model requires all predictor variables to be numeric
- But categorical data is often useful for prediction as well

Feature Engineering: Process of transforming data before model fitting so that the model can capture more kinds of patterns

One-hot encoding or nominal encoding (nominal features)

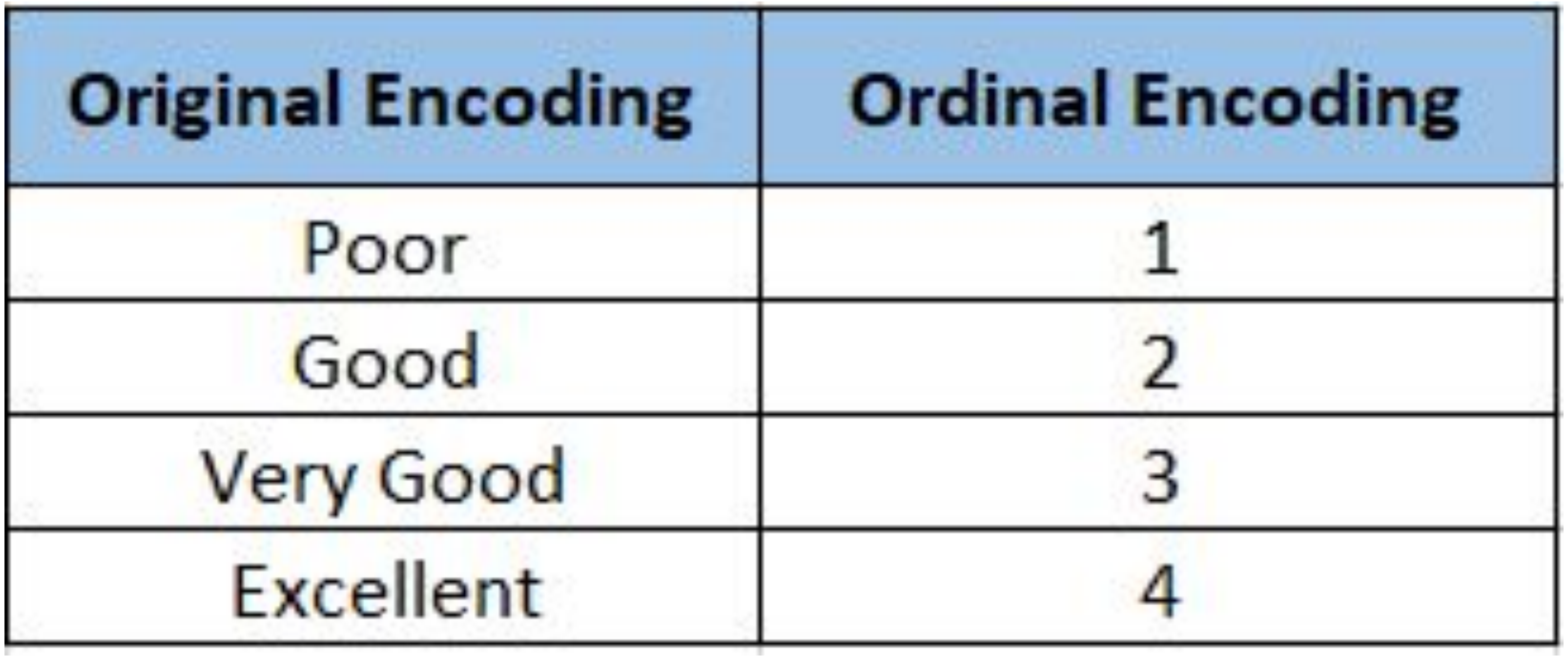
Most popular technique to transform categorical data (nominal) into numeric data - One-hot encoding

Four unique categories $\longrightarrow$ Four new features, only one feature has value 1 in each row

$$\begin{array}{l} \text{West} \\ \text{West} \\ \text{Northeast} \\ \text{South} \\ \text{West} \\ \text{Midwest} \end{array} \longrightarrow \begin{array}{cccc} \text{West} & \text{Northeast} & \text{South} & \text{Midwest} \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}$$

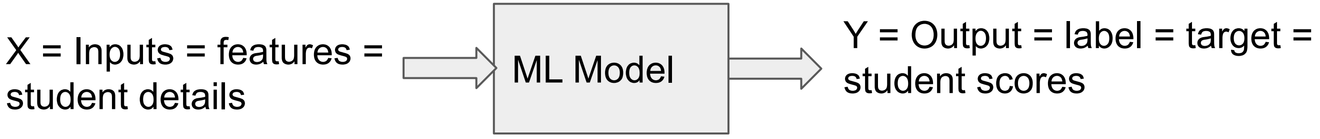
$$X$$

Ordinal encoding (ordinal features)



Original Encoding	Ordinal Encoding
Poor	1
Good	2
Very Good	3
Excellent	4

Practice


X = Inputs = features = student details
ML Model
Y = Output = label = target = student scores

Data:

https://drive.google.com/file/d/1XhBrWwwwXFUZcAolQk2kL_GJO7RXrLV3/view?usp=sharing

Github: https://github.com/chongjason914/scikit-learn-tutorial/blob/main/feature-encoding.ipynb

Tutorial:

https://towardsdatascience.com/guide-to-encoding-categorical-features-using-scikit-learn-for-machine-learning-5048997a5c79

Note: Do not worry about Pandas (get_dummies and map) methods and gradient boosting ML algorithm

THANK YOU