Linear Regression - Predicting Insurance Prices

(Exploratory Project)

Mr. Shrey Gupta

B.Tech Student
Mathematics and Computing
Indian Institute of Technology (BHU)
Varanasi – 221 005.

December 6, 2021

Outline

Definition 1.1

What is Machine Learning ?

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience.

Definition 1.2

What is Linear Regression ?

Supervised learning algorithm in which we have to fit a function f that maps our inputs X to the corresponding function values $f(x)$.

Introduction and Motivation

Linear Regression can be used to solve a variety of real-world problems like predicting the housing prices, predicting the sales for a particular year of a company and many more...
Suppose we are given the data of an insurance company which contains the charges they charge for their insurance for a person using the provided information by the person.

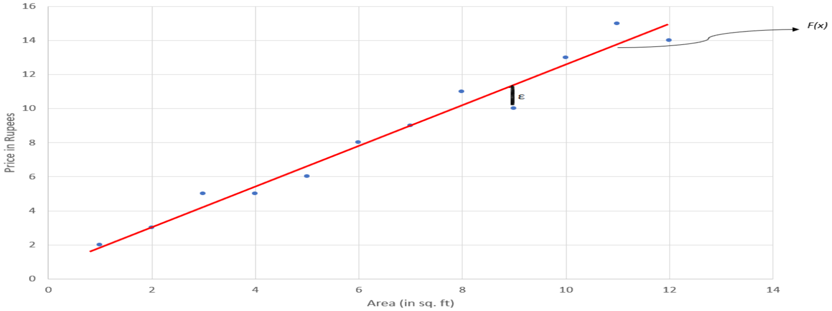
age	sex	bmi	children	smoker	region	charges
19	female	27.9	0	yes	southwest	16884.924
18	male	33.77	1	no	southeast	1725.5523
28	male	33	3	no	southeast	4449.462
33	male	22.705	0	no	northwest	21984.47061
32	male	28.88	0	no	northwest	3866.8552
31	female	25.74	0	no	southeast	3756.6216
46	female	33.44	1	no	southeast	8240.5896
37	female	27.74	3	no	northwest	7281.5056
37	male	29.83	2	no	northeast	6406.4107
60	female	25.84	0	no	northwest	28923.13692
25	male	26.22	0	no	northeast	2721.3208
62	female	26.29	0	yes	southeast	27808.7251
23	male	34.4	0	no	southwest	1826.843

How to solve the above Problem ?

- Data Analysis
- Cleaning the data and preparing the dataset for training
- Make a model that best fits the data
  - Maximum Likelihood Estimation(MLE)
  - Maximum a Posterior Estimation(MAP)
  - Bayesian Linear Regression(BLR)
- Make Predictions

Literature Survey

Consider the following graph:



$$f(x) = \theta_0 + \theta_1 x$$

$$y_n = f(x_n) + \epsilon,$$

where ϵ is the measurement noise.

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

Let us consider the likelihood function

$$p(y|x) = \mathcal{N}(y|f(x), \sigma^2)$$

where $x \in \mathbb{R}^D$ are the inputs,
and $y \in \mathbb{R}$ are the targets.
Here D refers to the number of features.
Also, $y = f(x) + \epsilon$
where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is an independent and identically distibuted random variable that represents the noise.

$$f(x) = x^T\theta$$

where x is the feature vector,
and θ is the model parameters vector.

Hence,

$$p(y|x) = \mathcal{N}(y|x^T\theta, \sigma^2)$$
$$\Leftrightarrow y = x^T\theta + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Consider the training set,

$$Z = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

consisting of N inputs $x_n \in \mathbb{R}^D$ and targets $y_n \in \mathbb{R}$, n=1, 2, …, N.

Now, $p(Y|X,\theta) = p(y_1, y_2, \ldots, y_n | x_1, x_2, \ldots, x_n, \theta)$
As we know that y_i and y_j are conditionally independent given their respective inputs x_i and x_j.

$$\begin{aligned} \therefore p(Y|X,\theta) &= \prod_{n=1}^{N} p(y_n|x_n,\theta) \\ &= \prod_{n=1}^{N} \mathcal{N}(y_n|x_n^T\theta, \sigma^2) \end{aligned}$$

where $X = \{x_1, x_2, \ldots, x_n\}$ is the input set.
and $Y = \{y_1, y_2, \ldots, y_n\}$ is the target set.

Maximizing the likelihood means maximizing the predictive distribution of the training data given the model parameters.
Mathematically, we have to evaluate

$$\theta_{ML} = \arg\max_{\theta} P(Y|X, \theta)$$

The negative log likelihood is also called Loss function.

$$\begin{aligned}\mathcal{L}(\theta) &= \frac{\sum_{n=1}^{N}(y_n - x_n^T\theta)^2}{2\sigma^2} \\ &= \frac{(y - X\theta)^T(y - X\theta)}{2\sigma^2}\end{aligned}$$

$$\boxed{\theta_{ML} = (X^T X)^{-1} X^T y}$$

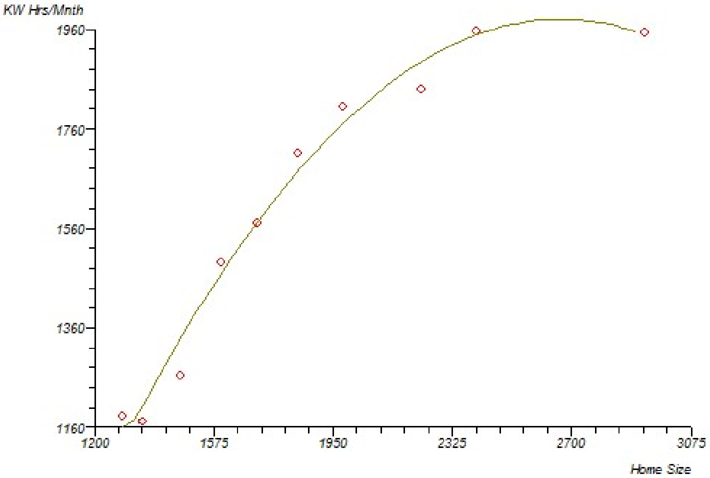
where,

$$X = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(D)} \\ x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(D)} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ x_N^{(1)} & x_N^{(2)} & \dots & x_N^{(D)} \end{bmatrix} \in \mathbb{R}^{\mathbb{NXD}}$$

is the feature matrix consisting of N training inputs and D features.

$$y = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix} \in \mathbb{R}^N \qquad \text{and} \qquad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ . \\ . \\ . \\ \theta_D \end{bmatrix} \in \mathbb{R}^D$$

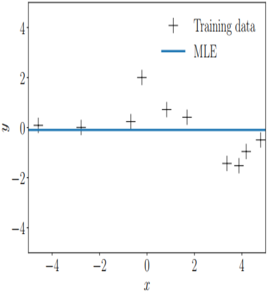
Consider the following graph:



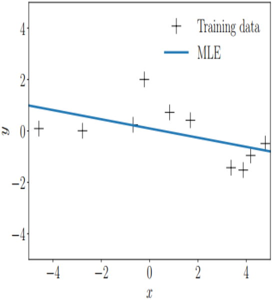
$p(y|x,\theta) = \mathcal{N}(y|\phi^T(x)\theta, \sigma^2)$
$\Leftrightarrow y = \phi^T(x)\theta + \epsilon = \sum_{k=0}^{K-1} \theta_k \phi_k(x) + \epsilon$ where $\phi : \mathbb{R}^D \to \mathbb{R}^K$ is a non linear transformation of inputs x.
$\phi_k : \mathbb{R}^D \to \mathbb{R}$ is the kth component of the vector ϕ.
For example

$$\phi(x) = \begin{bmatrix} 1 \\ x \\ x^2 \\ . \\ . \\ . \\ x^{K-1} \end{bmatrix} = \begin{bmatrix} \phi_0(x) \\ \phi_1(x) \\ \phi_2(x) \\ . \\ . \\ . \\ \phi_{K-1}(x) \end{bmatrix} \in \mathbb{R}^K$$

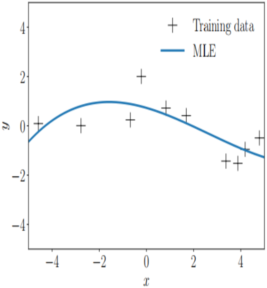
$$\boxed{\theta_{ML} = (\Phi^T \Phi)^{-1} \Phi^T y}$$


Training data
MLE
x
y

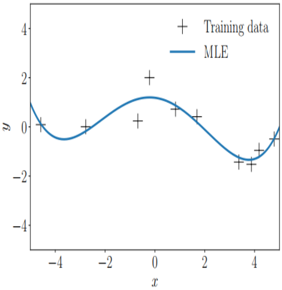
(a) $M = 0$


Training data
MLE
x
y

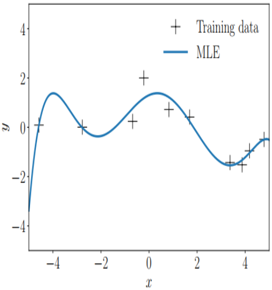
(b) $M = 1$


Training data
MLE
x
y

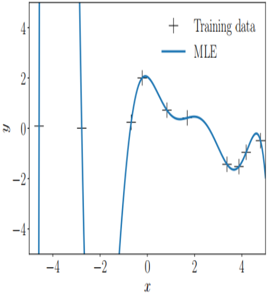
(c) $M = 3$


Training data
MLE
y
x

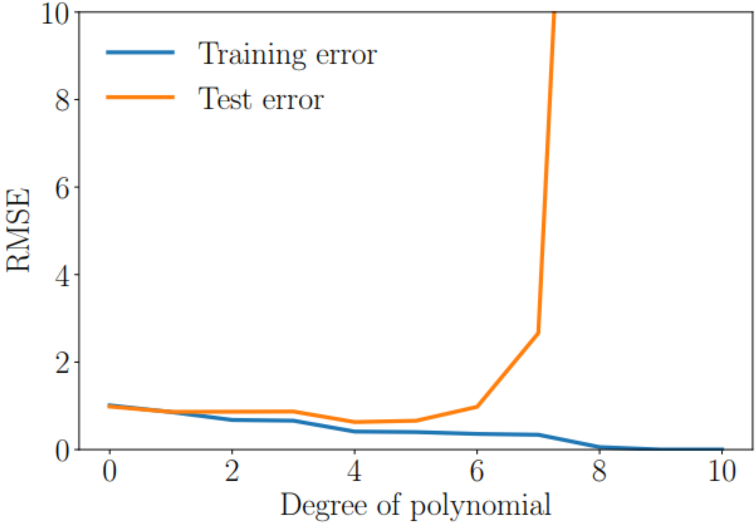
(d) $M = 4$


Training data
MLE
y
x

(e) $M = 6$

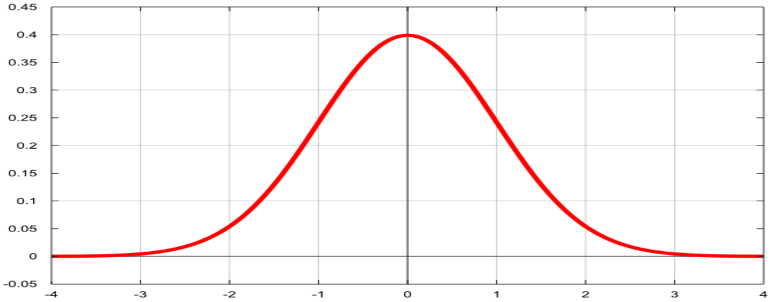

Training data
MLE
y
x

(f) $M = 9$


Training error
Test error
RMSE
Degree of polynomial
0
2
4
6
8
10

Why overfitting occurs? How to overcome it?

$$p(\theta) = \mathcal{N}(0, 1)$$



Given our training data X, Y. Here we will maximize the posterior distribution $p(\theta|X, Y)$. This method is called maximum a posterior(MAP) estimation.

Now applying Baye's theorum -

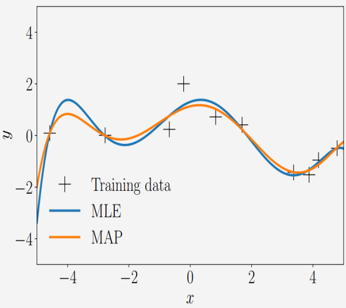
$$p(\theta|X, Y) = \frac{p(Y|X, \theta)p(\theta)}{p(Y|X)}$$

$$\log p(\theta|X, Y) = \log p(Y|X, \theta) + \log p(\theta) + const$$

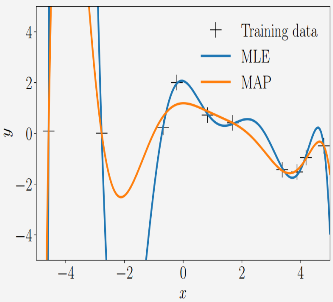
We assume that $p(\theta) = \mathcal{N}(0, b^2 I)$.

$$\boxed{\theta_{MAP} = (\Phi^T \Phi + \frac{\sigma^2}{b^2} I)^{-1} \Phi^T y}$$

$$\theta_{RLS} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T y$$



(a) Polynomials of degree 6.



(b) Polynomials of degree 8.

Bayesian linear regression does not attempt to compute a point estimate of the parameters, but instead the full posterior distribution over parameters(θ) is taken into account when making predictions. This means we compute the mean over all plausible parameter settings.
For Bayesian linear regression, consider,
prior $p(\theta) = \mathcal{N}(m_o, S_o)$
likelihood $p(y|x,\theta) = \mathcal{N}(y|\phi^T(x)\theta, \sigma^2)$

Prior Predictions

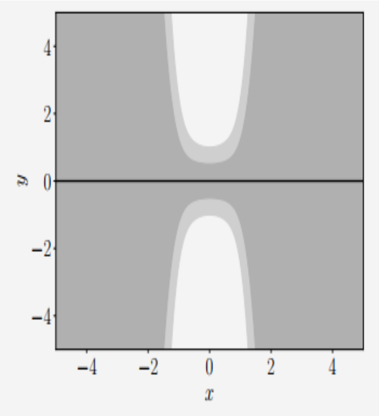
$$\begin{aligned} p(y_*|x_*) &= \int p(y_*|x_*,\theta)p(\theta)d\theta \\ &= E_\theta[p(y_*|x_*,\theta)] \end{aligned}$$

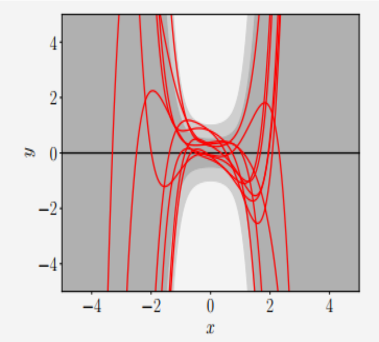
$$p(y_*|x_*) = \mathcal{N}(\phi^T(x_*)m_o, \phi^T(x_*)S_o\phi(x_*) + \sigma^2)$$

But if we want to find out the distribution of $f(x_*) = \phi^T(x)\theta$.

$$f(x_*) = \mathcal{N}(\phi^T(x_*)m_o, \phi^T(x_*)S_o\phi(x_*))$$

Example If we chose parameter prior to be $p(\theta) = \mathcal{N}(0, \frac{I}{4})$.





(b) Samples from the prior distribution over functions.

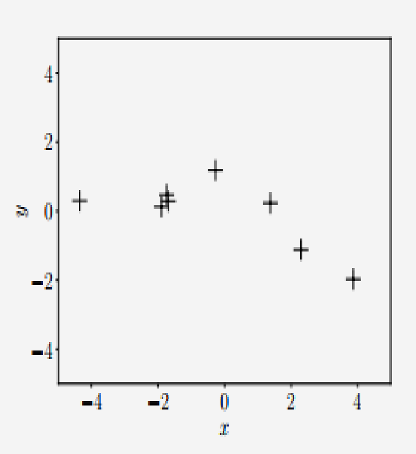
Posterior Predictions

$$p(\theta|X,Y) = \frac{p(Y|X,\theta)p(\theta)}{p(Y|X)}$$

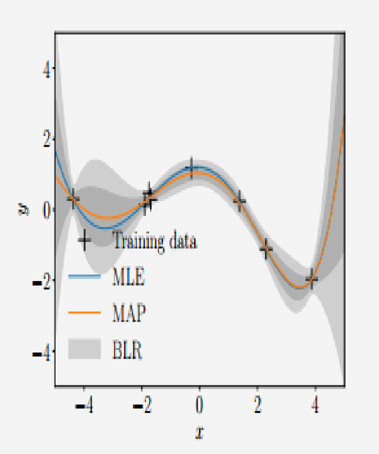
$$\begin{aligned} p(\theta|X,Y) &= \mathcal{N}(\theta|m_N, S_N) \\ S_N &= (\sigma^{-2}\Phi^T\Phi + S_o^{-1})^{-1} \\ m_N &= S_N(\sigma^{-2}\Phi^T y + S_o^{-1} m_o) \end{aligned}$$

$$\begin{aligned} p(y_*|X,Y,x_*) &= \int p(y_*|x_*,\theta)p(\theta|X,Y)d\theta \\ &= \int \mathcal{N}(y_*|\phi^T(x_*)\theta, \sigma^2)\mathcal{N}(\theta|m_N, S_N)d\theta \\ &= \mathcal{N}(y_*|\phi^T(x_*)m_N, \phi^T(x_*)S_N\phi(x_*) + \sigma^2) \end{aligned}$$

$$\therefore f(x_*) = \mathcal{N}(\phi^T(x_*)m_N, \phi^T(x_*)S_N\phi(x_*))$$



y
x

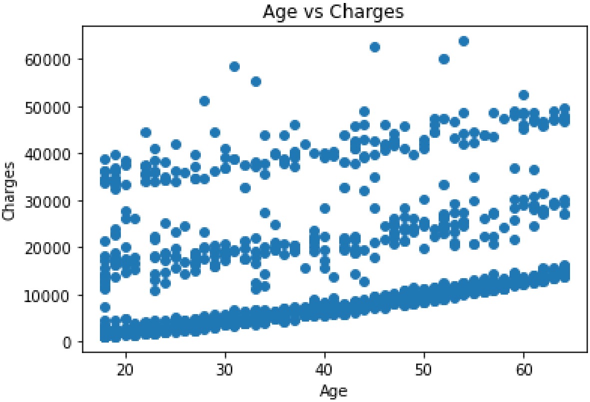


Training data
MLE
MAP
BLR
y
x

Given dataset,

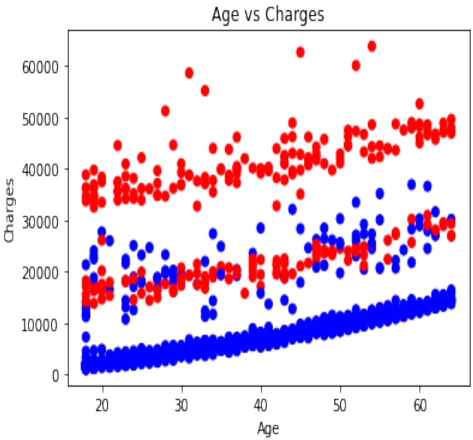
age	sex	bmi	children	smoker	region	charges
19	female	27.9	0	yes	southwest	16884.924
18	male	33.77	1	no	southeast	1725.5523
28	male	33	3	no	southeast	4449.462
33	male	22.705	0	no	northwest	21984.47061
32	male	28.88	0	no	northwest	3866.8552
31	female	25.74	0	no	southeast	3756.6216
46	female	33.44	1	no	southeast	8240.5896
37	female	27.74	3	no	northwest	7281.5056
37	male	29.83	2	no	northeast	6406.4107
60	female	25.84	0	no	northwest	28923.13692
25	male	26.22	0	no	northeast	2721.3208
62	female	26.29	0	yes	southeast	27808.7251
23	male	34.4	0	no	southwest	1826.843

Main Results



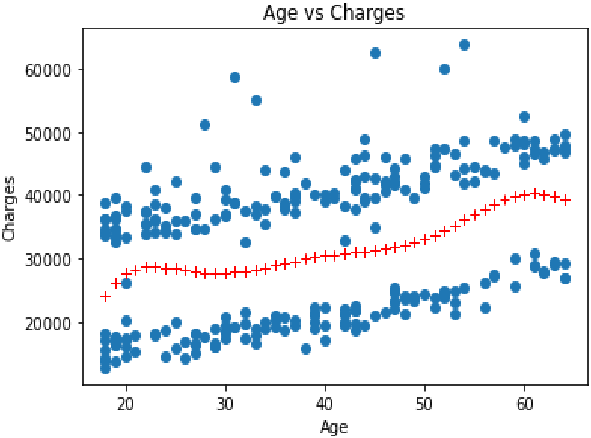
Correlation between Age and Charges = 0.2999

Main Results

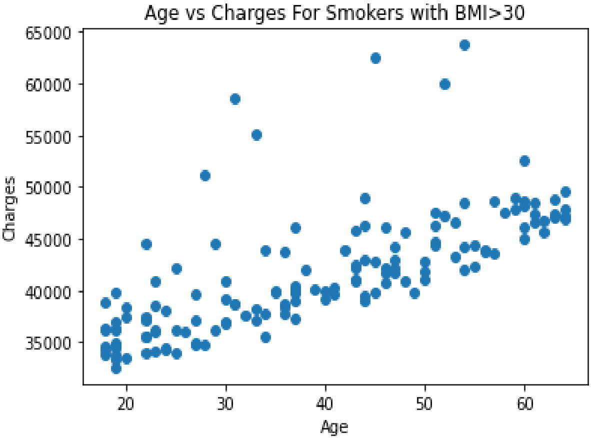


Red:Smoker Blue:Non-Smoker

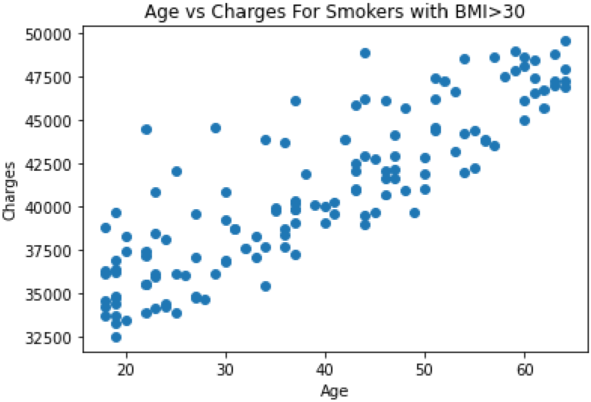
Smokers


Age vs Charges
Charges
60000
50000
40000
30000
20000
20
30
40
50
60
Age

Smokers with BMI>30

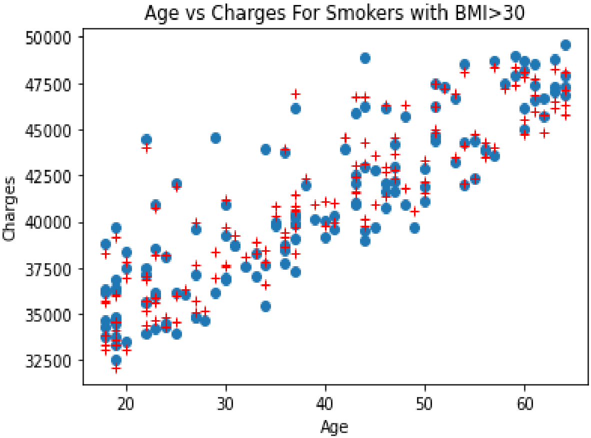


Smokers with BMI>30

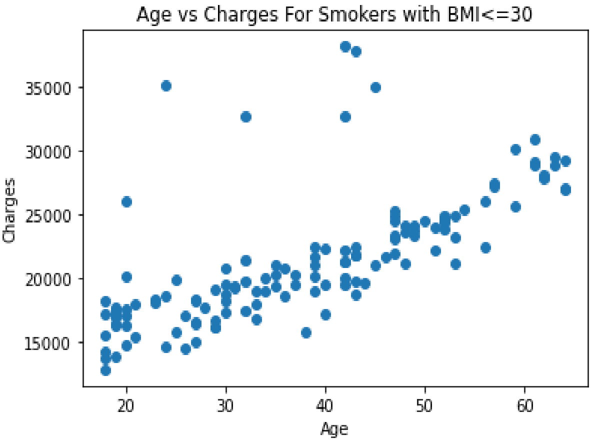


Correlation between Age and Charges = 0.8623745

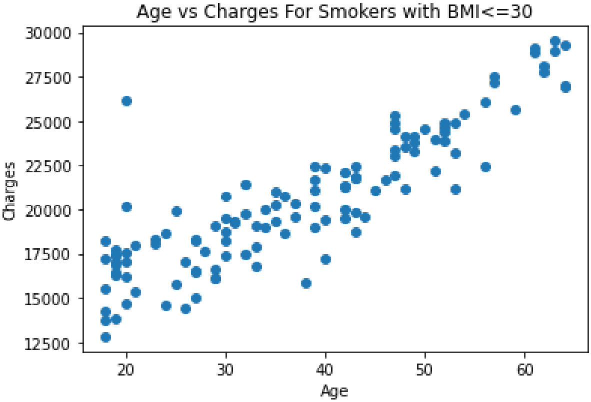
Smokers with BMI>30



Smokers with BMI<=30

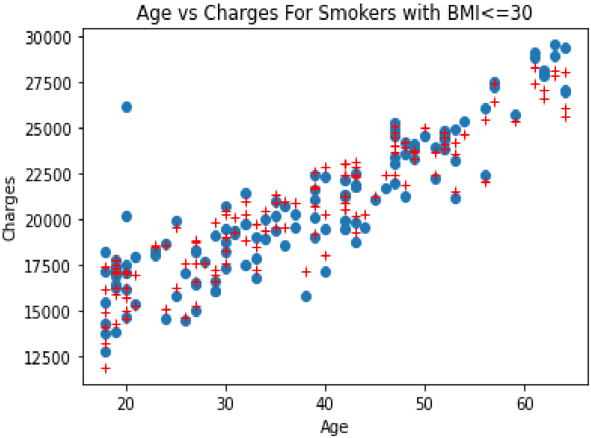


Smokers with BMI<=30

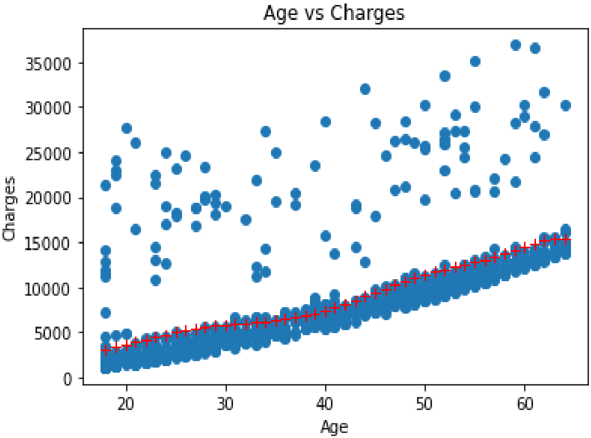


Correlation between Age and Charges = 0.8623745

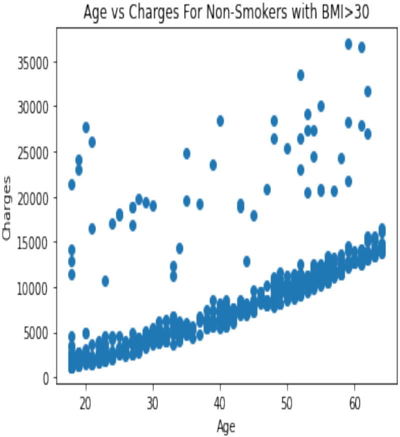
Smokers with BMI<=30

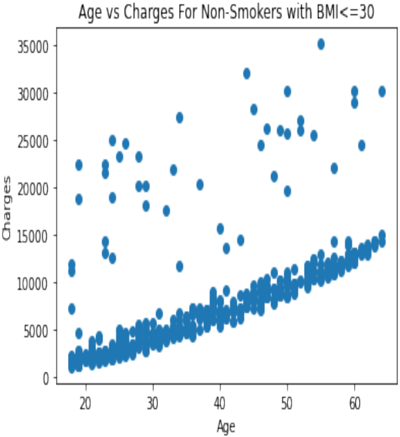


Non- Smokers

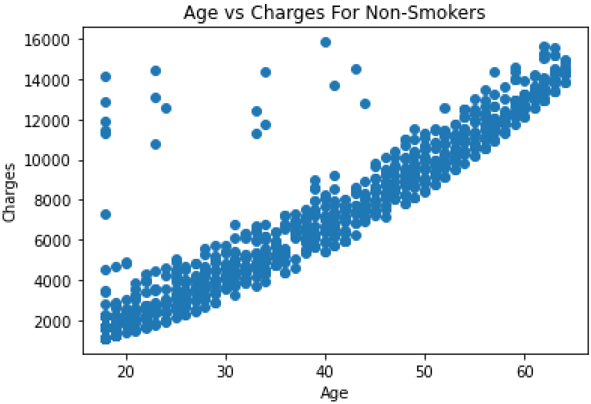

Age vs Charges
35000
30000
25000
20000
15000
10000
5000
0
Charges
20
30
40
50
60
Age

Non- Smokers


Age vs Charges For Non-Smokers with BMI>30
Charges
Age
0
5000
10000
15000
20000
25000
30000
35000
20
30
40
50
60

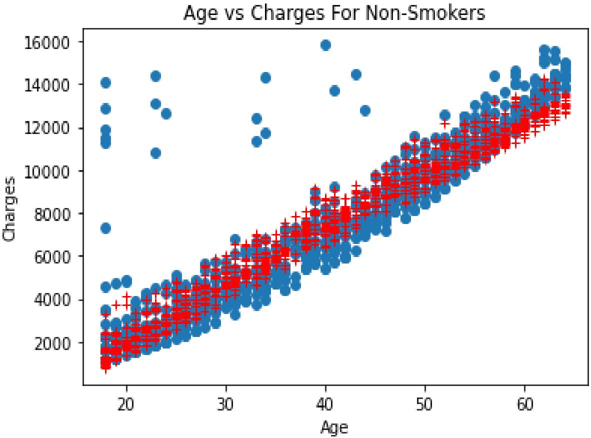

Age vs Charges For Non-Smokers with BMI<=30
Charges
Age
0
5000
10000
15000
20000
25000
30000
35000
20
30
40
50
60

Non- Smokers



Correlation between Age and Charges = 0.92874

Non- Smokers


Age vs Charges For Non-Smokers
16000
14000
12000
10000
8000
6000
4000
2000
Charges
20
30
40
50
60
Age

Main Results

```
def model1(x,Y, regularization=0, power=1):
  temp=np.ones(x.shape[0])
  X=np.c_[x,temp]
  theta=np.dot(np.dot(np.linalg.inv(np.dot(X.T,X)+regularization),X.T),Y)
  Y_pred=np.dot(X,theta)
  loss=np.sum(np.square(Y-Y_pred))/x.shape[0] + regularization*np.sum(np.square
 (theta))
  return theta,Y_pred,loss
```

Main Results

```
def predict(x):
  yp=[]
  x1=np.append(x,[1])
  x1=np.append(x1[:4],x1[5:])
  if x[4]==1:
    if x[2]>30:
      yp=np.dot(x1,theta1)
    else:
      yp=np.dot(x1,theta2)
  else:
    yp=np.dot(x1,theta3)
  return yp
```

Conclusion

- Based on the understandings and results from the Literature Survey we trained a model that predicts the insurance price for a person given his Age, Sex, BMI, Number of Children, Smoking Status and Region.
- First we analysed the dataset. The given data was too much scattered to be able to get a good model that fits it. So we segregated the data for smokers and non-smokers separately. Then we further segregated it using BMI. Still some scattered points were left due to unknown reasons of a particular person so we made a upper-bound to the data. This way we trained the model and finally we were able to get a decent result.

References

- Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong - Mathematics For Machine Learning-Cambridge University Press (2019)
- https://www.coursera.org/learn/machine-learning/home/welcome
- https://towardsdatascience.com/

THANK YOU