# Name : Shubham Sapkal

# Roll No. : 2012118

# subject: ML DL

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import statsmodels.api as sm

# We can override the default matplotlib styles with those of Seaborn

#import seaborn as sns

#sns.set()

# Load the data from a .csv in the same folder

data = pd.read\_csv('1.01 Simple linear regression.csv')

data

# This method gives us very nice descriptive statistics. We don't need this as of now, but will later on!

data.describe()

# Following the regression equation, our dependent variable (y) is the GPA

y = data ['GPA']

# Similarly, our independent variable (x) is the SAT score

x1 = data ['SAT']

# Plot a scatter plot (first we put the horizontal axis, then the vertical axis)

plt.scatter(x1,y)

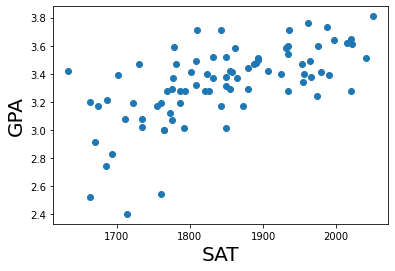
# Name the axes

plt.xlabel('SAT', fontsize = 20)

plt.ylabel('GPA', fontsize = 20)

# Show the plot

plt.show()



# Add a constant. Essentially, we are adding a new column (equal in lenght to x), which consists only of 1s

x = sm.add\_constant(x1)

# Fit the model, according to the OLS (ordinary least squares) method with a dependent variable y and an idependent x

results = sm.OLS(y,x).fit()

# Print a nice summary of the regression. That's one of the strong points of statsmodels -> the summaries

results.summary()

# Create a scatter plot

plt.scatter(x1,y)

# Define the regression equation, so we can plot it later

yhat = 0.0017\*x1 + 0.275

# Plot the regression line against the independent variable (SAT)

fig = plt.plot(x1,yhat, lw=4, c='orange', label ='regression line')

# Label the axes

plt.xlabel('SAT', fontsize = 20)

plt.ylabel('GPA', fontsize = 20)

plt.show()

