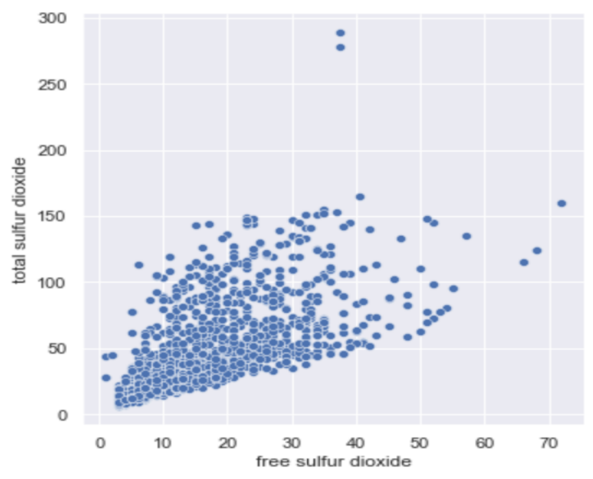
Machine Learning Assignment

Name: Aaron Dlima
Register No: 21BDA23

Document 5-6 key insights from EDA and support each point with a visualization

```
sns.relplot(y = "total sulfur dioxide", x = "free sulfur dioxide", data = redwine)
```

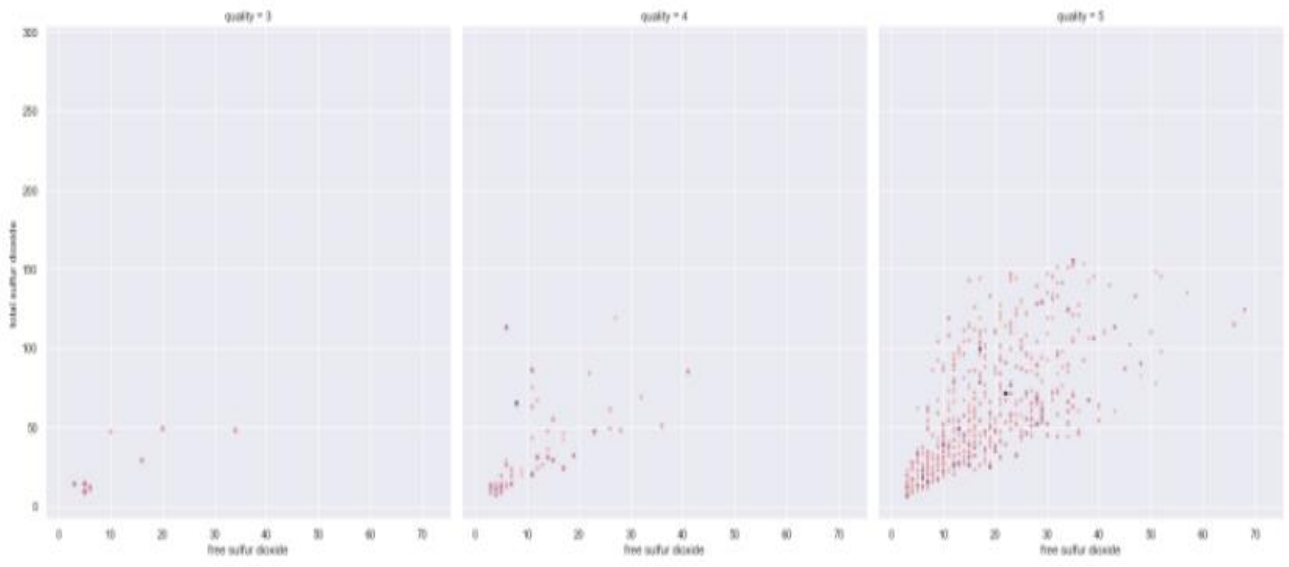
```
<seaborn.axisgrid.FacetGrid at 0x1ffea4acca0>
```

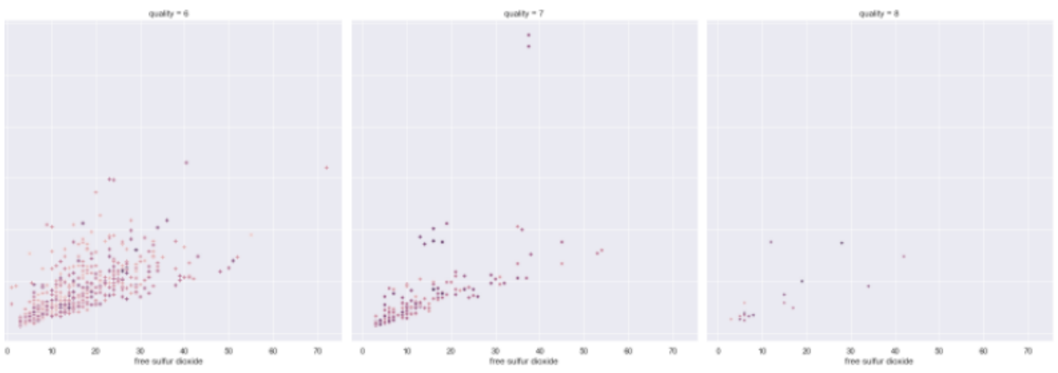


Here we can see thatthere is a positive correlation between the free sulfur dioxide and total sulfur dioxide. So we can say that the values go along.
https://seaborn.pydata.org/generated/seaborn.relplot.html
As shown above here we can see how total sulfur dioxide and free sulfur dioxide are related interms of each quality.

```
sns.relplot(y = "total sulfur dioxide", x = "free sulfur dioxide",
            sizes = (20,100),height = 7, hue= "alcohol",
            col = "quality",style='quality', data = redwine)
```

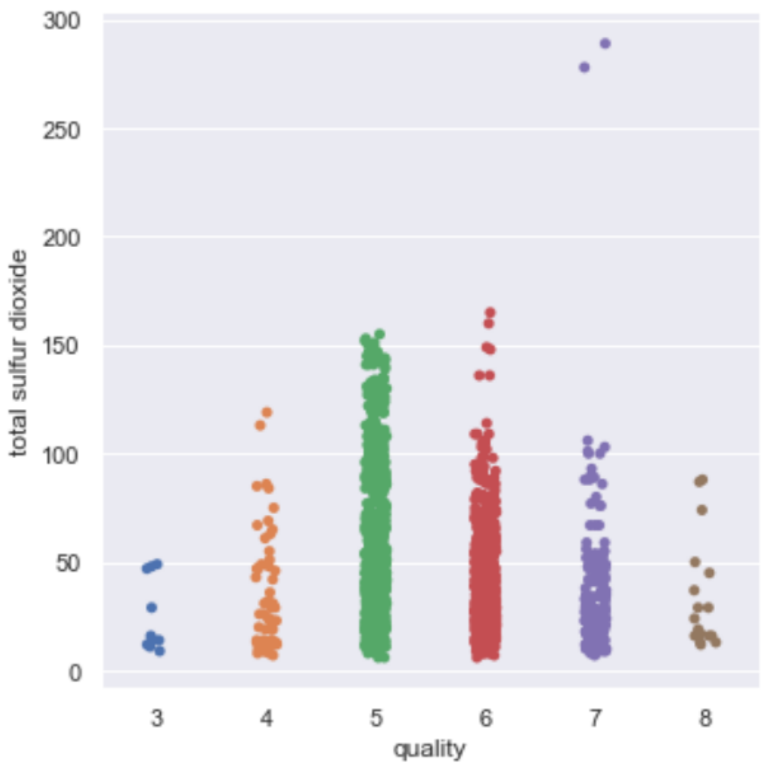
```
<seaborn.axisgrid.FacetGrid at 0x1add9d9a1f0>
```





```
sns.catplot(y = "total sulfur dioxide", x = "quality",data = redwine)
```

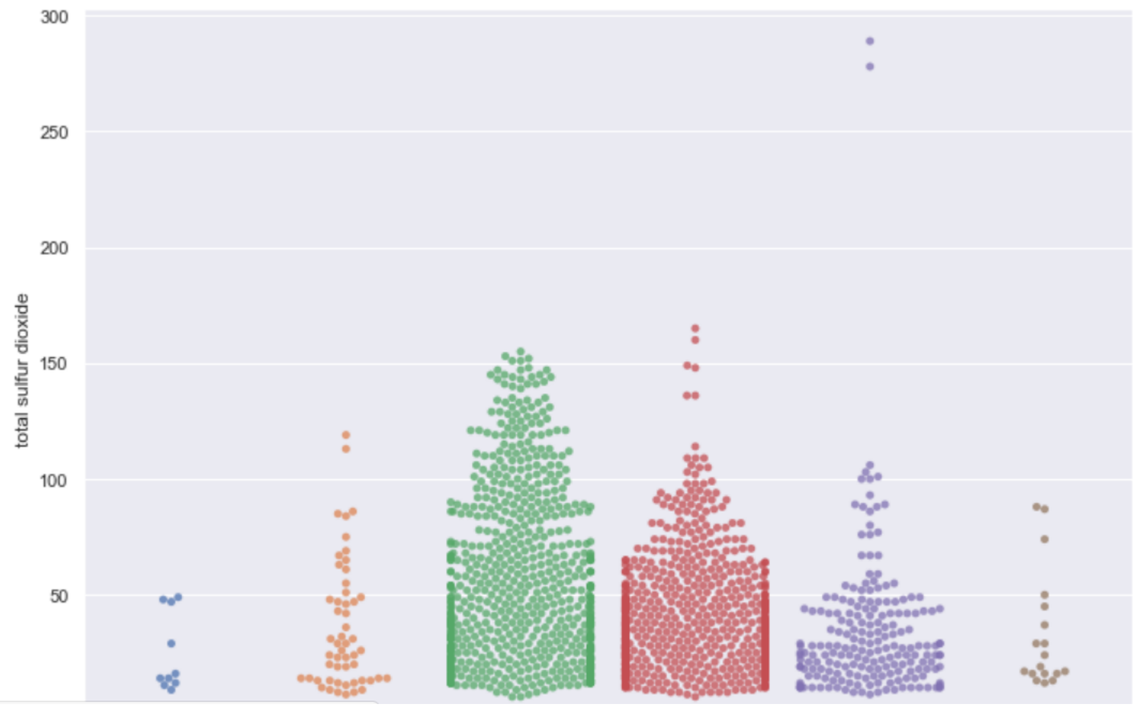
```
<seaborn.axisgrid.FacetGrid at 0x1ff81944fa0>
```



According to categories of quality we can see the amount of total sulfur dioxide. Here we can see that quality 5 and 6 are dense interms of total sulfur dioxide.
https://seaborn.pydata.org/generated/seaborn.catplot.html

```
sns.swarmplot(y = "total sulfur dioxide", x = "quality",alpha = .75,data = redwine)
```

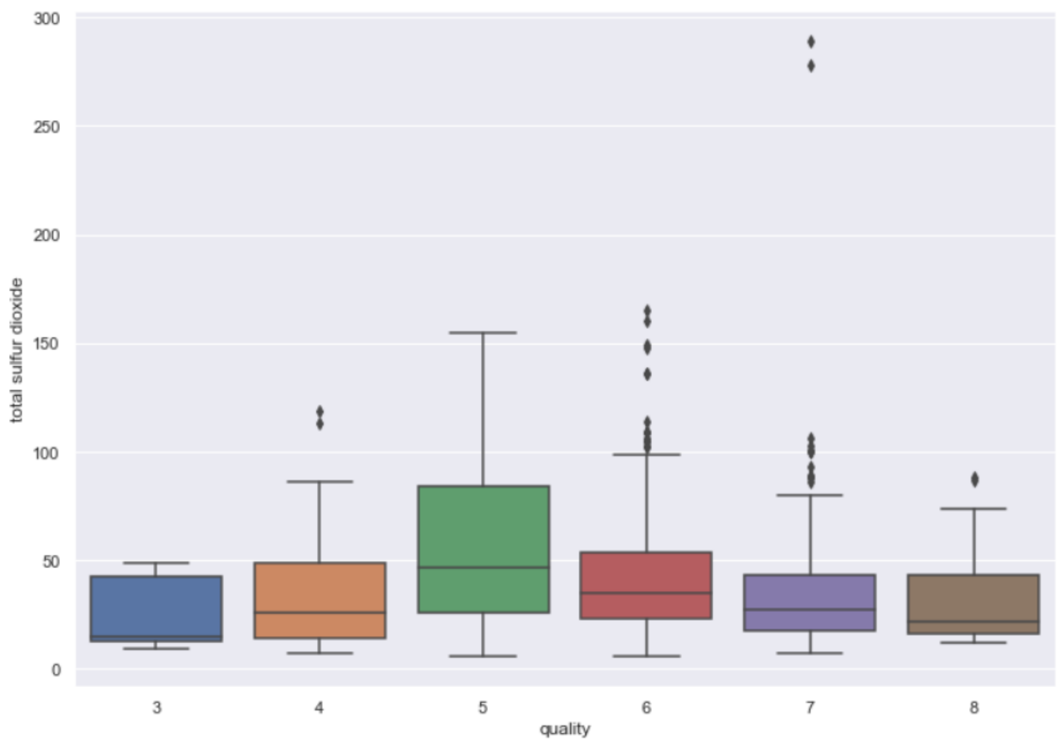
```
<AxesSubplot:xlabel='quality', ylabel='total sulfur dioxide'>
```



Here we have used swarm plot to understand the distribution better.
https://seaborn.pydata.org/generated/seaborn.swarmplot.html

```
sns.boxplot(y = "total sulfur dioxide", x = "quality",data = redwine)
```

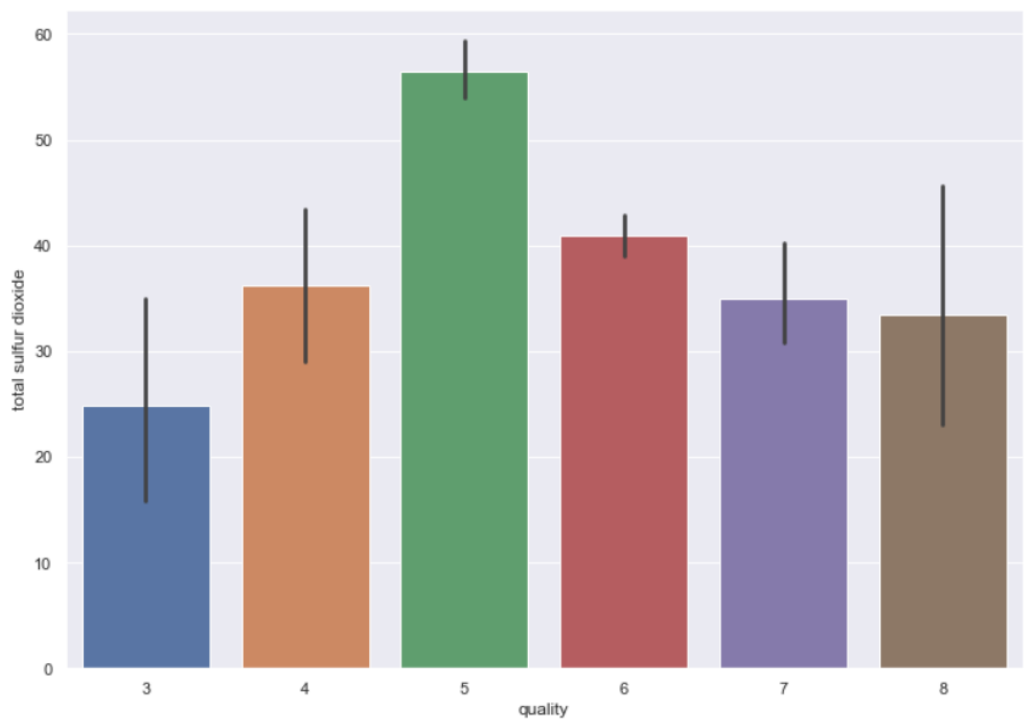
```
<AxesSubplot:xlabel='quality', ylabel='total sulfur dioxide'>
```



This provides a visual summary of the data interms of quality to quickly identify mean values, the dispersio and signs of skewness.Here we can see quartile, median and outliers under each quality.
https://seaborn.pydata.org/generated/seaborn.boxplot.html

```
sns.barplot(y = "total sulfur dioxide", x = "quality",data = redwine)
```

```
<AxesSubplot:xlabel='quality', ylabel='total sulfur dioxide'>
```

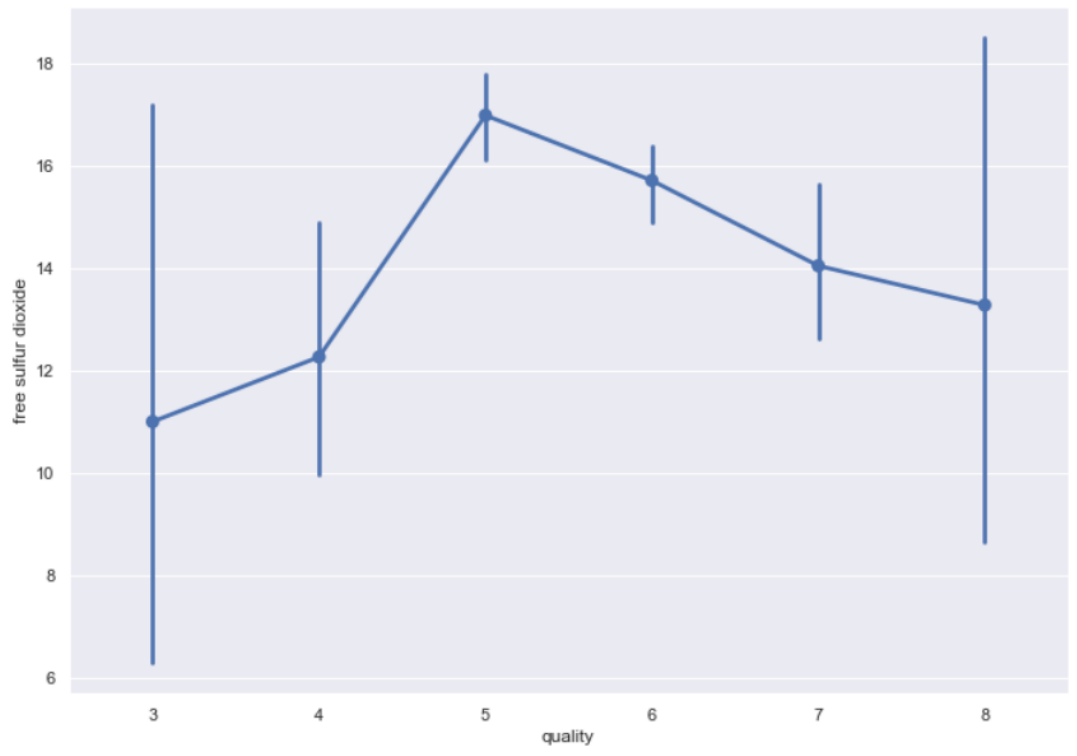


Here we can compare between different quality of the wine interms of total sulfur dioxide.
https://seaborn.pydata.org/generated/seaborn.barplot.html

provides some indication of the uncertainty around that estimate using error bars.

```
sns.pointplot(y = "free sulfur dioxide", x = "quality",data = redwine)
```

```
<AxesSubplot:xlabel='quality', ylabel='free sulfur dioxide'>
```



This provides some indication of the uncertainty around that estimate using error bars.
https://seaborn.pydata.org/generated/seaborn.pointplot.html
----

2. i) What are the assumptions of linear regression?
There are four assumptions of linear regression model:
Linearity: The relationship between X and the mean of Y is linear.
Homoscedasticity: The variance of residual is the same for any value of X.
Independence: Observations are independent of each other.
Normality: For any fixed value of X, Y is normally distributed.
https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R5_Correlation-Regression/R5_Correlation-Regression4.html#:~:text=There%20are%20four%20assumptions%20associated,are%20independent%20of%20each%20other.

----
ii) How can we evaluate a Regression model? Define each metric and its interpretation.
There are 3 main metrics for model evaluation in regression
1. R Square/Adjusted R Square
2. Mean Square Error(MSE)/Root Mean Square Error(RMSE)
3. Mean Absolute Error(MAE)

1. R Square/Adjusted R Square
R Square measures how much variability in dependent variable can be explained by the model. It is the square of the Correlation Coefficient(R) and that is why it is called R Square.

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

R Square value is between 0 to 1 and a bigger value indicates a better fit between prediction and actual value.R Square is a good measure to determine how well the model fits the dependent variables.
However, it does not take into consideration of overfitting problem. If our regression model has many independent variables, because the model is too complicated, it may fit very well to the training data but performs badly for testing data. That is why Adjusted R Square is introduced because it will penalize additional independent variables added to the model and adjust the metric to prevent overfitting issues.

2. Mean Square Error(MSE)/Root Mean Square Error(RMSE)
While R Square is a relative measure of how well the model fits dependent variables, Mean Square Error is an absolute measure of the goodness for the fit.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

It gives us an absolute number on how much oour predicted results deviate from the actual number. We cannot interpret many insights from one single result but it gives us a real number to compare against other model results and helps us select the best regression model.
Root Mean Square Error(RMSE) is the square root of MSE. It is used more commonly than MSE because firstly sometimes MSE value can be too big to compare easily. Secondly, MSE is calculated by the square of error, and thus square root brings it back to the same level of prediction error and makes it easier for interpretation.
3. Mean Absolute Error(MAE)
Mean Absolute Error(MAE) is similar to Mean Square Error(MSE). However, instead of the sum of square of error in MSE, MAE is taking the sum of the absolute value of error.

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

Compare to MSE or RMSE, MAE is a more direct representation of sum of error terms. MSE gives larger penalization to big prediction error by square it while MAE treats all errors the same.
https://towardsdatascience.com/what-are-the-best-metrics-to-evaluate-your-regression-model-418ca481755b

----

iii) Can R squared be negative?

Theoretically R Squared can be negative when the explanatory variables and the target variable are inversely related, that is when the model is inversely explaining the target.

----

iv) What is dummy variable trap?

The Dummy variable trap is a scenario where there are attributes that are highly correlated (Multicollinear) and one variable predicts the value of others. When we use *one-hot encoding* for handling the categorical data, then one dummy variable (attribute) can be predicted with the help of other dummy variables. Hence, one dummy variable is highly correlated with other dummy variables. Using all dummy variables for regression models leads to a *dummy variable trap*. So, the regression models should be designed to exclude one dummy variable.

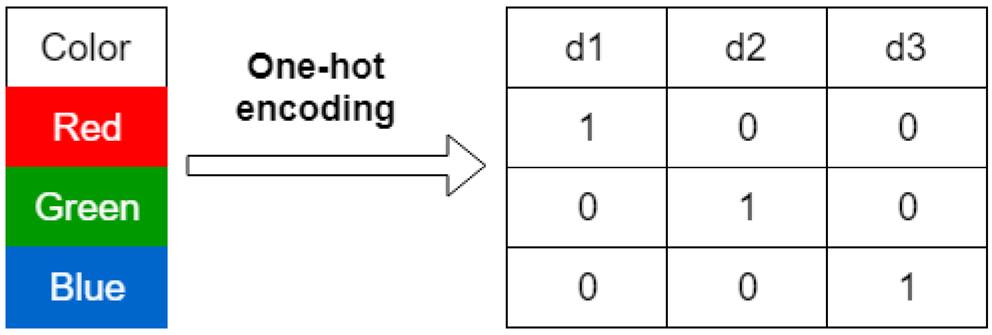
For Example –

Let's consider the case of gender having two values *male* (0 or 1) and *female* (1 or 0). Including both the dummy variable can cause redundancy because if a person is not male in such case that person is a female, hence, we don't need to use both the variables in regression models. This will protect us from the dummy variable trap.
https://www.geeksforgeeks.org/ml-dummy-variable-trap-in-regression-models/

----

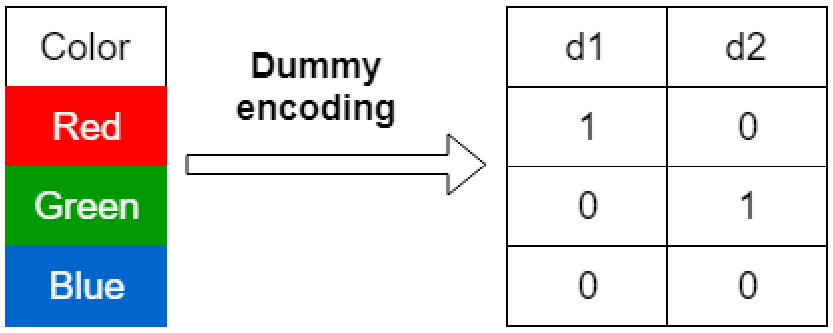
v) Is One Hot Encoding different from Dummy Variables?

In one-hot encoding, we create a new set of dummy (binary) variables that is equal to the number of categories (k) in the variable. For example, let's say we have a categorical variable *Color* with three categories called "Red", "Green" and "Blue", we need to use three dummy variables to encode this variable using one-hot encoding. A dummy (binary) variable just takes the value 0 or 1 to indicate the exclusion or inclusion of a category.



In one-hot encoding,
"Red" color is encoded as [1 0 0] vector of size 3.
"Green" color is encoded as [0 1 0] vector of size 3.
"Blue" color is encoded as [0 0 1] vector of size 3.

Dummy encoding
Dummy encoding also uses dummy (binary) variables. Instead of creating a number of dummy variables that is equal to the number of categories (k) in the variable, dummy encoding uses k-1 dummy variables. To encode the same *Color* variable with three categories using the dummy encoding, we need to use only two dummy variables.



In dummy encoding,
“Red” color is encoded as [1 0] vector of size 2.
“Green” color is encoded as [0 1] vector of size 2.
“Blue” color is encoded as [0 0] vector of size 2.
Dummy encoding removes a duplicate category present in the one-hot encoding.
https://towardsdatascience.com/encoding-categorical-variables-one-hot-vs-dummy-encoding-6d5b9c46e2db#:~:text=A%20dummy%20(binary)%20variable%20just,or%20inclusion%20of%20a%20category.&text=Image%20by%20author)-,In%20one%2Dhot%20encoding%2C,0%5D%20vector%20of%20size%203.
----
vi) How is polynomial regression different from linear regression?

Simple Linear Regression establishes the relationship between two variables using a straight line. It attempts to draw a line that comes closest to the data by finding the slope and intercept which define the line and minimize regression errors. Simple linear regression has only one x and one y variable.
$Y=\theta_o + \theta_1 X + \theta_2 X^2 + \ldots + \theta_m X^m$ + residual error
Polynomial Regression is a one of the types of linear regression in which the relationship between the independent variable x and dependent variable y is modeled as an *nth* degree polynomial. Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted E (y |x). Polynomial Regression provides the best approximation of the relationship between the dependent and independent variable.
https://www.i2tutorials.com/difference-between-simple-linear-regression-and-multi-linear-regression-and-polynomial-regression/
vii) Interpret the screenshot below from the notebook we discussed in class today:
Assuming it's sci-kit learn in python then model.score automates the prediction of data using X_test and compares it with Y_test and by default uses the R-squared metric to so (hence don't need to manually derive y_pred).

The first line says about the accuracy between the actual test data of the independent variable and the predicted values of the test data of independent varable.
Here 1 is the maximum value,it indicates perfect fit.
The second line is a prefect fit between the actual training set of data.
The third line of code and output is not 1 and it indicates high accuracy but not perfect fit between the test data of the dependent variables and independent variable.Similarly with fourth line of code and output.