**Warm Up Section**

* In this section, I have explore three data set - Penguin dataset, Diamond dataset, Insurance dataset.
* I have load the dataset using “pd.read\_csv()” command.
* I have shown five inbuilt functions in “pandas” library which are “describe()”, “head(10)”, dimension of data as row and column, “value\_counts()”, “info()”.
* Calculate the sum of missing entries in the dataset. I have used “Fill the missing values” technique in Penguin dataset, Diamond dataset and “Droping rows with missing values” in the Insurance dataset.
* Convert data type to catagorical.
* Draw visualization graphs as Histogram, Bar Plot, Pair Plot, Heat Map, Scatter plot.

**Linear Regression on the winequality-red Dataset**

I have a performed linear regression on winequality-red Dataseet in python and describe below step by step how I do the job.

* In linear regression, I have used winequality-red Dataseet.
* I have read the dataset with “pd.read\_csv”.
* I have identified some duplicate data set and remove them from the dataset.
* I have remove the null data row using the command “dropna()”.
* Describe the statistical values using “describe()” for the numerical column.
* Select features and target data and consider “quality” as target data.
* Define data matrix X ∈ RN×d and y ∈ RN where, d is the features.
* Define normalize function and normalize all of the fearture data.
* Divide the dataset into training (80%) and test (20%).
* Print training and test data shape.
* Evaluate the w as w = (XTX)-1 XTy.
* Usiing the predicting and calculating data evaluate the mean squared error (MSE).
* Scatter plot of the prediction vs the actual data values.
* Provide the loss value for training data and the weight vector and plot by comparing the predictions vs the actual test data.

In linear regression, the weight is calculated using the Ordinary Least Squares (OLS) technique. This technique have some benefit and drawback. In this techique there is no need any hyperparameters like learning rates which save time. In this case, dependent variable changes with the chage of any independent variables. There is also some drawbacks. A matrix inverse operation is included in this method. Iif the matrix become singular then inverse wiill not be possible.

**Logistic Regression on the Penguins Dataset**

I have a performed logistic regression on winequality-red Dataseet in python and describe below step by step how I do the job.

* In logistic regression, I have used Penguins Dataseet.
* I have read the dataset with “pd.read\_csv”.
* Determine the sum of missing entries and fill up missing value in each column.
* Describe the statistical values using “describe()” for the numerical column.
* Convert features with string datatype to categorical (species,island,sex).
* Define Normalize Function and normalize non-categorical features by finding the min and max values for each column where data renge is 0 to 1.
* Select features and target data and consider “sex” as target data.
* Define data matrix X ∈ RN×d and y ∈ RN where, d is the features.
* Divide the dataset into training (80%) and test (20%).
* Print training and test data shape
* Define Logistic Regression class using “class LogitRegression():”
* Fit the function, use sigmoid function, implement the loss function, use graient descent formula, evaluate the predicted result.
* I have used the hyperparameters: learning\_rate = 1e-3, 1e-4, 1e-5, 1e-6; iterations = 100000; weights = np.random.uniform(0, 1)
* I have printed out the loss values over each iterations with percentage of acciracy.
* In my analysis, best accuracy 85.45%

(Iteration : 90000

Loss = 0.5059757921080623

Accuracy = 85.45454545454545%)

* In my analysis, best accuracy 85.45%
* I have provided the consequences for four values of learning\_rate and observe that when the learning\_rate is goinng to small the loss function is increasing and accuracy is decreasing whereas when I insert learning\_rate = 1e-3 with iteration 100000, it shows minimum loss function with 85.45% accuracy.

Logistic regression is very esy to use and used to simulate the binary otcome. Logistic regression convergence quickley using the gradient descent function. Logistic regration can be extended with the Lasso and Ridge.