CSE574: INTRODUCTION TO MACHINE LEARNING

ASSIGNMENT 1

Part 1: Logistic Regression

Task: To perform logistic regression and will use a logistic function to model a binomial (Binary / Bernoulli) output variable.

Logistic regression:

Logistic regression model predicts that the observation belongs to a particular category. To generate these probabilities, logistic regression uses the sigmoid function. This function maps a real number to a value between 0 and 1.

Dataset:

penguins.csv

It contains three penguin species and includes measurements of bill length, bill depth, flipper length, and body mass. Overall, we are provided with 344 data samples.

The dataset consists of 7 columns:

- **species**: penguin species (Chinstrap, Adélie, or Gentoo)
- **bill_length_mm**: culmen length (mm)
- **bill_depth_mm**: culmen depth (mm)
- **flipper_length_mm**: flipper length (mm)
- **body_mass_g**: body mass (g)
- **island**: island name (Dream, Torgersen, or Biscoe) in the Palmer Archipelago (Antarctica)
- **sex**: penguin sex (female, male)

1. ***Provide your best accuracy***

   For the given dataset, we have performed logistic regression and the best accuracy obtained by our model is 0.78 with randomly selected data.

   The weights used to obtain the best accuracy are:

species	0.004635
island	-0.003309
bill_length_mm	-0.027727
bill_depth_mm	-0.021381
flipper_length_mm	0.103353
body_mass_g	-0.080350

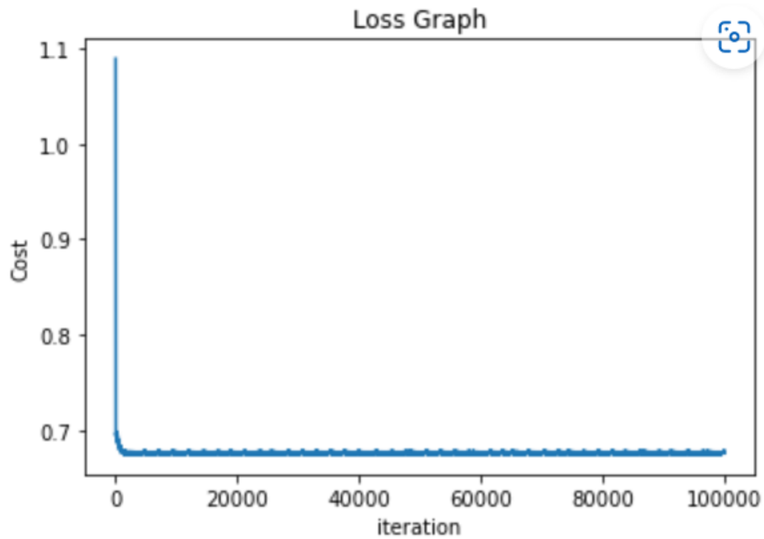
2. ***Include loss graph and provide a short analysis of the results.***

We had run our model with 3 learning rates **[0.1,0.01,0.001]** with 100000 iterations.

The loss graph obtained on each iteration is:

Learning rate = 0.1

Total iterations = 100000



The loss for each iteration has started converging with in the first 100 iterations, starting from 1.089 to 0.69. From 100^{th} iteration it slowly converged to 0.67 by the end of the process.

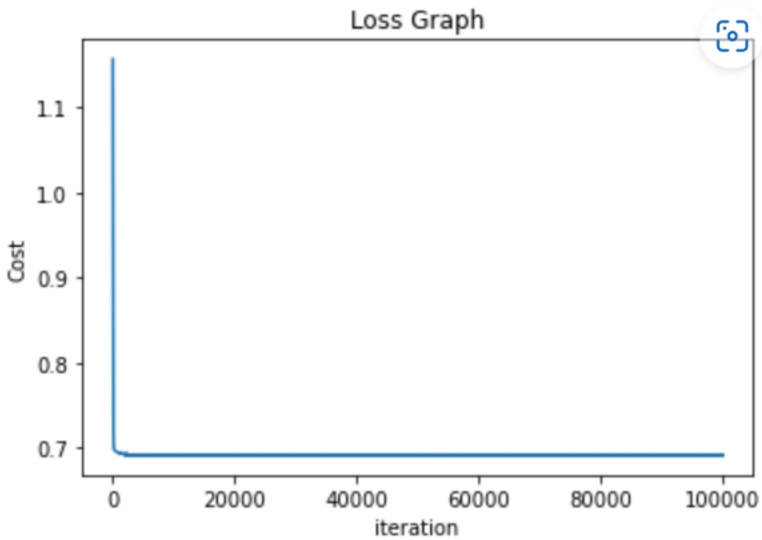
The accuracy obtained for the learning rate 0.1 is 0.72.

The weights are

species	0.053930
island	-0.004579
bill_length_mm	-0.216635
bill_depth_mm	-0.167576
flipper_length_mm	0.823427
body_mass_g	-0.612729

Learning rate = 0.01

Total iterations = 100000



The loss for each iteration has started converging with in the first 200 iterations, starting from 1.156 to 0.69. The cost got stabilized at 0.69 and continued till the end of iterations.

The accuracy obtained for the learning rate 0.01 is 0.76.

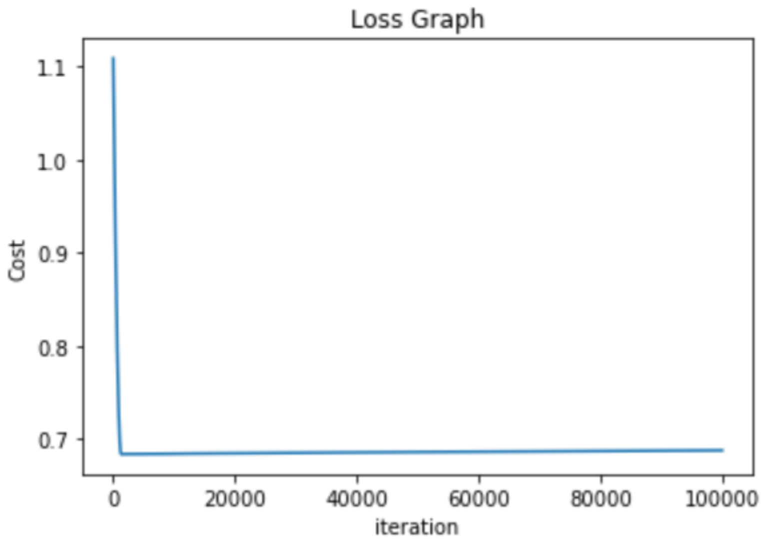
The weights are

species	0.004764
island	-0.001027
bill_length_mm	-0.024591
bill_depth_mm	-0.019910
flipper_length_mm	0.085657
body_mass_g	-0.066897

The accuracy obtained for learning rates 0.1 and 0.01 are similar. Now, we try with different learning rate 0.001.

Learning rate = 0.001

Total iterations = 100000



The loss for each iteration has started converging with in the first 1100 iterations, starting from 1.109 to 0.69. The cost then fluctuated a little between 0.68 and 0.67 and finally got stable around 0.68.

The accuracy obtained for the learning rate 0.001 is 0.826.

The weights are

species	0.022750
island	-0.004904
bill_length_mm	-0.103501
bill_depth_mm	-0.034542
flipper_length_mm	0.266279
body_mass_g	-0.205403

We have got the best accuracy for our model with the learning rate 0.001, Although the learning rate is important the weight of each feature is highly impacting the accuracy of the model.

Logistic Regression

Advantages

- It is easier to implement and can be trained efficiently.
- it can help understand the relationships between different variables and the impact of their outcomes.
- Logistic regression performs well when the data is linearly separable.
- It classifies the unknown records very fast.
- It can be easily extended to multiple classes.

Disadvantages

- If the number of features is higher than that of observations, then logistic regression will overfit the data.
- It can only predict discrete functions. Hence the label is bound to discrete number set.
- This model can only be used for linear models.
- The features must not be multicollinear.
- Independent variables used in logistic regression should be related by log odds (log(p/(1-p)))

Part 2: Linear Regression

Task: To Perform linear regression and test our model using ‘Diamonds.csv’ dataset.

Dataset:

Diamonds.csv

This Dataset contains information of 54000 different types of diamonds based on their cut, price, carat etc.

Columns:

Carat - It is the unit of measurement for the physical weight of diamonds.

Cut - The term "cut" in diamond refers to the way a diamond has been shaped and polished.

Color - The color of a diamond is another important characteristic that affects its value and appearance.

Clarity - The clarity of a diamond refers to the presence or absence of blemishes and inclusions within the stone.

Depth - The depth of a diamond refers to the height of the diamond measured from the table (top) to the culet (bottom).

Table - The table of a diamond refers to the flat, topmost facet of the diamond. It is the largest and most visible facet of the diamond when viewed from the top.

Price – The price of diamond in dollars.

X – length of diamond.

Y – width of diamond.

Z – depth of diamond.

There are around 54000 records in the data,

```
src_df = pd.read_csv('diamond.csv')
```

```
src_df.shape
```

```
(53940, 11)
```

We have 11 variables.

Features: carat, cut, color, clarity, depth, table, x, y, z

Target Variable: Price

Provide the main statistics about the entries of the dataset (mean, std, number of missing values, etc.)

Read, preprocess and print the main statistic about the dataset (your code from Part I can be reused).

```
src_df.describe()
```

5]:

	carat	depth	table	price	x	y	z
count	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000
mean	0.797940	61.749405	57.457184	3932.799722	5.731157	5.734526	3.538734
std	0.474011	1.432621	2.234491	3989.439738	1.121761	1.142135	0.705699
min	0.200000	43.000000	43.000000	326.000000	0.000000	0.000000	0.000000
25%	0.400000	61.000000	56.000000	950.000000	4.710000	4.720000	2.910000
50%	0.700000	61.800000	57.000000	2401.000000	5.700000	5.710000	3.530000
75%	1.040000	62.500000	59.000000	5324.250000	6.540000	6.540000	4.040000
max	5.010000	79.000000	95.000000	18823.000000	10.740000	58.900000	31.800000

Some of the important characteristics in our dataset are:

The average price of the diamond is $ 3932.79.

The maximum carat weight in the dataset is 5.01 carats.

The minimum depth is 43%

Correlation matrix

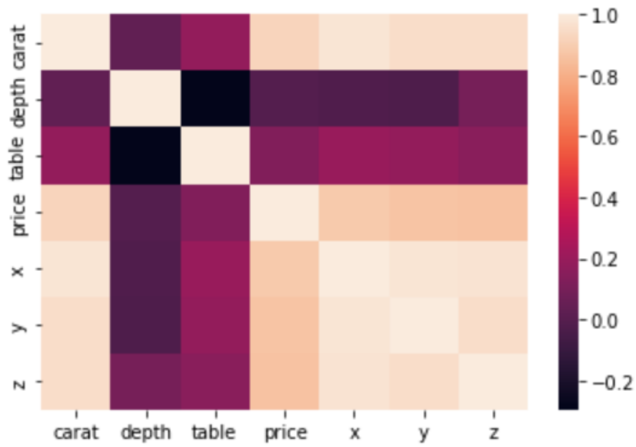
```
src_df.corr()
```

	carat	cut	color	clarity	depth	table	price	x	y	z
carat	1.000000	0.017124	0.291437	-0.214290	0.028224	0.181618	0.921591	0.975094	0.951722	0.953387
cut	0.017124	1.000000	0.000304	0.028235	-0.194249	0.150327	0.039860	0.022342	0.027572	0.002037
color	0.291437	0.000304	1.000000	-0.027795	0.047279	0.026465	0.172511	0.270287	0.263584	0.268227
clarity	-0.214290	0.028235	-0.027795	1.000000	-0.053080	-0.088223	-0.071535	-0.225721	-0.217616	-0.224263
depth	0.028224	-0.194249	0.047279	-0.053080	1.000000	-0.295779	-0.010647	-0.025289	-0.029341	0.094924
table	0.181618	0.150327	0.026465	-0.088223	-0.295779	1.000000	0.127134	0.195344	0.183760	0.150929
price	0.921591	0.039860	0.172511	-0.071535	-0.010647	0.127134	1.000000	0.884435	0.865421	0.861249
x	0.975094	0.022342	0.270287	-0.225721	-0.025289	0.195344	0.884435	1.000000	0.974701	0.970772
y	0.951722	0.027572	0.263584	-0.217616	-0.029341	0.183760	0.865421	0.974701	1.000000	0.952006
z	0.953387	0.002037	0.268227	-0.224263	0.094924	0.150929	0.861249	0.970772	0.952006	1.000000

From the correlation matrix, there is not much effect on price by features such as cut, color, clarity, depth, table. So, we dropped these columns. Heatmap for the following is,

```
## https://stackoverflow.com/questions/39409866/correlation-heatmap
sns.heatmap(src_df.corr(),
        xticklabels=src_df.corr().columns,
        yticklabels=src_df.corr().columns)
```

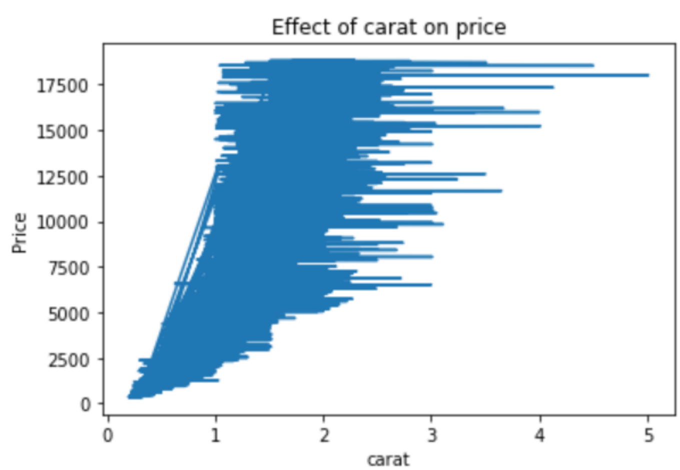
```
<AxesSubplot:>
```



<u>Checking the impact of carat on price</u>

The plot shows the relationship between the carat weight and price of diamonds in the dataset. Each point on the plot represents a single diamond, with the x-axis showing the carat weight and the y-axis showing the price in US dollars.

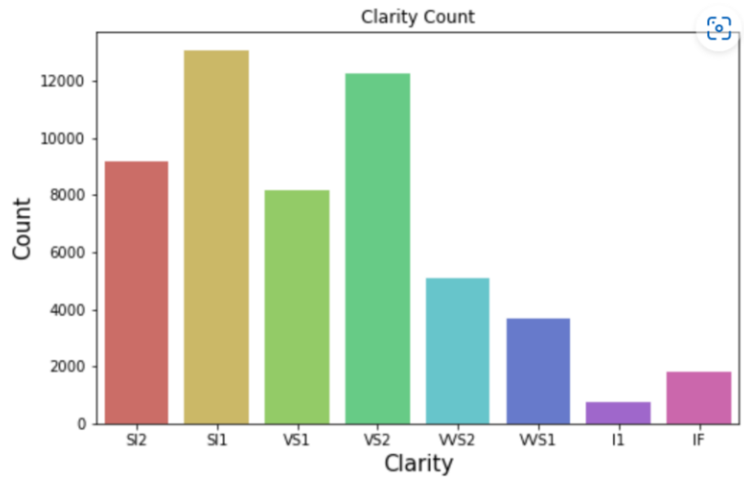
```
plt.plot(src_df['carat'],src_df['price'])
plt.title('Effect of carat on price')
plt.xlabel('carat')
plt.ylabel('Price')
plt.show()
```



creating a bar graph to count the number of filghts per airline company

count plot is used to print of seaborn library, using count plot we get counts per each kind of clarity. From the graph, the maximum diamonds are of category SI1 and least are of type I1.

```
plt.figure(figsize=(8,5))
sns.countplot(x=src_df['clarity'],palette='hls')
plt.title('Clarity Count',fontsize=12)
plt.xlabel('Clarity',fontsize=15)
plt.ylabel('Count',fontsize=15)
plt.show()
```



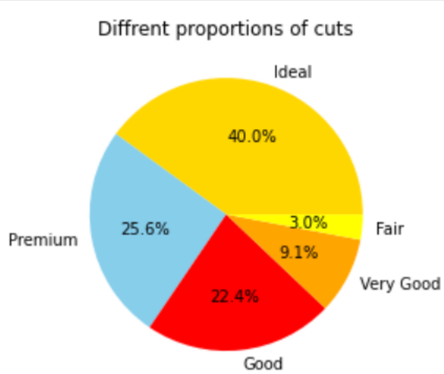
Using pie chart to find the percentage of customers travelling to a destination

We have created a pie plot to plot the different proportions of cuts available in diamonds. We have used 5 different colors to indicate 5 different cuts in our data. The ideal cut is about 40% of data and 3.0% of data is fair cut.

```
colors = ['gold', 'skyblue', 'red', 'orange','yellow']

plt.pie(src_df['cut'].value_counts(), labels=src_df['cut'].unique(), colors=colors, autopct='%1.1f%%')

plt.title('Diffrent proportions of cuts')
plt.show()
```

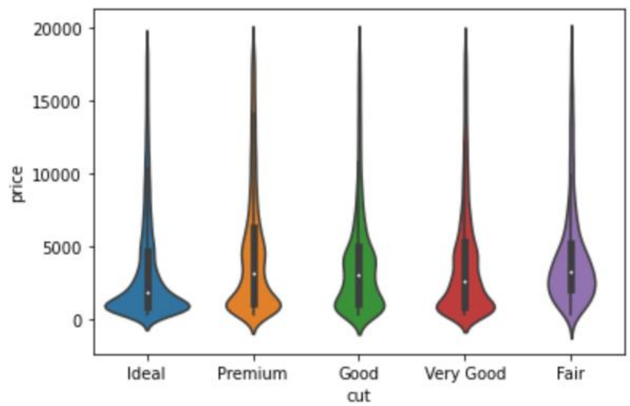


Using violin graph to compare the price of the ticket based on cut

The plot will show the distribution of diamond prices for each quality of cut, allowing for easy comparison of the central tendency, spread, and shape of the data across the different cut qualities.

```
sns.violinplot(x='cut', y='price', data=src_df)
```

```
<AxesSubplot:xlabel='cut', ylabel='price'>
```



Loss Value

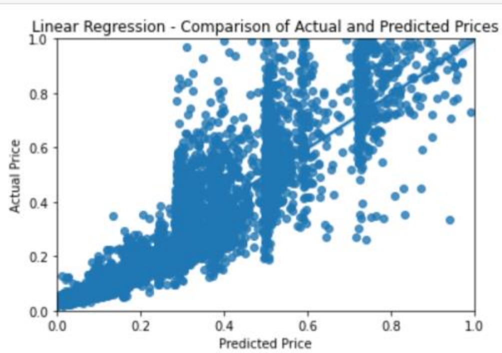
The loss value for Linear regression is calculated using MSE, and the loss value is `0.0067527169377345425`.

```
Linear_MSE = np.mean((y_test-y_pred_linear)**2)
print(Linear_MSE)
```

```
0.0067527169377345425
```

plot comparing the predictions vs the actual test data

```
sns.regplot(x=y_pred_linear, y=y_test);
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.xlabel('Predicted Price')
plt.ylabel('Actual Price')
plt.title('Linear Regression - Comparison of Actual and Predicted Prices')
plt.show()
```



The weights are, 2.434349, -0.550315,1.962669, -0.686875

```
linreg.weights
```

```
0    2.434349
1   -0.550315
2    1.962669
3   -0.686875
dtype: float64
```

<u>Benefits/Drawbacks of using OLS estimate for computing the weights</u>

<u>Benefits</u>

- Ordinary Least Square is a statistical method used to produce one straight line that minimizes the total squared error.
- OLS provides minimum variance mean unbiased estimation when the errors have finite variance.
- When the errors are normally distributed, OLS is the maximum likelihood estimator.
- OLS method is simple, and computation is easy.

<u>Drawbacks</u>

- Sometimes, it performs poorly when the dataset has single independent variables and multiple dependent variables sets.
- It might also perform very poor when some points in the training data have numerous numbers of small or large values present in it.
- To get reliable results the dataset must be very large.

<u>Benefits/Drawbacks of using a Linear Regression model</u>

<u>Benefits</u>

- Linear Regression is simple to implement, and it can produce reliable results.
- It perfectly fits linearly separable data and can be used to find the relationship between the variables.
- It provides interpretable and understanding results, which can be used in variety of fields.
- It can be used for both predictive and explanatory purposes.

<u>Drawbacks</u>

- When data has noise or outlier, then it tends to overfit which can't be controlled.
- It is irrelevant if the data is having non-linear tendencies.
- It is unstable in presence of correlated input attributes
- It gets confused by unnecessary attributes.
- It is influenced by the outliers.

Part 3: Ridge Regression

Loss Value

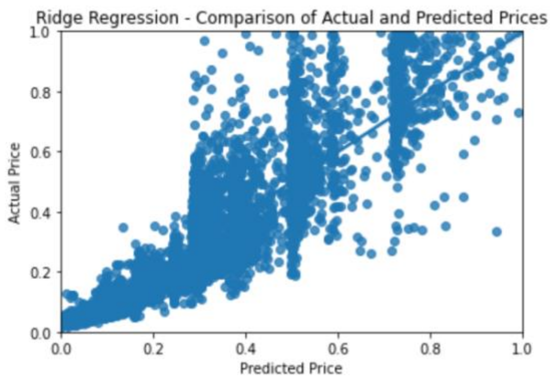
The loss value for Ridge regression is calculated using Squared Loss, and the loss value is `0.8430526280732467`.

```
Ridge_MSE = np.mean((y_test-y_pred_linear)**2)
penalty = alpha * np.dot(ridgreg.weights.T,ridgreg.weights)
Sqrd_loss = Ridge_MSE + penalty
print(Sqrd_loss)
```

```
0.8430526280732467
```

plot comparing the predictions vs the actual test data

```
sns.regplot(x=y_pred_ridge, y=y_test);
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.xlabel('Predicted Price')
plt.ylabel('Actual Price')
plt.title('Ridge Regression - Comparison of Actual and Predicted Prices')
plt.show()
```



The weights obtained are 2.431375, -0.451947,1.361170, -0.627995

```
ridgreg.weights
```

```
0    2.431375
1   -0.451947
2    1.361170
3   -0.627995
dtype: float64
```

Discuss the difference between Linear and Ridge regressions. What is the main motivation for using l2 regularization?

Linear regression	Ridge regression
It creates a line that best fits the relationship between dependent and independent variable.	It is a technique that used to analyze data that suffers from multicollinearity.
The main goal of linear regression is to minimize the mean squared error.	It involves additional penalty term that helps in shrinkage of the coefficients towards zero
Coefficients are estimated using ordinary least squares	The penalty term is proportional to the square of the magnitude of the coefficients

Main Motivation of L2 regularization

- L2 Regularization is used to prevent overfitting in the model.
- Overfitting is avoided by including the penalty term to the model.

Discuss the benefits/drawbacks of using a Ridge Regression model.

Benefits

- It performs well when there is a large multivariate data with the number of predictors larger than the number of observations.
- When there is multicollinearity, the Ridge estimator is preferentially effective at enhancing the least-squares estimate.
- It Prevents a model from overfitting
- The correct number of biases should be added to estimates to make it relatively credible approximations of genuine population values.

Drawbacks

- It can be sensitive to choose of the regularization parameter.
- It introduces another hyperparameter that needs to be tuned, this parameter controls the penalty in the model.
- The coefficients that produced by ridge regression models are biased.
- It may not perform well when the relation between the independent and dependent variable is complex.

Contributions

Team Member	Assignment Part	Contribution (%)
saitejad	Part 1	50
vvudhaya	Part 1	50
saitejad	Part 2	50
vvudhaya	Part 2	50
saitejad	Part 3	50
vvudhaya	Part 3	50

References

https://stackoverflow.com/questions/24147278/how-do-i-create-test-and-train-samples-from-one-dataframe-with-pandas

https://stackoverflow.com/questions/43777243/how-to-split-a-dataframe-in-pandas-in-predefined-percentages

https://www.geeksforgeeks.org/data-normalization-with-pandas/

https://wiki.python.org/moin/UsingPickle

https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/

https://www.engati.com/glossary/ridge-regression

https://stackoverflow.com/questions/39409866/correlation-heatmap