Mohd Zain 22MT0214 Logistic Regression

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from google.colab import files
al=files.upload()
```

- **Iris.csv**(text/csv) - 5107 bytes, last modified: 19/09/2019 - 100% done

```
Saving Iris.csv to Iris.csv
```

```
import io
df=pd.read_csv(io.BytesIO(al['Iris.csv']))
```

```
df.head(3)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa

```
df.tail(3)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica

```
df = df.drop(columns = ['Id'])
df.head(5)
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
df['Species'].value_counts()
```

```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

```
df.corr()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	1.000000	-0.109369	0.871754	0.817954
SepalWidthCm	-0.109369	1.000000	-0.420516	-0.356544
PetalLengthCm	0.871754	-0.420516	1.000000	0.962757
PetalWidthCm	0.817954	-0.356544	0.962757	1.000000

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Species'] = le.fit_transform(df['Species'])
df.head(100)
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
95	5.7	3.0	4.2	1.2	1
96	5.7	2.9	4.2	1.3	1
97	6.2	2.9	4.3	1.3	1
98	5.1	2.5	3.0	1.1	1
99	5.7	2.8	4.1	1.3	1

100 rows × 5 columns

```
from sklearn.model_selection import train_test_split
X = df.drop(columns = ['Species'])
Y = df['Species']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3)
```

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
```

```
model.fit(X_train, Y_train)
```

```
LogisticRegression()
```

```
Y_pred=model.predict(X_test)
Y_pred
```

```
array([0, 1, 2, 1, 0, 1, 0, 1, 0, 0, 0, 2, 0, 1, 1, 2, 1, 2, 0, 0, 1, 0,
       1, 1, 0, 0, 2, 0, 2, 1, 2, 1, 1, 1, 1, 2, 1, 0, 0, 1, 2, 2, 1, 1,
       1])
```

```
from sklearn.metrics import r2_score
```

```
acc=r2_score(Y_test,Y_pred)
acc*100
```

```
92.53731343283582
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test, Y_pred)
print(cm)
```

```
[[15  0  0]
 [ 0 18  0]
 [ 0  2 10]]
```

```
import seaborn as sns
ax=plt.axes()
df_cm=cm
sns.heatmap(df_cm, annot=True, annot_kws={"size": 30}, fmt='d',cmap="Blues", ax =
ax.set_title('Confusion Matrix')
plt.show()
```