

```
In [24]: ##### Scale data using StandardScaler #####
import functions as fns
from sklearn.datasets import load_iris
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
iris = load_iris()
```

```
In [25]: # scale_features_using_standard_scaler
data_scaled = fns.scale_features_using_min_max_scaler(iris.data)
```

```
In [26]: # Display DataFrame
df_data = fns.create_dataframe(data_scaled, iris.feature_names)
df_data.head(10)
```

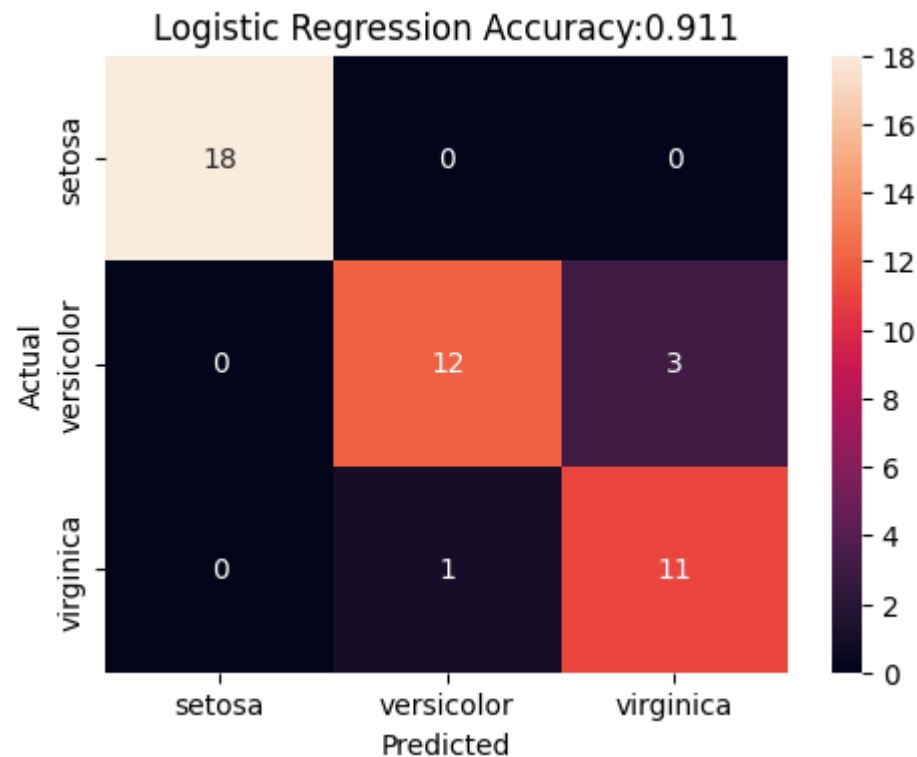
```
Out[26]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	0.222222	0.625000	0.067797	0.041667
1	0.166667	0.416667	0.067797	0.041667
2	0.111111	0.500000	0.050847	0.041667
3	0.083333	0.458333	0.084746	0.041667
4	0.194444	0.666667	0.067797	0.041667
5	0.305556	0.791667	0.118644	0.125000
6	0.083333	0.583333	0.067797	0.083333
7	0.194444	0.583333	0.084746	0.041667
8	0.027778	0.375000	0.067797	0.041667
9	0.166667	0.458333	0.084746	0.000000

```
In [27]: # Split the data into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data_scaled, iris.target, test_size=0.3)
```

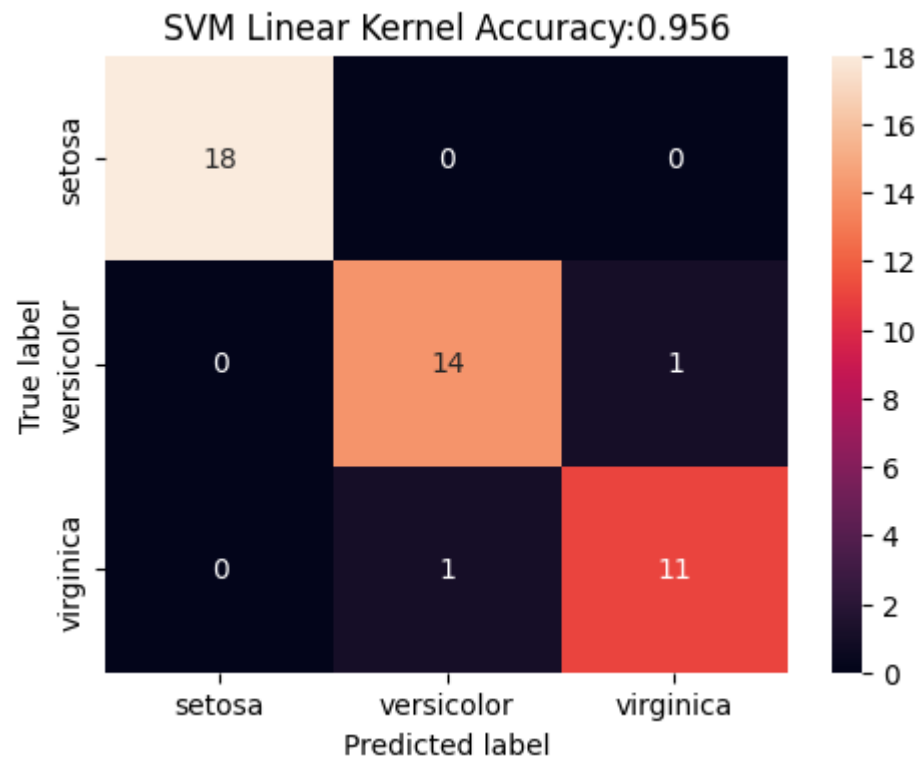
```
In [28]: # train_model_using_logistic_regression
cm, accuracyLR, precision, recall, f1 = fns.train_model_using_logistic_regression(X_train, X_test, y_train, y_test,
```

```
accuracy_LR : 0.911
precision_LR : 0.911
recall_LR : 0.911
f1-score_LR : 0.911
```



```
In [29]: # train_model_using_support_vector_classification
cm, accuracySVC, precision, recall, f1 = fns.train_model_using_support_vector_classification(X_train, X_test, y_tra
```

accuracy\_SVC : 0.956  
precision\_SVC : 0.956  
recall\_SVC : 0.956  
f1-score\_SVC : 0.956



```
In [30]: # compare_logistic_regression_vs_vector_classification_accuracy
fns.compare_logistic_regression_vs_vector_classification_accuracy(accuracyLR, accuracySVC);
```

	Model	Accuracy_score
0	Support Vector Classification	0.955556
1	Logistic Regression	0.911111

In [ ]: