In [85]:

```
########## Non Scaled Data ##########
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 [86]:

```
# scale_features_using_standard_scaler
data_not_scaled = iris.data
```

In [87]:

```
# Display DataFrame
df_data = fns.create_dataframe(data_not_scaled, iris.feature_names)
df_data.head(10)
```

Out[87]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
5	5.4	3.9	1.7	0.4
6	4.6	3.4	1.4	0.3
7	5.0	3.4	1.5	0.2
8	4.4	2.9	1.4	0.2
9	4.9	3.1	1.5	0.1

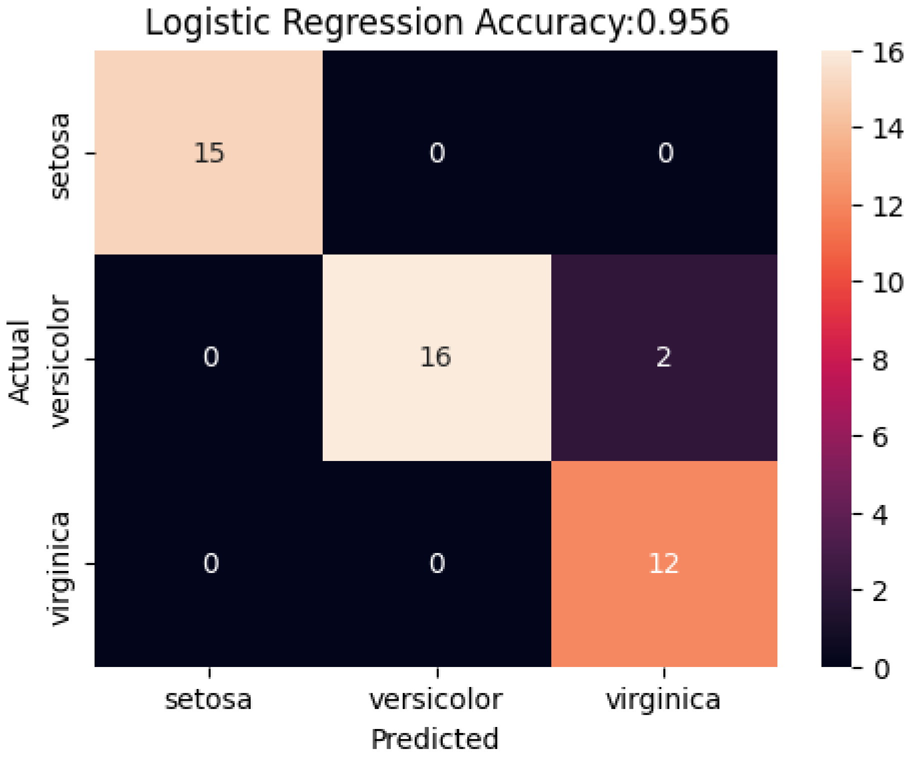
In [88]:

```
# 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_not_scaled, iris.target, test_size=0.3)
```

In [89]:

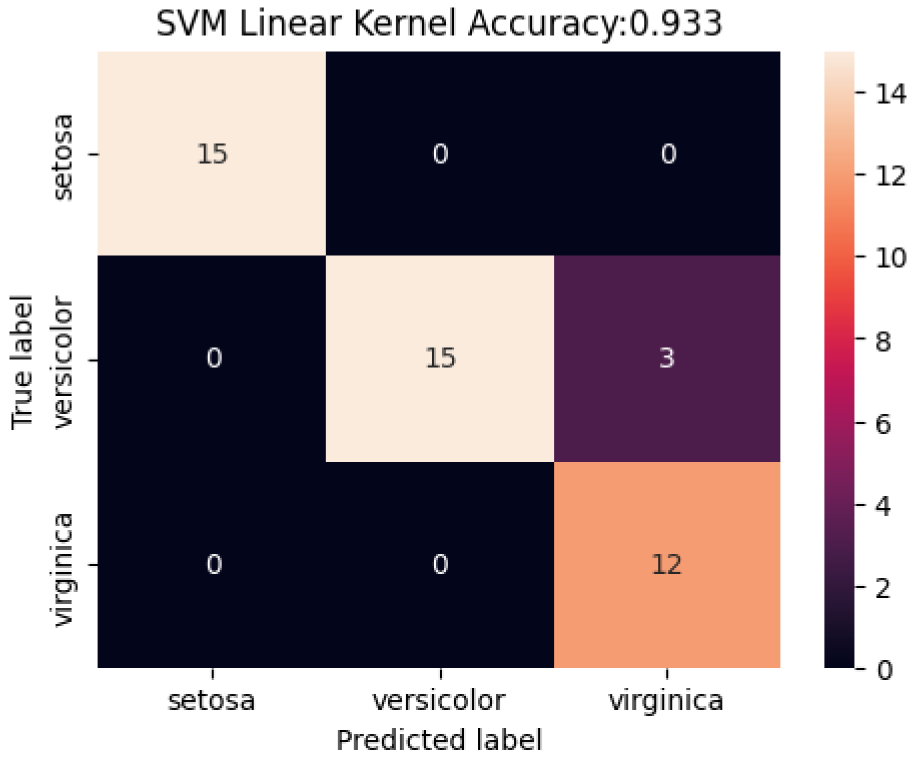
```
# train_model_using_logistic_regression
cm, accuracyLR, precision, recall, f1 = fns.train_model_using_logistic_regression(X_train, X_test, y_train, y_test,
```

```
accuracy_LR : 0.956
precision_LR : 0.956
recall_LR : 0.956
f1-score_LR : 0.956
```



In [90]:

```
# 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 : 0.933
precision_SVC : 0.933
recall_SVC : 0.933
f1-score_SVC : 0.933
```



```
In [91]: # compare_logistic_regression_vs_vector_classification_accuracy
         fns.compare_logistic_regression_vs_vector_classification_accuracy(accuracyLR, accuracySVC);
```

```
                            Model  Accuracy_score
0             Logistic Regression        0.955556
1   Support Vector Classification        0.933333
```

```
In [ ]:
```