Week-12

Performance analysis of classification models on a specific dataset

In [2]:
```
import pandas as pd, numpy as np, matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.model_selection import train_test_split as tts
from sklearn.metrics import *
```

In [4]:
```
wine_data = datasets.load_wine()
wine_data.keys()
```

Out[4]:
```
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names'])
```

In [10]:
```
wine_df = pd.DataFrame(wine_data.data, columns=wine_data['feature_names'])
wine_df['target'] = wine_data.target
```

In [19]:
```
x_train, x_test, y_train, y_test = tts(wine_df.iloc[:, :-1], wine_df.iloc[:, -1], test_size=0.2, random_state=42)
```

Model-1:SVM

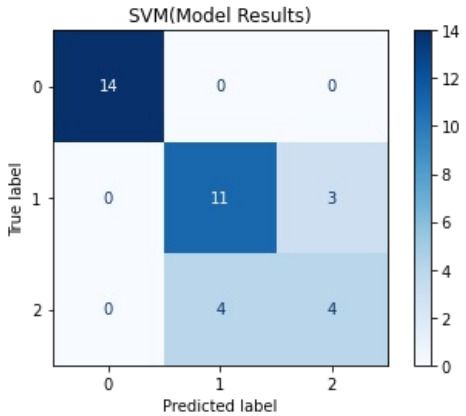
In [25]:
```
from sklearn import svm
model_1 = svm.SVC()
model_1.fit(x_train, y_train)
```

Out[25]:
```
SVC()
```

In [26]:
```
y_pred_1 = model_1.predict(x_test)
```

In [34]:
```
print(f"Accuracy of the Model: {accuracy_score(y_test, y_pred_1):.2f}")
cm = confusion_matrix(y_test, y_pred_1)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=np.unique(y_test))
disp.plot(cmap=plt.cm.Blues)
plt.title('SVM(Model Results)')
plt.show()
```

```
Accuracy of the Model: 0.81
```



Model-2: KNN

In [36]:
```
from sklearn.neighbors import KNeighborsClassifier as KNC
k = int(np.ceil(np.sqrt(len(y_test))))
```

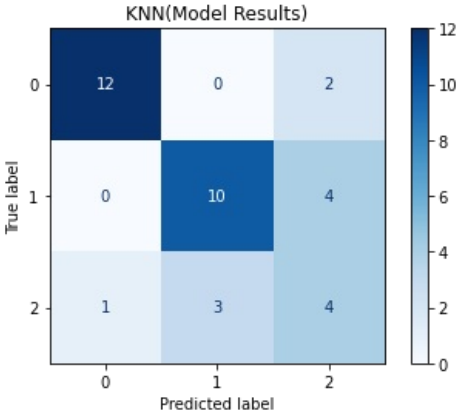
In [37]:
```
neigh = KNC(n_neighbors=k)
neigh.fit(x_train, y_train)
```

Out[37]:
```
KNeighborsClassifier(n_neighbors=6)
```

In [38]:
```
y_pred_2 = neigh.predict(x_test)
```

```
print(f"Accuracy of the Model: {accuracy_score(y_test, y_pred_2):.2f}")
cm = confusion_matrix(y_test, y_pred_2)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=np.unique(y_test))
disp.plot(cmap=plt.cm.Blues)
plt.title('KNN(Model Results)')
plt.show()
```

```
Accuracy of the Model: 0.72
```



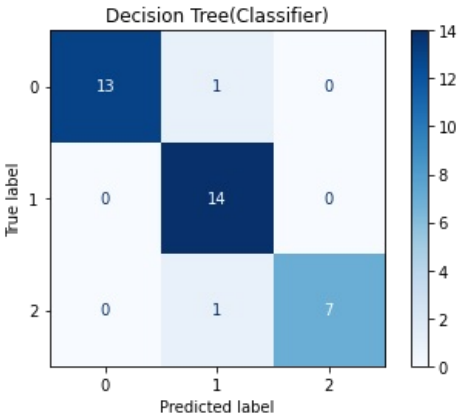
Model-3: Decision Tree Classifier

In [39]:
```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(random_state=0)
clf.fit(x_train, y_train)
```

Out[39]:
```
DecisionTreeClassifier(random_state=0)
```

In [40]:
```
y_pred_3 = clf.predict(x_test)
print(f"Accuracy of the Model: {accuracy_score(y_test, y_pred_3):.2f}")
cm = confusion_matrix(y_test, y_pred_3)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=np.unique(y_test))
disp.plot(cmap=plt.cm.Blues)
plt.title('Decision Tree(Classifier)')
plt.show()
```

```
Accuracy of the Model: 0.94
```



Model-04: Gauss Naives

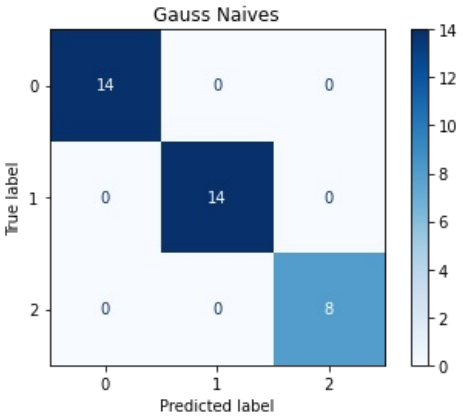
In [41]:
```
from sklearn.naive_bayes import GaussianNB as GNB
model_GNB = GNB()
model_GNB.fit(x_train, y_train)
```

Out[41]:
```
GaussianNB()
```

In [45]:
```
y_pred_4 = model_GNB.predict(x_test)
print(f"Accuracy of the Model: {accuracy_score(y_test, y_pred_4):.2f}")
cm = confusion_matrix(y_test, y_pred_4)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=np.unique(y_test))
```

```
disp.plot(cmap=plt.cm.Blues)
plt.title('Gauss Naives')
plt.show()
```

```
Accuracy of the Model: 1.00
```



In [64]:

```
plt.style.use("ggplot")
from sklearn.metrics import accuracy_score as AS
models = 'SVM KNN Decision_Tree Gauss_Naives'.split(" ")
val = np.array([AS(y_test, y_pred_1).round(2), AS(y_test, y_pred_2).round(2), AS(y_test, y_pred_3).round(2), AS(y
plt.figure(figsize=(10, 6))
plt.bar(models, val, color='skyblue')
plt.xlabel("Models")
plt.ylabel("Accuracy(%)")
plt.show()
```