The columns which I selected for prediction:

```
x = df[["Sex","Age","SibSp","Parch","Embarked_C","Embarked_Q","Embarked_S","Pclass_1","Pclass_2","Pclass_3"]]
y = df[["Survived"]]
```

The training and testing split:

```
y = df[[ Survived ]]
X_train, X_test, Y_train, Y_test = train_test_split(x, y, random_state=0)
# Fit (train) the Support Vector Machine classifier
```

The SVM models building:

```
poly_svm_clf = SVC(kernel="poly")
poly_svm_model = poly_svm_clf.fit(X_train, Y_train)
poly_svm_prediction = poly_svm_clf.predict(X_test)

print("Accuracy {0:.2f}%".format(100*accuracy_score(poly_svm_prediction, Y_test)))

print(confusion_matrix(poly_svm_prediction, Y_test))
```

```
# Fit (train) the Support Vector Machine classifier
linear_svm_clf = SVC(kernel = "linear")
linear_svm_model = linear_svm_clf.fit(X_train, Y_train)
linear_svm_prediction = linear_svm_clf.predict(X_test)

print("Accuracy {0:.2f}%".format(100*accuracy_score(linear_svm_prediction, Y_test)))

print(confusion_matrix(linear_svm_prediction, Y_test))
```

```
# Fit (train) the Support Vector Machine classifier
rbf_svm_clf = SVC(kernel = "rbf")
rbf_svm_model = rbf_svm_clf.fit(X_train, Y_train)
rbf_svm_prediction = rbf_svm_clf.predict(X_test)

print("Accuracy {0:.2f}%".format(100*accuracy_score(rbf_svm_prediction, Y_test)))

print(confusion_matrix(rbf_svm_prediction, Y_test))
```

```
# Fit (train) the Support Vector Machine classifier
sigmoid_svm_clf = SVC(kernel = "sigmoid")
sigmoid_svm_model = sigmoid_svm_clf.fit(X_train, Y_train)
sigmoid_svm_prediction = sigmoid_svm_clf.predict(X_test)

print("Accuracy {0:.2f}%".format(100*accuracy_score(sigmoid_svm_prediction, Y_test)))

print(confusion_matrix(sigmoid_svm_prediction, Y_test))
```

The accuracy for each SVC with different kernels:

```
# Fit (train) the Support Vector Machine classifier
poly_svm_clf = SVC(kernel="poly")
poly_svm_model = poly_svm_clf.fit(X_train, Y_train)
poly_svm_prediction = poly_svm_clf.predict(X_test)

print("Accuracy {0:.2f}%".format(100*accuracy_score(poly_svm_prediction, Y_test)))

print(confusion_matrix(poly_svm_prediction, Y_test))
```

9] ✓ 0.0s

```
Accuracy 65.02%
[[139  78]
 [  0   6]]
```

```
# Fit (train) the Support Vector Machine classifier
linear_svm_clf = SVC(kernel = "linear")
linear_svm_model = linear_svm_clf.fit(X_train, Y_train)
linear_svm_prediction = linear_svm_clf.predict(X_test)

print("Accuracy {0:.2f}%".format(100*accuracy_score(linear_svm_prediction, Y_test)))

print(confusion_matrix(linear_svm_prediction, Y_test))
```

[10] ✓ 0.0s

```
Accuracy 78.03%
[[115  25]
 [ 24  59]]
```

```
# Fit (train) the Support Vector Machine classifier
rbf_svm_clf = SVC(kernel = "rbf")
rbf_svm_model = rbf_svm_clf.fit(X_train, Y_train)
rbf_svm_prediction = rbf_svm_clf.predict(X_test)

print("Accuracy {0:.2f}%".format(100*accuracy_score(rbf_svm_prediction, Y_test)))

print(confusion_matrix(rbf_svm_prediction, Y_test))
```

✓ 0.0s

```
Accuracy 65.47%
[[137  75]
 [  2   9]]
```

```
# Fit (train) the Support Vector Machine classifier
sigmoid_svm_clf = SVC(kernel = "sigmoid")
sigmoid_svm_model = sigmoid_svm_clf.fit(X_train, Y_train)
sigmoid_svm_prediction = sigmoid_svm_clf.predict(X_test)

print("Accuracy {0:.2f}%".format(100*accuracy_score(sigmoid_svm_prediction, Y_test)))

print(confusion_matrix(sigmoid_svm_prediction, Y_test))
```

✓ 0.0s

```
Accuracy 60.54%
[[101  50]
 [ 38  34]]
```