

SVM

SVM classification

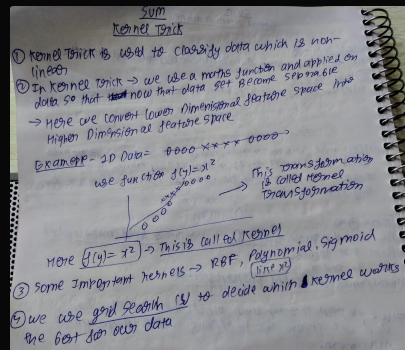
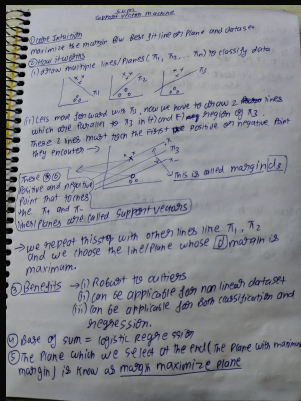
Videos

[video](#)

Maths -> skipped

[video](#)

[video](#)



svm 1.jpg

Geometric Intuition

code

```
from sklearn.svm import SVC
```

```
svm_classifier = SVC(kernel='linear', C=1.0,  
random_state=42)
```

```
svm_classifier.fit(X_train, y_train)
```

Kernel Trick SVM

Geometric Intuition

code

```
from sklearn.svm import SVC  
from sklearn.model_selection import GridSearchCV
```

```
svm = SVC()
```

```
param_grid = {  
    'kernel': ['linear', 'rbf', 'poly'],  
    'C': [0.1, 1, 10, 100],  
    'gamma': [0.1, 0.5, 1, 2, 5]  
}
```

```
grid_search = GridSearchCV(svm, param_grid, cv=5)  
grid_search.fit(X_train, y_train)
```

```
best_params = grid_search.best_params_  
best_score = grid_search.best_score_
```

```
print(f"Best parameters: {best_params}")  
print(f"Best cross-validation score: {best_score:.2f}")
```

Evaluate the model on the test set using
the best parameters:

```
best_svm = SVC(**best_params)  
best_svm.fit(X_train, y_train)
```

```
y_pred = best_svm.predict(X_test)  
test_accuracy = accuracy_score(y_test, y_pred)
```

```
print(f"Test accuracy with best parameters: {test_accuracy:.2f}")
```

SVM Regression -> not Used very much

code

```
from sklearn.svm import SVR
```

```
svr = SVR(kernel='rbf', C=100, gamma=0.1,  
epsilon=.1)
```

```
svr.fit(X_train, y_train)
```