SVM Solution_with_visualization

January 27, 2026

[18]:
```
# Import statements
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
import pandas as pd
import numpy as np
```

[19]:
```
# Read the data.
data = np.asarray(pd.read_csv('data.csv', header=None))
# Assign the features to the variable X, and the labels to the variable y.
X = data[:,0:2]
y = data[:,2]
```

[20]:
```
# TODO: Create the model and assign it to the variable model.
# Find the right parameters for this model to achieve 100% accuracy on the␣
 ↪dataset.
model = SVC(kernel='rbf', gamma=27)

# TODO: Fit the model.
model.fit(X,y)

# TODO: Make predictions. Store them in the variable y_pred.
y_pred = model.predict(X)

# TODO: Calculate the accuracy and assign it to the variable acc.
acc = accuracy_score(y, y_pred)
```

[21]:
```
# TODO: Make predictions. Store them in the variable y_pred.
y_pred = model.predict(X)

# TODO: Calculate the accuracy and assign it to the variable acc.
acc = accuracy_score(y, y_pred)
print(f"Accuracy: {acc*100:.2f}%")
```

```
Accuracy: 100.00%
```

[22]:
```
from matplotlib.colors import ListedColormap

# Colormap custom: fundal deschis + verde
```

```
cmap_light = ListedColormap(["#eaffea", "#7fc97f"])

plt.figure(figsize=(7, 6))
plt.contourf(xx, yy, Z, alpha=0.6, cmap=cmap_light)

plt.scatter(X[:, 0], X[:, 1], c=y, edgecolor="k",␣
 ↪cmap=ListedColormap(["#2c7bb6", "#1b7837"]))

if hasattr(model, "support_vectors_"):
    sv = model.support_vectors_
    plt.scatter(sv[:, 0], sv[:, 1], s=90, facecolors="none", edgecolor="k")

plt.title("SVC decision boundary (kernel = rbf)")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.xlim(0, 1)
plt.ylim(0, 1)
plt.show()
```