

```

In [13]: import numpy as np
import matplotlib.pyplot as plt
from sklearn.svm import SVC

# Support für Datum und Zeit
import time

# Definierte Datenpunkte und Labels: y=+1: A bis E, und y=-1: F bis J
X = np.array([[1, 1],[-1, 1],[3, 2],[-2, -2],[3, 1],[2, 1],[1, -2],[-2, 1],[-1, -4],[1, 3]]) # Datebpunkte
y = np.array([1, 1, 1, 1, 1, -1, -1, -1, -1, -1]) # Labels: +1 für A bis E u
nd -1 für F bis J

# Erstelle das SVM-Modell mit einem polynomialen Kernel (Grad 15)
svm_model = SVC(kernel='poly', degree=20, coef0=1) # coef0=1 entspricht dem
konstanten Term in K
svm_model.fit(X, y)

# Bereich für x1 und x2 (damit die Linie in einem sinnvollen Bereich dargeste
llt wird)
x1_range = np.linspace(-3, 3, 100)
x2_range = np.linspace(-3, 3, 100)
xx1, xx2 = np.meshgrid(x1_range, x2_range)
grid = np.c_[xx1.ravel(), xx2.ravel()]

# Vorhersagen für das Gitter
Z = svm_model.predict(grid)
Z = Z.reshape(xx1.shape)

# Plot der Datenpunkte und der Entscheidungsfläche
plt.figure(figsize=(8, 6))

# Plot der Entscheidungsebene
plt.contourf(xx1, xx2, Z, alpha=0.3, cmap='coolwarm', levels=np.arange(-1.5,
1.5, 1))

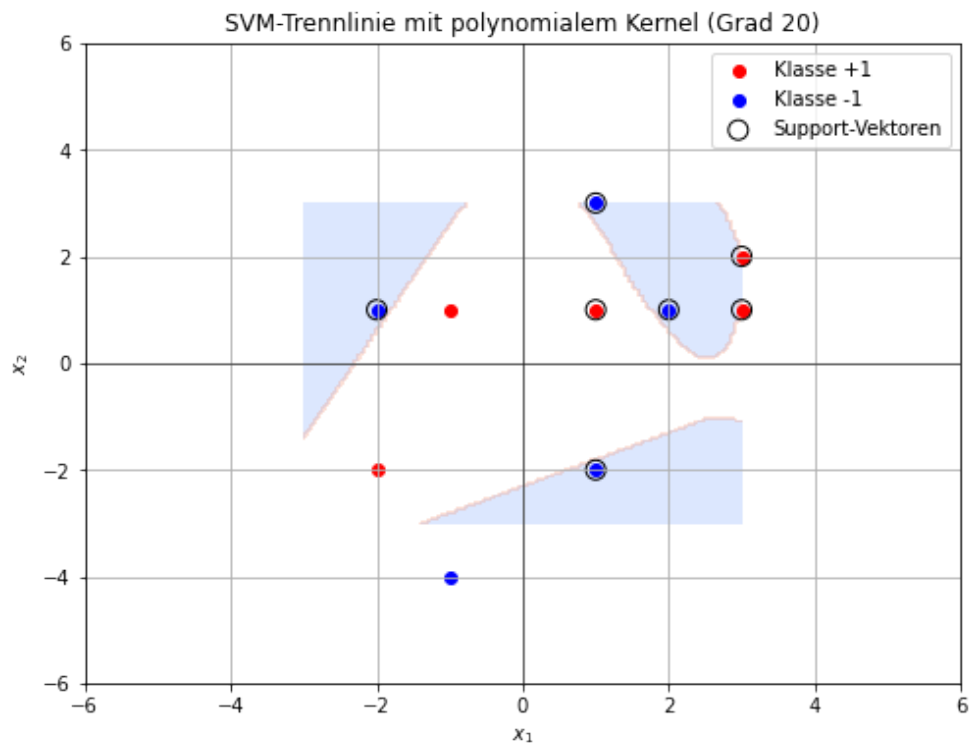
# Plot der Datenpunkte
plt.scatter(X[y == 1, 0], X[y == 1, 1], color='red', label='Klasse +1')
plt.scatter(X[y == -1, 0], X[y == -1, 1], color='blue', label='Klasse -1')

# Plot der Support-Vektoren
plt.scatter(svm_model.support_vectors_[0], svm_model.support_vectors_[1],
s=100, facecolors='none', edgecolors='k', label='Support-Vektore
n')

# Achsenbeschriftungen und Titel
plt.xlim(-6, 6)
plt.ylim(-6, 6)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.xlabel('$x_1$')
plt.ylabel('$x_2$')
plt.title('SVM-Trennlinie mit polynomialen Kernel (Grad 20)')
plt.legend()
plt.grid(True)
plt.show()

# Ausgabe der Support-Vektoren
print('Dies sind die Support-Vektoren:')
print(svm_model.support_vectors_)

```



Dies sind die Support-Vektoren:

```
[[ 2.  1.]
 [ 1. -2.]
 [-2.  1.]
 [ 1.  3.]
 [ 1.  1.]
 [ 3.  2.]
 [ 3.  1.]]
```

```
In [14]: # print current date and time

print("Date & Time:",time.strftime("%d.%m.%Y  %H:%M:%S"))
# end of import test
print ("*** End of SVM_Trennlinie-PK(20)-10DP Jupyter Notebook ***")
```

Date & Time: 18.10.2024 20:29:12

*** End of SVM_Trennlinie-PK(20)-10DP Jupyter Notebook ***