Task#1

Solution:

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import matplotlib.pyplot as plt
# Load dataset
data = pd.read_csv(r"C:\Users\DELL\Downloads\svm_dataset.csv")
print("Columns:", data.columns)
X = data.iloc[:, 0:2].values   #  first 2 columns are features
y = data.iloc[:, -1].values    #  last column is target
X_train, X_test, y_train, y_test = train_test_split( *arrays: X, y, test_size=0.3, random_state=42)
model = SVC(kernel='linear')
model.fit(X_train, y_train)
print("Accuracy:", model.score(X_test, y_test))
y_pred = model.predict(X_test)
plt.scatter(X[:, 0], X[:, 1], c=y, cmap='coolwarm', alpha=0.7)
for i, (x1, x2) in enumerate(X_test):
    plt.text(x1 + 0.05, x2 + 0.05,  s: f'{y_pred[i]}', fontsize=9, color='black')
ax = plt.gca()
xlim = ax.get_xlim()
```

```python
ylim = ax.get_ylim()
xx = np.linspace(xlim[0], xlim[1])
yy = np.linspace(ylim[0], ylim[1])
YY, XX = np.meshgrid( *xi: yy, xx)
xy = np.vstack([XX.ravel(), YY.ravel()]).T
Z = model.decision_function(xy).reshape(XX.shape)
ax.contour( *args: XX, YY, Z, colors='k', levels=[-1, 0, 1], linestyles=['--', '-', '--'])
plt.title('SVM Decision Boundary & Predictions')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.show()
```

Output:

```
C:\Users\DELL\PycharmProjects\PythonProject\.venv\Scripts\python.exe C:\Users\DELL\PycharmPro
Columns: Index(['ID', 'Feature1 (X1)', 'Feature2 (X2)', 'Label (Y)'], dtype='object')
Accuracy: 1.0
```

SVM Decision Boundary & Predictions