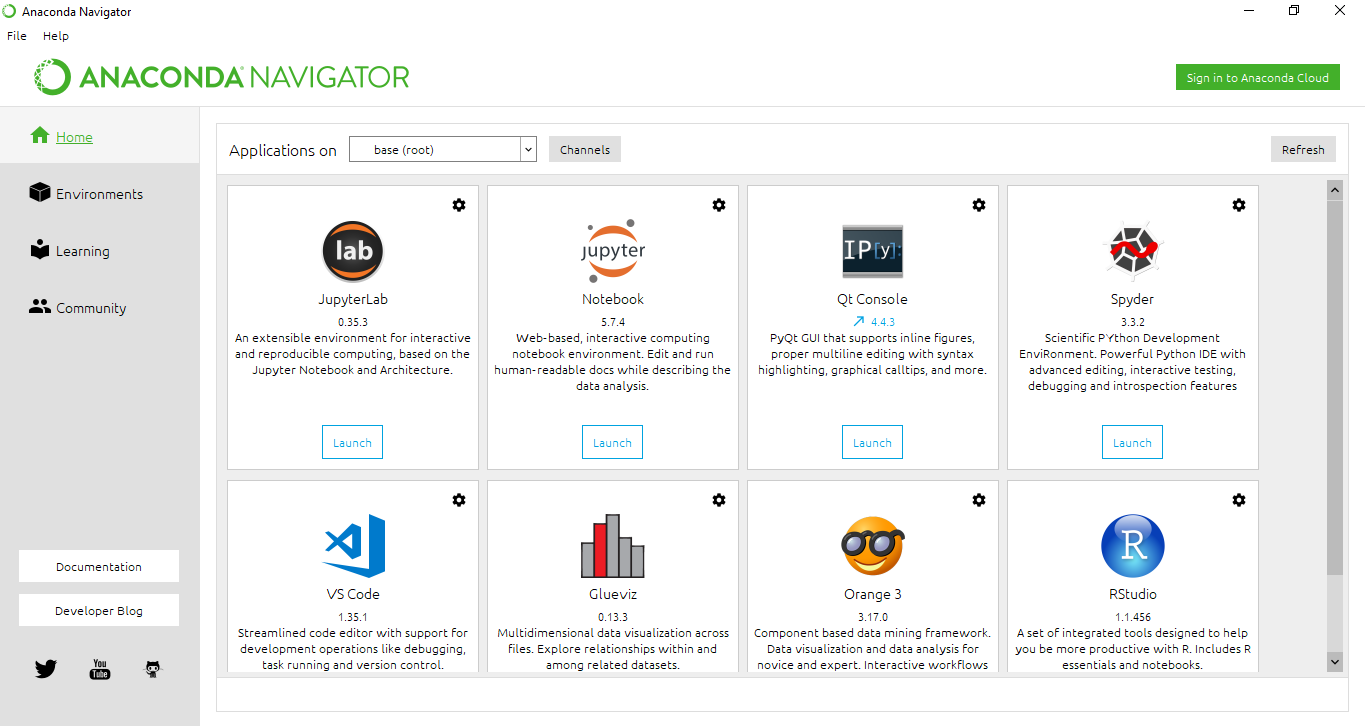
**Support Vector Machine (SVM) Classification using Anaconda Navigator (Jupyter Notebook)**

1. In this worksheet, we will be working on Connecting Oracle Server from Tableau.
2. Before following below steps, make sure that **Anaconda Navigator** is installed on your laptop. For installation, please refer the following link.

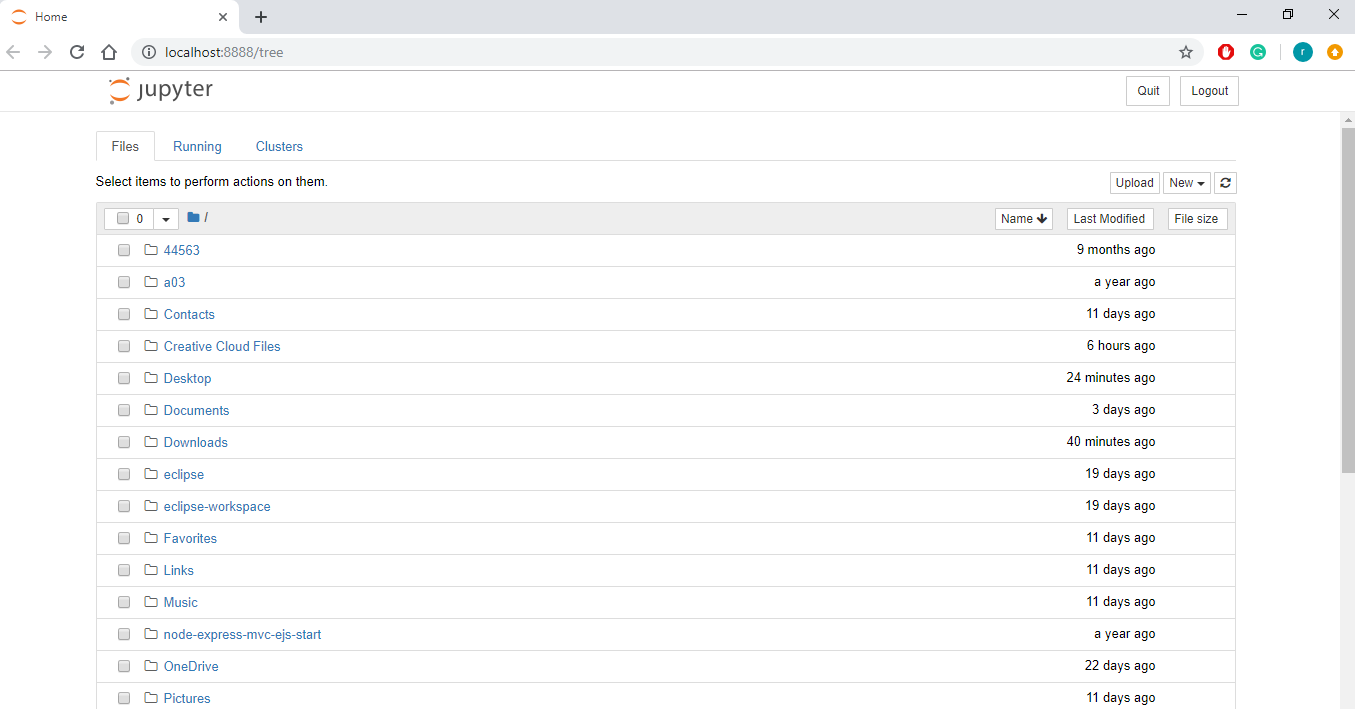
<https://docs.anaconda.com/anaconda/install/windows/>

1. If it is already installed or after installing by following above link, open the Anaconda Navigator application.



1. Now, launch the Jupyter notebook and you can see the following link in your browser.

http://localhost:8888/?token=deb5f868599589bfe4df234ac73dd7095ffc31e05ba1ae47



**SVM Classification**

**Objective:** To predict whether the given input is a muffin or a cupcake using SVM.

**Dataset**:



**Step 1:** Create a new Jupyter notebook and name it as “SVM Classifier”.

**Step 2:** Import the packages using the following code.

# Packages for analysis

import pandas as pd

import numpy as np

from sklearn import svm

# Packages for visuals

import matplotlib.pyplot as plt

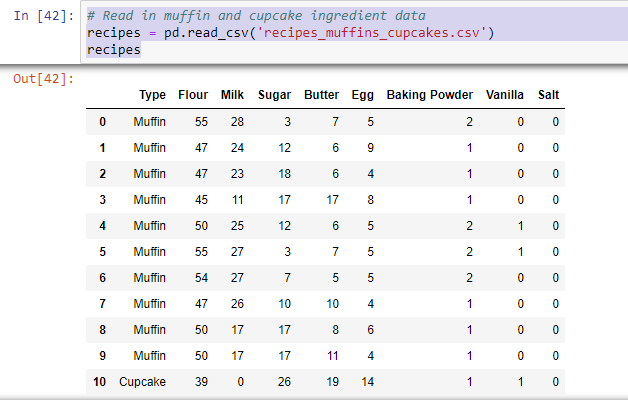
import seaborn as sns; sns.set(font\_scale=1.2)

**Step 3:** Import the data using the following code.

# Read in muffin and cupcake ingredient data

recipes = pd.read\_csv('recipes\_muffins\_cupcakes.csv')

recipes

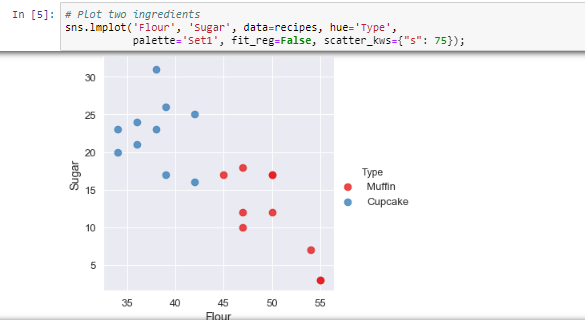
**Output:** 

**Step 4:** Plot the data for the given dataset using the following code.

# Plot two ingredients

sns.lmplot('Flour', 'Sugar', data=recipes, hue='Type',

palette='Set1', fit\_reg=False, scatter\_kws={"s": 75});

**Output:** 

**Step 5:** Categorizing the types as numerical using the following code.

# Specify inputs for the model

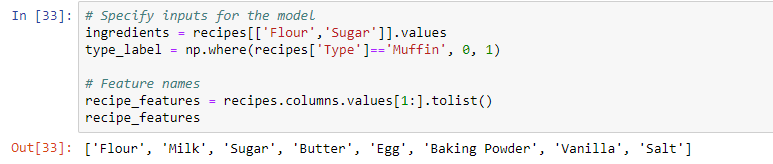
ingredients = recipes[['Flour','Sugar']].values

type\_label = np.where(recipes['Type']=='Muffin', 0, 1)

# Feature names

recipe\_features = recipes.columns.values[1:].tolist()

recipe\_features

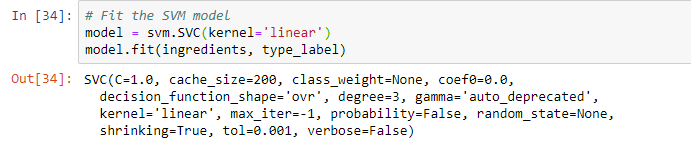
**Output:** 

**Step 6:** Instantiate SVC classifier and perform fit on the model using the following code.

# Fit the SVM model

model = svm.SVC(kernel='linear')

model.fit(ingredients, type\_label)

**Output:** 

**Step 7:** Get the separating hyperplane and plot the parallels to the separating hyperplane that pass through the support vectors using the following code.

# Get the separating hyperplane

w = model.coef\_[0]

a = -w[0] / w[1]

xx = np.linspace(30, 60)

print(xx)

yy = a \* xx - (model.intercept\_[0]) / w[1]

print(yy)

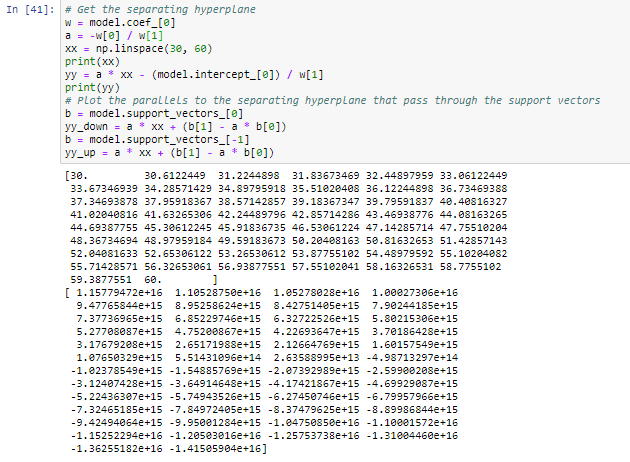
# Plot the parallels to the separating hyperplane that pass through the support vectors

b = model.support\_vectors\_[0]

yy\_down = a \* xx + (b[1] - a \* b[0])

b = model.support\_vectors\_[-1]

yy\_up = a \* xx + (b[1] - a \* b[0])

**Output:**

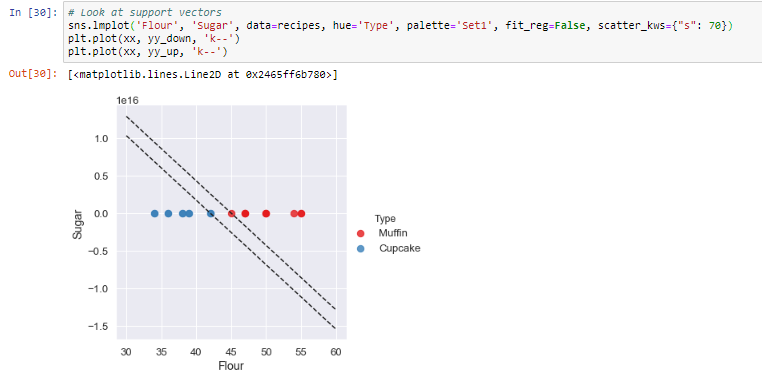
**Step 8:** Create the support vectors using the following code.

# Look at support vectors

sns.lmplot('Flour', 'Sugar', data=recipes, hue='Type', palette='Set1', fit\_reg=False, scatter\_kws={"s": 70})

plt.plot(xx, yy\_down, 'k--')

plt.plot(xx, yy\_up, 'k--')

**Output:** 

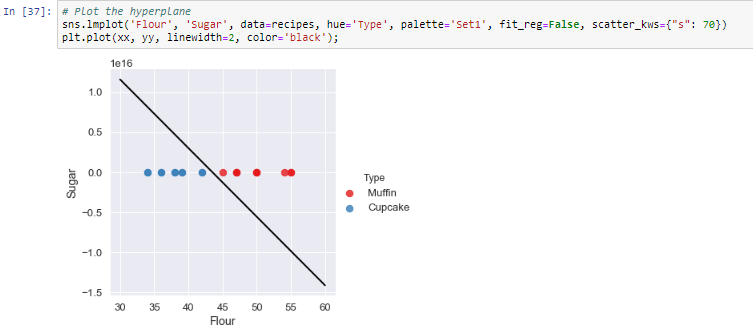
**Step 9:** Plot the hyperplane using the following code.

# Plot the hyperplane

sns.lmplot('Flour', 'Sugar', data=recipes, hue='Type', palette='Set1', fit\_reg=False, scatter\_kws={"s": 70})

plt.plot(xx, yy, linewidth=2, color='black');

**Output:**



**Step 10:** Create a midpoint hyperplane using the support vectors with the following code.

# Look at the margins and support vectors

sns.lmplot('Flour', 'Sugar', data=recipes, hue='Type', palette='Set1', fit\_reg=False, scatter\_kws={"s": 70})

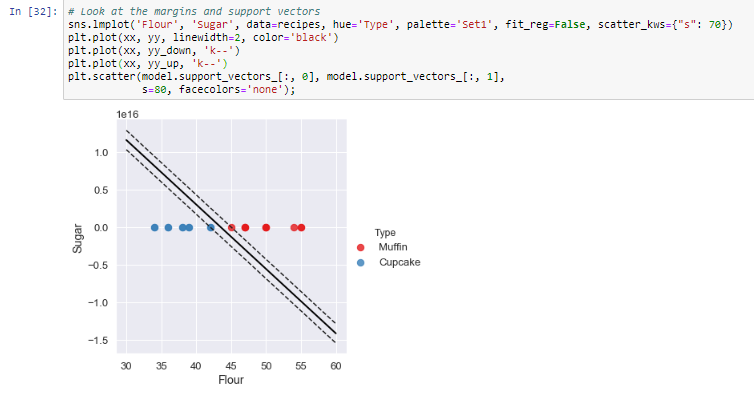
plt.plot(xx, yy, linewidth=2, color='black')

plt.plot(xx, yy\_down, 'k--')

plt.plot(xx, yy\_up, 'k--')

plt.scatter(model.support\_vectors\_[:, 0], model.support\_vectors\_[:, 1],

s=80, facecolors='none');

**Output:** 

**Step 11:** Create a function to predict whether a given recipe is a muffin or a cupcake using the following code.

# Create a function to guess whether a recipe is a muffin or a cupcake

def muffin\_or\_cupcake(flour, sugar):

if(model.predict([[flour, sugar]]))==0:

print('You\'re looking at a muffin recipe!')

else:

print('You\'re looking at a cupcake recipe!')

**Output:** 