09_exercise_support_vector_machines_robertson

April 14, 2025

Exercise 9: Support Vector Machines

CPSC 381/581: Machine Learning

Yale University

Instructor: Alex Wong

Student: Hailey Robertson

Prerequisites:

1. Enable Google Colaboratory as an app on your Google Drive account
2. Create a new Google Colab notebook, this will also create a "Colab Notebooks" directory under "MyDrive" i.e.

```
/content/drive/MyDrive/Colab Notebooks
```

3. Create the following directory structure in your Google Drive

```
/content/drive/MyDrive/Colab Notebooks/CPSC 381-581: Machine Learning/Exercises
```

4. Move the 09_exercise_support_vector_machines.ipynb into

```
/content/drive/MyDrive/Colab Notebooks/CPSC 381-581: Machine Learning/Exercises
```

so that its absolute path is

```
/content/drive/MyDrive/Colab Notebooks/CPSC 381-581: Machine Learning/Exercises/09_exercise_su
```

In this exercise, we will optimize a perceptron, logistic regression, and support vector machine on 4 datasets. We can compare classification accuracy across all the datasets to see which method is the best.

Submission:

1. Implement all TODOs in the code blocks below.
2. Report your validation scores for each method averaged over 10 trials.

```
***** Mean accuracy across 10 trials *****
Perceptron: 0.7528143274853802
Logistic Regression: 0.9443969298245612
Support Vector Machine: 0.9682346491228071
```

3. List any collaborators.

```
Collaborators: None.
```

Import packages

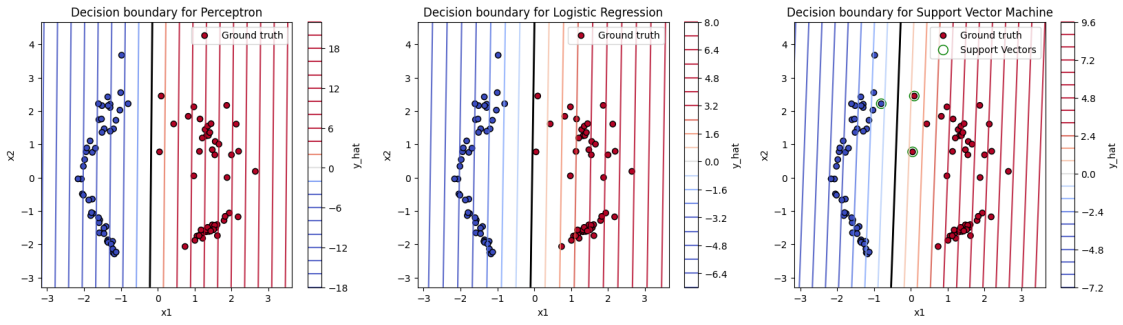
[1]:
```
import numpy as np
import sklearn.datasets as skdata
import sklearn.metrics as skmetrics
from sklearn.svm import SVC
from sklearn.linear_model import Perceptron
from sklearn.linear_model import LogisticRegression
from matplotlib import pyplot as plt
import warnings

warnings.filterwarnings(action='ignore')
np.random.seed = 1
```

Visualize the decision boundary and support vectors of different classifiers

[ ]:
```
# Generate synthetic data
X, y = skdata.make_classification(
    n_features=2,
    n_classes=2,
    n_redundant=0,
    n_clusters_per_class=2,
    n_samples=100,
    class_sep=1.5,
    random_state=1)

methods = [
    'perceptron',
    'logistic_regression',
    'support_vector_machine'
]

# DONE: Create figure with figsize=(20, 5)
fig = plt.figure(figsize=(20, 5))

# DONE: Enumerate through methods with index
for method in methods:

    # Instantiate model
    if method == 'perceptron':

        # DONE: Instantiate perceptron model with tolerance of 1e-1 and alpha␣
 ↪of 0
        model = Perceptron(tol=1e-1, alpha=0)

    elif method == 'logistic_regression':

        # DONE: Instantiate logistic regression model with tolerance of 1e-1
```

```
        model = LogisticRegression(tol=1e-1)

    elif method == 'support_vector_machine':

        # DONE: Instantiate SVC (Support Vector Machine Classifier) with␣
↪tolerance of 1e-1 and C=1e10 (simulates a hard-SVM) using a linear kernel
        model = SVC(tol=1e-1, C=1e10, kernel='linear')

    else:
        raise ValueError('Unsupported method: {}'.format(method))

    # DONE: Train the model
    model.fit(X, y)


    # DONE: Get x1_min and x1_max (0-th dimension), and x2_min and x2_max (1-st␣
↪dimension) from X
    # Subtract 1 to give a little buffer in plots
    x1_min, x1_max = X[:, 0].min() - 1, X[:, 0].max() +1
    x2_min, x2_max = X[:, 1].min() -1, X[:, 1].max() + 1

    # DONE: Create 2 linspaces: one from x1_min to x1_max and the other from␣
↪x1_min to x2_max with 500 units
    x1_linspace = np.linspace(x1_min, x1_max, 500)
    x2_linspace = np.linspace(x2_min, x2_max, 500)

    # DONE: Create meshgrid for x1 and x2 using linspaces
    x1, x2 = np.meshgrid(x1_linspace, x2_linspace)

    # DONE: Predict values for every point in meshgrid
    # maybe dstack?
    all_Xs = np.c_[x1.ravel(), x2.ravel()]
    y_hat = model.decision_function(all_Xs)


    # DONE: Reshape y_hat to x1 or x2's shape
    y_hat = y_hat.reshape(x1.shape)

    # DONE: Instantiate axis for subplot of a 1 x 3 figure
    ax = fig.add_subplot(1, 3, methods.index(method) + 1)

    # DONE: Plot Contour for predictions with levels=20, cmap='coolwarm',␣
↪alpha=0.8, vmin=-3, vmax=3
    contour = ax.contour(x1, x2, y_hat, levels=20, cmap='coolwarm', alpha=0.8,␣
↪vmin=-3, vmax=3)

    # DONE: Create colorbar for contour on axis and set its label to 'y_hat'
```

```
    cbar = plt.colorbar(contour, ax=ax, )
    cbar.set_label('y_hat')


    # DONE: Plot decision boundary using levels=[0], colors='black',␣
↪linewidths=2
    decision_boundary = ax.contour(x1, x2, y_hat, levels=[0], colors='black',␣
↪linewidths=2)

    # DONE: Create scatter plot for X and set its color to y with␣
↪edgecolor='black', cmap='coolwarm', label='Ground truth'
    ax.scatter(X[:, 0], X[:, 1], c=y, edgecolor='black', cmap='coolwarm',␣
↪label='Ground truth')

    # DONE: If support vector machine
    # Create scatter plot of support vectors with s=100, facecolors='none',␣
↪edgecolors='green', label='Support Vectors'
    if method == 'support_vector_machine':
        ax.scatter(model.support_vectors_[:, 0], model.support_vectors_[:, 1],␣
↪s=100, facecolors='none', edgecolors='green', label='Support Vectors')

    # DONE: Set title to ''Decision boundary for {}'
    # ALso clean up for better viz practices
    ax.set_title('Decision boundary for {}'.format(method.replace('_', ' ').
↪title()))


    # DONE: Set xlabel to 'x1'
    ax.set_xlabel('x1')

    # DONE: Set ylabel to 'x2'
    ax.set_ylabel('x2')

    # DONE: Set legend with loc='upper right'
    ax.legend(loc='upper right')

plt.show()
```



Load datasets

[12]:
```
# Load datasets
datasets = [
    skdata.load_iris(),
    skdata.load_breast_cancer(),
    skdata.load_digits(),
    skdata.load_wine()
]

dataset_names = [
    'Iris',
    'Breast cancer',
    'Digits',
    'Wine'
]
```

Compare Perceptron, Logistic Regression, and Support Vector Machines across all datasets

[ ]:
```
# Define lists to hold validation scores across trials
scores_val = {
    'perceptron' : [],
    'logistic_regression': [],
    'support_vector_machine' : []
}

# Perform 10 trials of experiments
n_trial = 10

for n in range(n_trial):

    print('***** TRIAL {} *****\n'.format(n))

    # Zip up all dataset options
    dataset_options = zip(
        datasets,
        dataset_names)

    for dataset, dataset_name in dataset_options:

        '''
        Create the training and validation splits
        '''
        X = dataset.data
        y = dataset.target
```

```
        print('Preprocessing the {} dataset ({} samples, {} feature␣
↪dimensions)'.format(dataset_name, X.shape[0], X.shape[1]))

        # Shuffle the dataset based on sample indices
        shuffled_indices = np.random.permutation(X.shape[0])

        # Choose the first 80% as training set and the next 20% as validation
        train_split_idx = int(0.80 * X.shape[0])

        train_indices = shuffled_indices[0:train_split_idx]
        val_indices = shuffled_indices[train_split_idx:]

        # Select the examples from X and y to construct our training and␣
↪validation sets
        X_train, y_train = X[train_indices, :], y[train_indices]
        X_val, y_val = X[val_indices, :], y[val_indices]

        for method in ['perceptron', 'logistic_regression',␣
↪'support_vector_machine']:

            print('***** Experiments on the {} dataset using {} model *****'.
↪format(
                dataset_name,
                method))

            # Instantiate model
            if method == 'perceptron':

                # DONE: Instantiate perceptron model with tolerance of 1e-1 and␣
↪alpha of 0
                model = Perceptron(tol=1e-1, alpha=0)

            elif method == 'logistic_regression':

                # DONE: Instantiate logistic regression model with tolerance of␣
↪1e-1
                model = LogisticRegression(tol=1e-1)

            elif method == 'support_vector_machine':

                # DONE: Instantiate SVC (Support Vector Machine Classifier)␣
↪with tolerance of 1e-1 and C=1 (soft SVM) using a linear kernel
                model = SVC(tol=1e-1, C=1, kernel='linear')

            else:
```

```
                raise ValueError('Unsupported method: {}'.format(method))


            # DONE: Train the model
            model.fit(X_train, y_train)


            # DONE: Score model using mean accuracy on training set
            predictions_train = model.predict(X_train)
            score_train = skmetrics.accuracy_score(y_train, predictions_train)
            print('Training set mean accuracy: {:.4f}'.format(score_train))

            # DONE: Score model using mean accuracy validation set
            predictions_val = model.predict(X_val)
            score_val = skmetrics.accuracy_score(y_val, predictions_val)
            print('Validation set mean accuracy: {:.4f}'.format(score_val))

            # DONE: Append score to validation scores for the given method
            scores_val[method].append(score_val)


        print('')

# DONE: Compute mean over trials for each method
mean_scores_val_perceptron = np.mean(scores_val['perceptron'])
mean_scores_val_logistic = np.mean(scores_val['logistic_regression'])
mean_scores_val_svm = np.mean(scores_val['support_vector_machine'])

print('***** Mean accuracy across {} trials *****'.format(n_trial))

print('Perceptron: {}'.format(mean_scores_val_perceptron))
print('Logistic Regression: {}'.format(mean_scores_val_logistic))
print('Support Vector Machine: {}'.format(mean_scores_val_svm))
```

```
***** TRIAL 0 *****

Preprocessing the Iris dataset (150 samples, 4 feature dimensions)
***** Experiments on the Iris dataset using perceptron model *****
Training set mean accuracy: 0.8667
Validation set mean accuracy: 0.9333
***** Experiments on the Iris dataset using logistic_regression model *****
Training set mean accuracy: 0.9750
Validation set mean accuracy: 0.9333
***** Experiments on the Iris dataset using support_vector_machine model *****
Training set mean accuracy: 0.9917
Validation set mean accuracy: 0.9667

Preprocessing the Breast cancer dataset (569 samples, 30 feature dimensions)
```

```
***** Experiments on the Breast cancer dataset using perceptron model *****
Training set mean accuracy: 0.8593
Validation set mean accuracy: 0.8772
***** Experiments on the Breast cancer dataset using logistic_regression model
*****
Training set mean accuracy: 0.9209
Validation set mean accuracy: 0.9737
***** Experiments on the Breast cancer dataset using support_vector_machine
model *****
Training set mean accuracy: 0.9604
Validation set mean accuracy: 0.9737

Preprocessing the Digits dataset (1797 samples, 64 feature dimensions)
***** Experiments on the Digits dataset using perceptron model *****
Training set mean accuracy: 0.9722
Validation set mean accuracy: 0.9333
***** Experiments on the Digits dataset using logistic_regression model *****
Training set mean accuracy: 0.9687
Validation set mean accuracy: 0.9278
***** Experiments on the Digits dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9833

Preprocessing the Wine dataset (178 samples, 13 feature dimensions)
***** Experiments on the Wine dataset using perceptron model *****
Training set mean accuracy: 0.4789
Validation set mean accuracy: 0.5556
***** Experiments on the Wine dataset using logistic_regression model *****
Training set mean accuracy: 0.9225
Validation set mean accuracy: 0.9722
***** Experiments on the Wine dataset using support_vector_machine model *****
Training set mean accuracy: 0.9930
Validation set mean accuracy: 1.0000

***** TRIAL 1 *****

Preprocessing the Iris dataset (150 samples, 4 feature dimensions)
***** Experiments on the Iris dataset using perceptron model *****
Training set mean accuracy: 0.6833
Validation set mean accuracy: 0.6000
***** Experiments on the Iris dataset using logistic_regression model *****
Training set mean accuracy: 0.9667
Validation set mean accuracy: 0.9667
***** Experiments on the Iris dataset using support_vector_machine model *****
Training set mean accuracy: 0.9750
Validation set mean accuracy: 1.0000

Preprocessing the Breast cancer dataset (569 samples, 30 feature dimensions)
```

```
***** Experiments on the Breast cancer dataset using perceptron model *****
Training set mean accuracy: 0.8879
Validation set mean accuracy: 0.8421
***** Experiments on the Breast cancer dataset using logistic_regression model
*****
Training set mean accuracy: 0.9407
Validation set mean accuracy: 0.9035
***** Experiments on the Breast cancer dataset using support_vector_machine
model *****
Training set mean accuracy: 0.9692
Validation set mean accuracy: 0.9649

Preprocessing the Digits dataset (1797 samples, 64 feature dimensions)
***** Experiments on the Digits dataset using perceptron model *****
Training set mean accuracy: 0.9743
Validation set mean accuracy: 0.9361
***** Experiments on the Digits dataset using logistic_regression model *****
Training set mean accuracy: 0.9645
Validation set mean accuracy: 0.9306
***** Experiments on the Digits dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9667

Preprocessing the Wine dataset (178 samples, 13 feature dimensions)
***** Experiments on the Wine dataset using perceptron model *****
Training set mean accuracy: 0.4718
Validation set mean accuracy: 0.4444
***** Experiments on the Wine dataset using logistic_regression model *****
Training set mean accuracy: 0.9718
Validation set mean accuracy: 0.9444
***** Experiments on the Wine dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9444

***** TRIAL 2 *****

Preprocessing the Iris dataset (150 samples, 4 feature dimensions)
***** Experiments on the Iris dataset using perceptron model *****
Training set mean accuracy: 0.6583
Validation set mean accuracy: 0.6667
***** Experiments on the Iris dataset using logistic_regression model *****
Training set mean accuracy: 0.9750
Validation set mean accuracy: 0.9667
***** Experiments on the Iris dataset using support_vector_machine model *****
Training set mean accuracy: 0.9750
Validation set mean accuracy: 1.0000

Preprocessing the Breast cancer dataset (569 samples, 30 feature dimensions)
```

```
***** Experiments on the Breast cancer dataset using perceptron model *****
Training set mean accuracy: 0.6791
Validation set mean accuracy: 0.5877
***** Experiments on the Breast cancer dataset using logistic_regression model
*****
Training set mean accuracy: 0.9319
Validation set mean accuracy: 0.9298
***** Experiments on the Breast cancer dataset using support_vector_machine
model *****
Training set mean accuracy: 0.9582
Validation set mean accuracy: 0.9737

Preprocessing the Digits dataset (1797 samples, 64 feature dimensions)
***** Experiments on the Digits dataset using perceptron model *****
Training set mean accuracy: 0.9722
Validation set mean accuracy: 0.9556
***** Experiments on the Digits dataset using logistic_regression model *****
Training set mean accuracy: 0.9694
Validation set mean accuracy: 0.9472
***** Experiments on the Digits dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9861

Preprocessing the Wine dataset (178 samples, 13 feature dimensions)
***** Experiments on the Wine dataset using perceptron model *****
Training set mean accuracy: 0.5352
Validation set mean accuracy: 0.4167
***** Experiments on the Wine dataset using logistic_regression model *****
Training set mean accuracy: 0.9648
Validation set mean accuracy: 0.9722
***** Experiments on the Wine dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9444

***** TRIAL 3 *****

Preprocessing the Iris dataset (150 samples, 4 feature dimensions)
***** Experiments on the Iris dataset using perceptron model *****
Training set mean accuracy: 0.7000
Validation set mean accuracy: 0.6333
***** Experiments on the Iris dataset using logistic_regression model *****
Training set mean accuracy: 0.9583
Validation set mean accuracy: 0.9667
***** Experiments on the Iris dataset using support_vector_machine model *****
Training set mean accuracy: 0.9833
Validation set mean accuracy: 0.9667

Preprocessing the Breast cancer dataset (569 samples, 30 feature dimensions)
```

```
***** Experiments on the Breast cancer dataset using perceptron model *****
Training set mean accuracy: 0.5560
Validation set mean accuracy: 0.4649
***** Experiments on the Breast cancer dataset using logistic_regression model
*****
Training set mean accuracy: 0.9297
Validation set mean accuracy: 0.9211
***** Experiments on the Breast cancer dataset using support_vector_machine
model *****
Training set mean accuracy: 0.9670
Validation set mean accuracy: 0.9474

Preprocessing the Digits dataset (1797 samples, 64 feature dimensions)
***** Experiments on the Digits dataset using perceptron model *****
Training set mean accuracy: 0.9603
Validation set mean accuracy: 0.9500
***** Experiments on the Digits dataset using logistic_regression model *****
Training set mean accuracy: 0.9631
Validation set mean accuracy: 0.9556
***** Experiments on the Digits dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9833

Preprocessing the Wine dataset (178 samples, 13 feature dimensions)
***** Experiments on the Wine dataset using perceptron model *****
Training set mean accuracy: 0.4930
Validation set mean accuracy: 0.6389
***** Experiments on the Wine dataset using logistic_regression model *****
Training set mean accuracy: 0.9718
Validation set mean accuracy: 0.9722
***** Experiments on the Wine dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9444

***** TRIAL 4 *****

Preprocessing the Iris dataset (150 samples, 4 feature dimensions)
***** Experiments on the Iris dataset using perceptron model *****
Training set mean accuracy: 0.6917
Validation set mean accuracy: 0.5667
***** Experiments on the Iris dataset using logistic_regression model *****
Training set mean accuracy: 0.9750
Validation set mean accuracy: 0.9000
***** Experiments on the Iris dataset using support_vector_machine model *****
Training set mean accuracy: 0.9917
Validation set mean accuracy: 0.9333

Preprocessing the Breast cancer dataset (569 samples, 30 feature dimensions)
```

```
***** Experiments on the Breast cancer dataset using perceptron model *****
Training set mean accuracy: 0.8967
Validation set mean accuracy: 0.8860
***** Experiments on the Breast cancer dataset using logistic_regression model
*****
Training set mean accuracy: 0.9253
Validation set mean accuracy: 0.9386
***** Experiments on the Breast cancer dataset using support_vector_machine
model *****
Training set mean accuracy: 0.9604
Validation set mean accuracy: 0.9649

Preprocessing the Digits dataset (1797 samples, 64 feature dimensions)
***** Experiments on the Digits dataset using perceptron model *****
Training set mean accuracy: 0.9749
Validation set mean accuracy: 0.9500
***** Experiments on the Digits dataset using logistic_regression model *****
Training set mean accuracy: 0.9680
Validation set mean accuracy: 0.9472
***** Experiments on the Digits dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9750

Preprocessing the Wine dataset (178 samples, 13 feature dimensions)
***** Experiments on the Wine dataset using perceptron model *****
Training set mean accuracy: 0.5282
Validation set mean accuracy: 0.5833
***** Experiments on the Wine dataset using logistic_regression model *****
Training set mean accuracy: 0.9437
Validation set mean accuracy: 0.9444
***** Experiments on the Wine dataset using support_vector_machine model *****
Training set mean accuracy: 0.9930
Validation set mean accuracy: 0.9722

***** TRIAL 5 *****

Preprocessing the Iris dataset (150 samples, 4 feature dimensions)
***** Experiments on the Iris dataset using perceptron model *****
Training set mean accuracy: 0.8667
Validation set mean accuracy: 0.8667
***** Experiments on the Iris dataset using logistic_regression model *****
Training set mean accuracy: 0.9833
Validation set mean accuracy: 0.8667
***** Experiments on the Iris dataset using support_vector_machine model *****
Training set mean accuracy: 0.9833
Validation set mean accuracy: 0.9667

Preprocessing the Breast cancer dataset (569 samples, 30 feature dimensions)
```

```
***** Experiments on the Breast cancer dataset using perceptron model *****
Training set mean accuracy: 0.9165
Validation set mean accuracy: 0.8860
***** Experiments on the Breast cancer dataset using logistic_regression model
*****
Training set mean accuracy: 0.9385
Validation set mean accuracy: 0.8947
***** Experiments on the Breast cancer dataset using support_vector_machine
model *****
Training set mean accuracy: 0.9714
Validation set mean accuracy: 0.9386

Preprocessing the Digits dataset (1797 samples, 64 feature dimensions)
***** Experiments on the Digits dataset using perceptron model *****
Training set mean accuracy: 0.9652
Validation set mean accuracy: 0.9611
***** Experiments on the Digits dataset using logistic_regression model *****
Training set mean accuracy: 0.9666
Validation set mean accuracy: 0.9556
***** Experiments on the Digits dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9917

Preprocessing the Wine dataset (178 samples, 13 feature dimensions)
***** Experiments on the Wine dataset using perceptron model *****
Training set mean accuracy: 0.5282
Validation set mean accuracy: 0.4167
***** Experiments on the Wine dataset using logistic_regression model *****
Training set mean accuracy: 0.9718
Validation set mean accuracy: 0.9722
***** Experiments on the Wine dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9444

***** TRIAL 6 *****

Preprocessing the Iris dataset (150 samples, 4 feature dimensions)
***** Experiments on the Iris dataset using perceptron model *****
Training set mean accuracy: 0.9333
Validation set mean accuracy: 0.9333
***** Experiments on the Iris dataset using logistic_regression model *****
Training set mean accuracy: 0.9750
Validation set mean accuracy: 1.0000
***** Experiments on the Iris dataset using support_vector_machine model *****
Training set mean accuracy: 0.9750
Validation set mean accuracy: 0.9667

Preprocessing the Breast cancer dataset (569 samples, 30 feature dimensions)
```

```
***** Experiments on the Breast cancer dataset using perceptron model *****
Training set mean accuracy: 0.8549
Validation set mean accuracy: 0.8246
***** Experiments on the Breast cancer dataset using logistic_regression model
*****
Training set mean accuracy: 0.9253
Validation set mean accuracy: 0.9474
***** Experiments on the Breast cancer dataset using support_vector_machine
model *****
Training set mean accuracy: 0.9516
Validation set mean accuracy: 0.9737

Preprocessing the Digits dataset (1797 samples, 64 feature dimensions)
***** Experiments on the Digits dataset using perceptron model *****
Training set mean accuracy: 0.9631
Validation set mean accuracy: 0.9528
***** Experiments on the Digits dataset using logistic_regression model *****
Training set mean accuracy: 0.9617
Validation set mean accuracy: 0.9472
***** Experiments on the Digits dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9722

Preprocessing the Wine dataset (178 samples, 13 feature dimensions)
***** Experiments on the Wine dataset using perceptron model *****
Training set mean accuracy: 0.6056
Validation set mean accuracy: 0.7222
***** Experiments on the Wine dataset using logistic_regression model *****
Training set mean accuracy: 0.9718
Validation set mean accuracy: 0.9722
***** Experiments on the Wine dataset using support_vector_machine model *****
Training set mean accuracy: 0.9930
Validation set mean accuracy: 1.0000

***** TRIAL 7 *****

Preprocessing the Iris dataset (150 samples, 4 feature dimensions)
***** Experiments on the Iris dataset using perceptron model *****
Training set mean accuracy: 0.9250
Validation set mean accuracy: 0.8667
***** Experiments on the Iris dataset using logistic_regression model *****
Training set mean accuracy: 0.9667
Validation set mean accuracy: 0.9333
***** Experiments on the Iris dataset using support_vector_machine model *****
Training set mean accuracy: 0.9917
Validation set mean accuracy: 0.9333

Preprocessing the Breast cancer dataset (569 samples, 30 feature dimensions)
```

```
***** Experiments on the Breast cancer dataset using perceptron model *****
Training set mean accuracy: 0.8923
Validation set mean accuracy: 0.8772
***** Experiments on the Breast cancer dataset using logistic_regression model
*****
Training set mean accuracy: 0.9385
Validation set mean accuracy: 0.9649
***** Experiments on the Breast cancer dataset using support_vector_machine
model *****
Training set mean accuracy: 0.9648
Validation set mean accuracy: 0.9561

Preprocessing the Digits dataset (1797 samples, 64 feature dimensions)
***** Experiments on the Digits dataset using perceptron model *****
Training set mean accuracy: 0.9694
Validation set mean accuracy: 0.9333
***** Experiments on the Digits dataset using logistic_regression model *****
Training set mean accuracy: 0.9673
Validation set mean accuracy: 0.9472
***** Experiments on the Digits dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9806

Preprocessing the Wine dataset (178 samples, 13 feature dimensions)
***** Experiments on the Wine dataset using perceptron model *****
Training set mean accuracy: 0.5563
Validation set mean accuracy: 0.4444
***** Experiments on the Wine dataset using logistic_regression model *****
Training set mean accuracy: 0.9859
Validation set mean accuracy: 0.8611
***** Experiments on the Wine dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9444

***** TRIAL 8 *****

Preprocessing the Iris dataset (150 samples, 4 feature dimensions)
***** Experiments on the Iris dataset using perceptron model *****
Training set mean accuracy: 0.3333
Validation set mean accuracy: 0.4333
***** Experiments on the Iris dataset using logistic_regression model *****
Training set mean accuracy: 0.9333
Validation set mean accuracy: 0.9667
***** Experiments on the Iris dataset using support_vector_machine model *****
Training set mean accuracy: 0.9917
Validation set mean accuracy: 1.0000

Preprocessing the Breast cancer dataset (569 samples, 30 feature dimensions)
```

```
***** Experiments on the Breast cancer dataset using perceptron model *****
Training set mean accuracy: 0.8110
Validation set mean accuracy: 0.7807
***** Experiments on the Breast cancer dataset using logistic_regression model
*****
Training set mean accuracy: 0.9341
Validation set mean accuracy: 0.9211
***** Experiments on the Breast cancer dataset using support_vector_machine
model *****
Training set mean accuracy: 0.9670
Validation set mean accuracy: 0.9386

Preprocessing the Digits dataset (1797 samples, 64 feature dimensions)
***** Experiments on the Digits dataset using perceptron model *****
Training set mean accuracy: 0.9701
Validation set mean accuracy: 0.9528
***** Experiments on the Digits dataset using logistic_regression model *****
Training set mean accuracy: 0.9645
Validation set mean accuracy: 0.9694
***** Experiments on the Digits dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9861

Preprocessing the Wine dataset (178 samples, 13 feature dimensions)
***** Experiments on the Wine dataset using perceptron model *****
Training set mean accuracy: 0.6549
Validation set mean accuracy: 0.6389
***** Experiments on the Wine dataset using logistic_regression model *****
Training set mean accuracy: 0.9859
Validation set mean accuracy: 0.9167
***** Experiments on the Wine dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9444

***** TRIAL 9 *****

Preprocessing the Iris dataset (150 samples, 4 feature dimensions)
***** Experiments on the Iris dataset using perceptron model *****
Training set mean accuracy: 0.8417
Validation set mean accuracy: 0.7333
***** Experiments on the Iris dataset using logistic_regression model *****
Training set mean accuracy: 0.9833
Validation set mean accuracy: 0.9333
***** Experiments on the Iris dataset using support_vector_machine model *****
Training set mean accuracy: 0.9833
Validation set mean accuracy: 0.9667

Preprocessing the Breast cancer dataset (569 samples, 30 feature dimensions)
```

```
***** Experiments on the Breast cancer dataset using perceptron model *****
Training set mean accuracy: 0.7473
Validation set mean accuracy: 0.7807
***** Experiments on the Breast cancer dataset using logistic_regression model
*****
Training set mean accuracy: 0.9253
Validation set mean accuracy: 0.9561
***** Experiments on the Breast cancer dataset using support_vector_machine
model *****
Training set mean accuracy: 0.9626
Validation set mean accuracy: 0.9561

Preprocessing the Digits dataset (1797 samples, 64 feature dimensions)
***** Experiments on the Digits dataset using perceptron model *****
Training set mean accuracy: 0.9791
Validation set mean accuracy: 0.9639
***** Experiments on the Digits dataset using logistic_regression model *****
Training set mean accuracy: 0.9659
Validation set mean accuracy: 0.9639
***** Experiments on the Digits dataset using support_vector_machine model *****
Training set mean accuracy: 1.0000
Validation set mean accuracy: 0.9778

Preprocessing the Wine dataset (178 samples, 13 feature dimensions)
***** Experiments on the Wine dataset using perceptron model *****
Training set mean accuracy: 0.6338
Validation set mean accuracy: 0.7222
***** Experiments on the Wine dataset using logistic_regression model *****
Training set mean accuracy: 0.9648
Validation set mean accuracy: 0.9722
***** Experiments on the Wine dataset using support_vector_machine model *****
Training set mean accuracy: 0.9930
Validation set mean accuracy: 1.0000

***** Mean accuracy across 10 trials *****
Perceptron: 0.7528143274853802
Logistic Regression: 0.9443969298245612
Support Vector Machine: 0.9682346491228071
```

[ ]: