Machine Learning Lab 5

SVM

Support vector machines (SVMs) are a type of machine learning algorithm that can be used for both classification and regression tasks. SVMs work by finding a hyperplane that separates the data into two classes with the maximum margin. The margin is the distance between the hyperplane and the closest data points on either side. SVMs are trained by finding the hyperplane that maximizes the margin. This is done by solving a quadratic programming problem. Once the hyperplane is found, it can be used to classify new data points by predicting on which side of the hyperplane they fall.

Support vector regression (SVR) is a type of support vector machine that can be used for regression tasks. SVR works by finding a hyperplane that best fits the data. The hyperplane is found by minimizing the error between the predicted values and the actual values. SVR is trained by solving a quadratic programming problem. Once the hyperplane is found, it can be used to predict the values of new data points by finding the point on the hyperplane that is closest to the new data point's features.

SVMs and SVRs are powerful machine learning algorithms that can be used to solve a variety of classification and regression problems. However, they can be computationally expensive to train, especially for large datasets.

Scikit-learn provides various tools for model fitting, data preprocessing, model selection, model evaluation, and many other utilities. We will be using this library to create models of SVMs.

About the datasets

Classification dataset 1

The Cy Young Voting dataset consists of data related to the voting results for Major League Baseball's (MLB) Cy Young Award, which is presented annually to the best pitchers in the American League and National League. The task here is to predict the Cy Young Award winners based on their season statistics.

Sample dataset:

Pitcher	Wins	Win_pct	ERA	Strikeouts	Innings_pitched	target
38	19	0.67857	3.87	215	216.3	0
45	19	0.7037	3.24	185	255.3	0
22	22	0.6875	3.37	261	256.7	1
24	7	0.58333	2.27	65	87.3	0
18	20	0.66667	2.36	228	267	1

Features:

- Pitcher: An identifier for the pitcher.
- Wins: The number of games the pitcher won.
- Win_pct: The winning percentage of the pitcher.
- ERA: Earned Run Average, a measure of a pitcher's effectiveness.
- Strikeouts: The total number of strikeouts recorded by the pitcher.
- Innings_pitched: The total number of innings the pitcher has pitched.
- target: This could be a binary variable indicating whether the pitcher won the Cy Young Award (1) or not (0) in the context of this dataset.

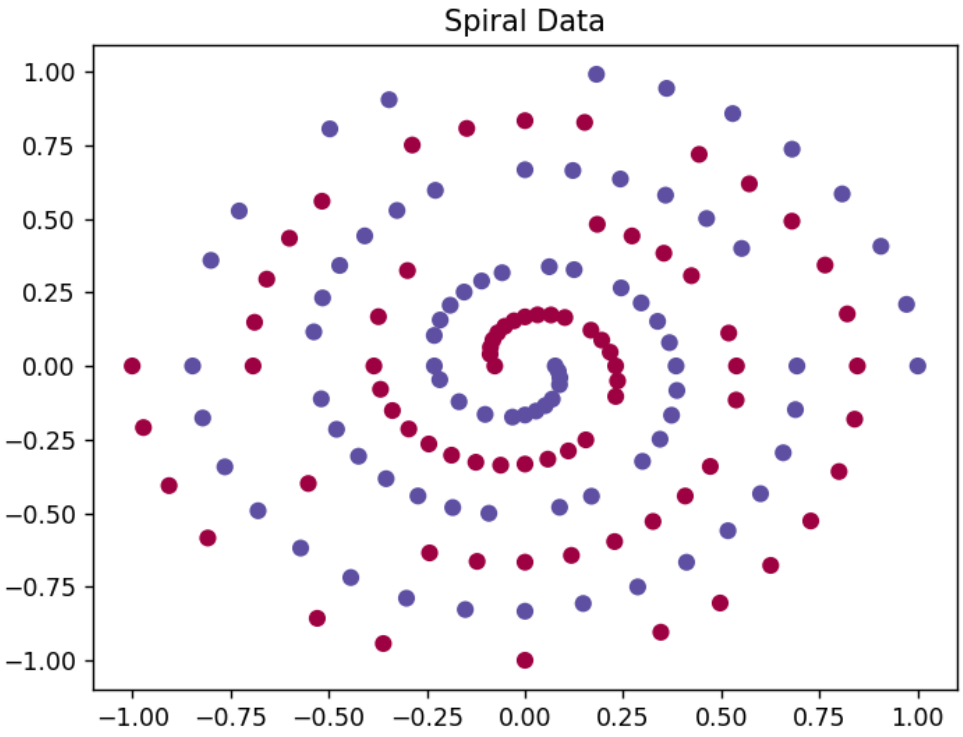
Regression dataset

This dataset consists of two columns. The percentage of alcohol present in the wine is determined using the fluidity of the wine.

- Fluidity: This represents a measure of how easily a substance flows
- Alcohol: This variable gives a measure of the percentage of alcohol in the fluid

Classification dataset 2 (spiral data)

SVMs can efficiently perform classification on non-linear data such as the spiral shown below, when you make use of kernel methods. This dataset has two classes (-1 and 1) in the shape of two interwoven spirals in the feature space.



Your task

Your task is to fill the following functions:

Function name	Input	Output
`dataset_read`	`dataset_path: str` - Path to the dataset	`X: pd.DataFrame` – Features `y: pd.Series` – Target labels
`preprocess`	`X: pd.DataFrame` – Feature matrix `y: pd.Series` – Target labels	
`train_regression_model`	`X_train: np.ndarray` – Training features `y_train: np.ndarray` – Training target	`None` (`self.model` should be trained on the regression dataset)
`train_classification_model`	`X_train: np.ndarray` – Training features `y_train: np.ndarray` – Training target	`None` (`self.model` should be trained on the classification dataset)
`train_spiral_model`	`X_train: np.ndarray` – Training features `y_train: np.ndarray` – Training target	`None` (`self.model` should be trained on the classification dataset)

General Instructions:

- The `dataset_read` and `preprocess` functions are to be filled separately in the all the three of classification and regression tasks.
- In `preprocess`, you are free to use any pre-processing you wish to use. Also standardise the feature set here, as SVMs perform better with normalized data
- Stick to using sklearn's SVM module to define the models.
- In the dataset provided, the last column has the target variable. The previous columns are explanatory variables.
- You may write additional helper functions.
- Do not make changes to the test file provided to you, and do not hardcode values.

The accuracy of the regression task is measured using the Mean Absolute Percentage Error (MAPE).
To pass all the test cases, the classification model must have test accuracy >=83% and the regression model must have test accuracy >=91%.

Testing your code

1. Rename the SVM.py file as `Lab5_<SRN>.py`
2. Run the command `python test_SVM.py --ID Lab5_<SRN>`, for example, `python test_SVM.py --ID Lab5_PES1UG22CS001`

Submission guidelines

- Submit only the Python solution file: named as `Lab5_<SRN>.py`
- Remove all print statements.
- Failing some hidden cases will lead to partial marks.
- Test cases provided to you are for reference only, hidden test cases will be similar