INNOPOLIS UNIVERSITY

Introduction to Machine Learning
Assignment 1

Bachelors

Spring

2023

1 General Instructions

You are required to submit your solutions via Moodle as a single zip file. The zip archive should contain a single Ipynb file and all data files as csv files. Please, put your name and email at innopolis.university as the first line in the notebook. Source code should be clean, easy to read, and well documented.

Bonus points may be awarded for elegant solutions. However, these bonus points will only be able to cancel the effect of penalties.

Plagiarised solutions will be heavily penalized for all parties involved. Also, do not just copy and paste solutions from the Internet. You are allowed to collaborate on general ideas with other students as well as consult books and Internet resources. However, be sure to credit the sources you use and type all the code,documentation by yourself.

2 Disclaimer

The objective of this assignment is for you to demonstrate your understanding and mastery of the materials that are being covered in the course of Introduction to Machine Learning. However, we are making an assumption that an average student of this course did not have any knowledge of machine learning prior to entering this class. Therefore, for this assignment, we are keeping things simple. You are given specific instructions on what you should do. The purpose is that for now you should just demonstrate that you can follow specific instructions to solve a machine learning task. Consider this as an opportunity to learn, demonstrate your understanding of the course material and easily earn all points. Your next assignment will be more challenging and open-ended.

3 Objectives of the Assignment

- The first objective is to allow you how to demonstrate to us that you can use grid search to find the best hyperparameters for a machine learning model.
- The second objective is to make you focus on data cleaning: Correction, completion, creation, and conversion.
- Finally, The third objective is to see that you can compare various machine learning models and know how to select the appropriate metric

for a given task.

Task one covers objective one, while task two will cover objectives two and three.

4 Task 1

4.1 Linear regression - 15 Points

In this task you will solve a synthetically generated regression problem. We have stored these data in the file "task_1.csv". This file contains "X_train", "y_train", "X_test", "y_test" columns. "X_train" Contains 1 feature and has 30 data points.

This dataset doesn't require any cleaning, imputing, encoding or scaling.

1. Plot X_train against y_train to visualize the data.
2. Using the linear regression model from sklearn, fit the model to the dataset.
3. predict on X_test and evaluate the performance while printing MSE, RMSE, MAE and R2 score.
4. Plot the test data and the predictions of the linear model on X_test.

Did the linear regression model give a good fit? elaborate your answer.

4.2 Polynomial regression - 15 Points

Let's try to fit a polynomial regression model to the same dataset and compare the performance. In polynomial regression you are going to deal with 1 hyper parameter, this hyper parameter is the degree. Because we don't know what is the correct degree, we have to tune this value using Grid Search algorithm. Grid Search link

1. By using the sklearn pipeline, construct a polynomial regression pipeline consisting of polynomial features class and linear regression class.
2. Use GridSearch to find the best polynomial regression model and print the best parameters. "degrees = range(2, 10)", cross-validation = 8, scoring = negative mean squared error, using the best params, predict on "X_test" and evaluate using MSE, RMSE, MAE and R2 score

3. It's a good programming exercise to manually implement grid search yourself. Manually loop over the degrees and construct a pipeline for each degree and evaluate the pipelines using the same metric "negative MSE" and cross-validation = 8 and check if you got the same result as Grid Search.
4. Plot the test data and the predictions of the best degree polynomial model on X_test

5 Task 2

In this task, you will be solving a binary classification task. You will be classifying a Pokemon whether it is a legendary Pokemon or not.
Information about the dataset:

- There are 36 columns that represent the features of pokemon.
- There are 801 rows in the dataset, each row is encoding a pokemon.
- The label column is "is_legendary" which tells you if the pokemon is legendary or not.
- There are 3 feature columns that are missing some values.
- There are 2 feature columns that should be removed.
- There is 1 feature column that should be categorically encoded.

5.1 Data preprocessing - 30 Points

In this section you have to load the data, analyze the features, decide which features should be removed, what should be encoded, and what should be imputed.

1. Load the pokemon dataset using pandas dataframe.
2. Remove the redundant 2 features and say why they should be removed.
3. Split the dataset into train/test with a ratio of 0.8/0.2. Is the dataset balanced?
4. Explore the dataset using pd.head(), pd.info() and check for missing values and Impute them using the SimpleImputer. You can use 'mean' strategy or 'most frequent'
5. Double check that there are no missing values.

6. Identify and encode the categorical feature using OneHot encoder.
7. Scale the data using MinMax normalization or StandardScaler.
8. plot the correlation matrix. Answer questions: Are there highly co-related features in the dataset? Is it a problem? Preprocess data if necessary.

5.2 Model fitting and comparison - 40 Points

You have to train and evaluate several classification models including:

- Sklearn Logistic Regression classifier
- Sklearn KNN classifier
- Sklearn Gaussian Naive-Bayes classifier

There are hyper-parameters for logistic regression and KNN which we need to tune using **Grid-Search**.
The hyperparameters of the logistic regression model are the regularization strength 'C', the regularization technique "penalty" l1: Lasso and l2: Ridge and the solver.
The hyperparameter of the KNN is the K value which is the number of neighbors, the metric which is the distance computation function and the weight, whether to give all the neighbouring points the same weight or give a weight based on the distance.

1. Using GridSearchCV, find best hyper-parameters for the Logistic Regression model. Try different variations with penalty: ['l1','l2'], regularization strength: np.logspace(-3,3,7), solver : ['newton-cg', 'lbfgs', 'liblinear']
2. Evaluate the logistic regression model with the best parameters on the test data using accuracy, precision, recall, and F1 score.
3. Print the regression coefficients and find out the names of the top 5 most influencing features and the top 5 ignored features.
4. Using GridSearchCV, find best hyper-parameter for the KNN model. Try different variations with k :list(range(1, 15)), 'weights':['uniform', 'distance'], 'metric':['euclidean', 'manhattan', 'chebyshev', 'cosine']
5. Evaluate the KNN model with the best parameters on the test data using accuracy, precision, recall and F1 score.

6. Fit Gaussian Naive-Bayes to the data and evaluate on the test dataset while printing the metrics.
7. Which metric is most appropriate for this task and why?
8. Compare the 3 classifiers in terms of accuracy, precision, recall and F1-score. What is the best model for this task? and based on what did you pick it?

You'll find a markdown cell for each theoritical question and you should answer that question there.

6 Bonus task - 10 Points

In the previous task, we implemented a binary classifier to classify if a pokemon is legendary or not. In this task, we will take this concept one step further and our problem will be multi-class classification problem. There are two techniques where you can solve this problem, the first one being One-vs-All and the second one being Multinomial.
The multinomial classifier does not classifier each class seperately, instead it uses the softmax function to predict if a single data point falls in one of the 'N' classes.
The one-vs-all strategy, consists in fitting one classifier per class. For each classifier, the class is fitted against all the other classes. The class with the largest probability is chosen.

The dataset contains 3 features and 1 outcome variable with 3 classes. The dataset is clean and preprocessed for you and there are no missing values.

1. Load the data from "bonus_train.csv" and "bonus_test.csv" using pandas dataframe and then split the training data to "X_train" and "y_train", split the test data to "X_test" and "y_test".
2. Using seaborn library, plot the training data using the pairplot with kind = "scatter" and hue="target"
3. using Logistic regression model from sklearn, fit the model to the training data, first by using the multinomial technique and second by using one-vs-rest
4. Evaluate the Logistic Regression models using the mean accuracy on the test dataset.

5. Use GridSearch to tune the C value of the logistic regression, and the multi class. use the range ”np.logspace(-10, 10,7)” and multi_class : [’multinomial’, ’ovr’]
6. Comment on why one was better than the other.
7. Lastly visualize the decision boundary of the best performing model on the training dataset in 2D, Hint: ”fit the model on just 2 features and plot”.