# **DPS920 – Assignment 2**

Summer 2025

| Total Mark: | 7.5 marks (7.5% of the total course grade) |
| --- | --- |
| Submission file(s): | * Python files for train and inference * Assignment2.docx (this document with some sample tests) |
| Deadline | * July 28th, 2025 |

If you are unable to complete the assignment on-time for any legit reason, please provide documentation explaining your absence (e.g., an appointment confirmation or a work letter).

Please submit the submission file(s) through Learn@Seneca. Make sure to use GitHub and provide the link to your GitHub account for all your contributions in the box below:

|  |  |
| --- | --- |
| Project GitHub repository: | https://github.com/SychAndrii/DPS920-Assignment2 |

**Please attach some of your test images along with any required explanations in this document.**

1. In Folder Q1, there is a dataset in which we aim to estimate the house price using two features: the number of bedrooms and the basement area.

Use Multiple Linear Regression for this task. Display the coefficients of the model and calculate the MAE (Mean Absolute Error) and MSE (Mean Squared Error). Search about RMSE (Root Mean Squared Error) and explain the trade-offs between these metrics. Finally report RMSE score of your model.

Perform this task using both LinearRegression and SGDRegressor.

Additionally, study the MAPE (Mean Absolute Percentage Error) metric using [this link](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_percentage_error.html), and apply it to evaluate your model.

MAE calculates the average size of the errors by taking their absolute values. Because it doesn’t square the differences, each error contributes equally regardless of its size. This makes MAE straightforward to interpret since it’s in the same units as the target variable, and it isn’t heavily influenced by outliers. However, the downside is that it treats small and large errors the same way, so it doesn’t highlight situations where the model makes particularly big mistakes.

MSE takes a different approach by squaring each error before averaging them. This squaring makes larger errors weigh much more in the final score. That’s useful if large deviations are especially costly or critical to your application. The trade-off is that the result is in squared units, which makes it less intuitive to compare directly to the target variable’s scale.

RMSE, or Root Mean Squared Error, is essentially a compromise between the two. It keeps the squaring step from MSE, so it still penalizes large errors more heavily, but then applies a square root to bring the value back to the original units. This makes RMSE easier to interpret than MSE while still giving more weight to big mistakes compared to MAE.

The trade-off comes down to what you care about: MAE is simple and robust, MSE is stricter with large errors but less interpretable, and RMSE strikes a balance by keeping the original units while still emphasizing big deviations.

1. For the Cat and Dog dataset provided in the Q2 folder, perform classification using all the methods you know and try to achieve the best possible result. Compare the algorithms carefully and tune the parameters so that the best result can be obtained.

Save the trained model and test it on several images from the internet. Was the model able to correctly predict the images?

For the Cat and Dog dataset, I implemented classification using KNeighborsClassifier, LogisticRegression, and SGDClassifier. I experimented with different parameters, including changing n\_neighbors for KNN and adjusting learning parameters for SGD.

* Across all tests, the highest accuracy I was able to achieve was 0.6.
* In most cases, KNN performed the best compared to Logistic Regression and SGD Classifier when predicting images.
* After training, I saved the best-performing model and tested it on several images from the internet. The model was able to predict some images correctly, but due to the limited dataset and only two classes (Cat/Dog), images outside of these categories were still classified as either cat or dog.

A close-up of a dog

AI-generated content may be incorrect.

A dog with its tongue out

AI-generated content may be incorrect.

A cat lying on the floor

AI-generated content may be incorrect.

1. The MNIST dataset is one of the most well-known datasets in the field of image processing. It contains 60,000 images related to handwritten digits from 0 to 9 and is provided as a CSV file in the Q3 folder. In this file, each image is represented as a flattened vector. Classify this dataset using different methods and try to achieve at least 90% accuracy.

The training set contained 59,999 samples with 784 features each, and the test set contained 9,999 samples with the same feature size. The k-nearest neighbors model achieved an accuracy of 97.0%, while the SGD classifier reached 91.6%. The best-performing model was KNN with 97.0% accuracy.

* True label: 4, Predicted: 4
* True label: 9, Predicted: 9
* True label: 5, Predicted: 5