Lab 1: Jupyter Notebooks, Regression, Classification and Genetic Algorithms

September 2023

1 In this lab...

1.1 Topics to cover

- Setting up and running Jupyter Notebooks
- Techniques: Regression, Classification, GA, and GP
- Data-sets (CSV file): samples data, house price regression data, iris flower classification data

1.2 Requirements

- Python installed on your computer.
- Basic Python programming, data handling using Pandas, Jupyter Notebook.
- Libraries: scikit-learn, numpy, matplotlib, seaborn.

1.3 How to launch Jupyter Notebook?

Jupyter Notebook is a web-based interactive computing platform. This tool is going to be used during the entire module and we will use it to execute code and perform data analysis. Please follow the next steps to open any of the Jupyte Notebooks.

1.3.1 Managed Windows PC

1. Open Anaconda
2. Change the current file path to the appropriate directory.
    - **Hint**: For this use the following command *cd filepath* in the command prompt.

3. Open Jupyter Notebook

    - **Hint**: For this use the following command *jupyter-notebook* in the command prompt.

After this Jupyter Notebook will open in the default web browser. You expect to see something similar as below.

1.4 Tasks

1. **Import necessary libraries:** Begin by importing the required libraries, including Pandas, NumPy, Seaborn, matplotlib, and Scikit-learn.

2. **Load the dataset:** Load the dataset into a Pandas DataFrame and inspect the first few rows to understand the data.

3. **Libraries:** scikit-learn, numpy, matplotlib, seaborn.

4. **Data Preprocessing:** Check for any missing values in the dataset and handle them appropriately (e.g., by filling with mean or median values, or removing them). Perform any necessary feature scaling if needed.

5. **Data Visualization:** Create scatter plots or other suitable visualizations to explore the relationships between the features and the target variable.

6. **Data Splitting:** Split the dataset into training and testing sets. Use between 70 to 80

7. **Initialize Model:** Initialize a model using Scikit-learn and train it on the training data.

8. **Model Evaluation:** Evaluate the performance of the trained model on the testing data using appropriate metrics.

9. **Making Predictions:** Use the trained model to make predictions on new data. You can either use sample data or create new data points.

2 Regression

Regression is a statistical and machine learning technique used to model the relationship between one or more independent variables (features) and a dependent variable (target). The goal of regression analysis is to find a mathematical function that best fits the data and can predict the target variable based on the provided features.

2.1 Key Concepts

- **Dependent Variable (Target):** The variable being predicted or explained in regression analysis.
- **Independent Variables (Features):** The variables that are used to predict the dependent variable. They can be one or more in number.
- **Regression Equation/Model:** A mathematical equation that describes the relationship between the independent variables and the dependent variable.
- **Coefficient/Parameter:** The values that are estimated by the regression model to quantify the effect of each independent variable on the dependent variable.
- **Residuals:** The differences between the actual values of the dependent variable and the values predicted by the regression model.
- **Ordinary Least Squares (OLS):** A common method for estimating the coefficients in linear regression by minimizing the sum of squared residuals.
- **Types of Regression:** Different types of regression include:
  - **Linear Regression:** Modeling a linear relationship between the independent and dependent variables.
  - **Multiple Regression:** Extending linear regression to multiple independent variables.
  - **Logistic Regression:** Used for binary classification tasks, not regression in the traditional sense.
  - **Polynomial Regression:** Modeling nonlinear relationships using polynomial functions.
  - **Ridge, Lasso, Elastic Net:** Regularized regression methods that prevent overfitting.

2.2 Applications

- **Predictive Modeling:** Regression is used to predict numerical values, such as predicting stock prices, house prices, or sales.

- **Relationship Analysis:** Regression helps analyze and quantify relationships between variables, such as understanding the impact of advertising spending on sales.
- **Trend Analysis:** Regression is used to identify trends and patterns in time series data.
- **Forecasting:** Regression can be used for predicting future values based on historical data.
- **Risk Assessment:** In finance and insurance, regression can be used to assess risk and predict default rates.

In this lab exercise, you will work with a dataset and apply the concepts of regression using the Scikit-learn library in Python. Linear regression is a simple yet powerful algorithm for predicting numerical values based on input features.

2.3 Dataset

- Use house price, forest fire dataset (provided)
- There are some missing data

2.4 Task 0: Data Visualization

1. Visualization helps to understand the relationship between features (inputs) and labels (outputs) in any given dataset.
2. Follow *sampledata.ipynb* to visualize sample dataset.
    - **Hint:** if at any stage *python* fails to load one of the modules, it might be because it is not installed. To install a module use *pip install MODULENAME* where *MODULENAME* is the name of the module to install.

2.5 Task 1: predict house price

1. Follow *houseprice.ipynb* for data visualization
2. Follow *Regression.ipynb* script

2.6 Report

1. Plot correlation between features.
2. Plot and compare mean square error obtained using Decision Tree, Random Forest, support vector regression.
3. Analyze which technique performed better and why?

3 Classification

Classification is a fundamental concept in machine learning and statistics, involving the process of categorizing data points into predefined classes or categories based on their features. The primary goal of classification is to develop a model or algorithm that can accurately assign new, unseen data points to the appropriate class based on patterns learned from labeled training data.

3.1 Key Concepts

- **Classes or Categories:** The distinct groups into which data points are to be categorized. Each class represents a specific outcome or label.
- **Features:** The measurable attributes or characteristics of data points that are used to make classification decisions.
- **Training Data:** Labeled data used to train a classification model. Each data point is associated with its correct class label.
- **Model or Algorithm:** The method or technique used to learn the relationships between features and classes from the training data.
- **Predictions:** Applying the trained model to new, unseen data points to predict their class labels.
- **Evaluation Metrics:** Metrics such as accuracy, precision, recall, F1score, and confusion matrix are used to assess the performance of classification models.

3.2 Types of Classification

- **Binary Classification:** Assigning data points to one of two possible classes. Examples include spam detection and medical diagnosis.
- **Multi-class Classification:** Categorizing data points into more than two classes. Each data point belongs to only one class.
- **Multi-label Classification:** Assigning multiple labels to each data point. A data point can belong to more than one class simultaneously.
- **Imbalanced Classification:** When the distribution of classes in the dataset is heavily skewed, requiring special techniques to handle class imbalance.

3.3 Applications

- **Image Classification:** Categorizing images into classes, such as identifying objects or animals in images.

- **Medical Diagnosis:** Identifying diseases or conditions based on patient data and medical records.
- **Fraud Detection:** Detecting fraudulent transactions or activities based on historical data.
- **Customer Churn Prediction:** Predicting whether customers are likely to leave a service or subscription.
- **Language Identification:** Determining the language of a given text document.

In this lab exercise, you will work with a dataset and apply the concepts of classification using the Scikit-learn library in Python.

3.4 Dataset

We will use a dataset containing information about various flowers. The dataset has the following features:

- Sepal length (in centimeters)
- Sepal width (in centimeters)
- Petal length (in centimeters)
- Petal width (in centimeters)
- Species of the flower (target variable)

3.5 Task 1: predict classification of flowers

1. Follow *IrisVisualization.ipynb* for data visualization
2. Follow *Classification.ipynb* script

3.6 Report

1. Create a new notebook
2. Use diabetes dataset
3. Visualize dataset
4. Plot confusion matrix.
5. Plot and compare accuracy obtained using Decision Tree, Random Forest, support vector classifier, and KNN.
6. Analyze which technique performed better and why.

4 Genetic Algorithms

Genetic Algorithms are a family of AI methods that are inspired by the principles of evolution by natural selection. They can be thought of as a specialised sort of search algorithm, whereby good solutions are found by randomly changing the properties of members of the population according to the "fitness" of individuals. The theory, as originally proposed by Charles Darwin and Alfred Russell Wallace is

- **variation** In all species individuals vary in their genetic makeup, which makes individuals vary from each other
- **inheritance** Individuals pass genes to their offspring, so children are similar to parents, *but not identical*
- **selection** those children which are a "better fit" to their environment are more likely to survive to reproduce
- **time** over time, these factors cause species to adapt to their environment

Genetic algorithms try to emulate these phenomena programatically in order to solve a wide range of problems. The solution is encoded on a *genome* - a (usually) numeric description of the solution to the given problem, and it is tested by a *fitness function* (also known as an objective function)

4.1 Key Concepts

- **Populations of solutions:** The algorithm is inherently parallel - many solutions are available at any one time
- **Generations of populations:** The algorithm iterates over many generations, creating new populations from the previous generation.
- **Fitness function** Each individual is assigned a fitness value, which influences whether that individual is selected to generate individuals in the next generation
- **Variation by inheritance (crossover)** A new individual is created by combing the genomes from the parent genomes in some way
- **Variation by poor copying (mutation)** Individual elements in the new genome are randomly adjusted, emulating errors in seen in DNA copying in biology

4.2 Tasks

1. Load the Jupyter Notebook titled *GeneticAlgorithmsLab.ipynb* and save it with a new name.

2. In your new notebook, run the basic algorithm as provided for the "one-max" problem
3. Experiment with different parameter values for mutation rate `R_MUT`, crossover rate `R_CROSS` and population size `N_POP` until you have found reasonably well-performing values
4. Adapt the script to return the number of evaluations (gen*npop) for a given run and ONLY output a statement reporting this
5. Calculate the search space size for genome length 20 to 50 bits and plot this
6. Compare different effects of high and low values of `R_MUT` and `R_CROSS` for a sensible set of bit lengths

4.3 Report

Your finished notebook should contain:

1. Adaptation to the provided algorithm to return the number of evaluations
2. Plot of genome length on x axis and search space size on y axis
3. Plot of number of evaluations for several replicats of a range of values of
4. Discussion and interpretation of the results you have generated
5. Once your notebook is completed, print to pdf