

Data Science Programming FINAL ASSIGNMENT

Student Name: Raneem Yahya Sa’deh

Student ID: 22210016

Instructor Name: Eng. Islam Al Omari / Eng. Waed

Course Name: Data Science

HTU Course No: 10204281

BTEC Unit No:

Academic Year Summer 2023

A grey logo on a black background

Description automatically generated with low confidence

A graph paper with lines

Description automatically generatedA black background with a black square

Description automatically generated with medium confidence

# **Data Science Programming**

A yellow and blue snake logo

Description automatically generated

Contents

[Data Science Programming 2](https://hatuniversity-my.sharepoint.com/personal/22210016_htu_edu_jo/Documents/DataScience&Python(FINAL).docx#_Toc145337648)

[What is Data Science? 4](#_Toc145337649)

[Data Structure in Python 5](#_Toc145337650)

[**Built-in data structures:** 5](#_Toc145337651)

[**Tuples:** 5](#_Toc145337652)

[**Lists:** 6](#_Toc145337653)

[**Sets:** 7](#_Toc145337654)

[**Dictionary:** 7](#_Toc145337655)

[**User-defined Data Structure:** 8](#_Toc145337656)

[Python libraries: 9](#_Toc145337657)

[**NumPy:** 9](#_Toc145337658)

[**Pandas:** 10](#_Toc145337659)

[**SciKit-Learn:** 10](#_Toc145337660)

[Plotting and Visualization libraries: 11](#_Toc145337661)

[**Matplotlib:** 11](#_Toc145337662)

[What is Python? 13](#_Toc145337663)

[Load And Prepare The Data 14](#_Toc145337664)

[Compare the different models: 17](#_Toc145337665)

[Visualization 17](#_Toc145337666)

[**Box Plot** 17](#_Toc145337667)

[**Bar Chart** 20](#_Toc145337668)

[**Radar Chart** 21](#_Toc145337669)

[Thar Choice of Data Structure 23](#_Toc145337670)

[**NumPy** 23](#_Toc145337671)

[**Pandas** 24](#_Toc145337672)

[**Sklearn** 24](#_Toc145337673)

[The effectiveness of different models 25](#_Toc145337674)

[**Random Forest:** 25](#_Toc145337675)

[**K-Nearest Neighbors (KNN):** 25](#_Toc145337676)

[**Decision Tree Classifier:** 25](#_Toc145337677)

[Recommendations 26](#_Toc145337678)

[References 27](#_Toc145337679)

# **A hand holding a glowing circle with icons and text Description automatically generatedWhat is Data Science?**

**Data science combines math and statistics, specialized programming, advanced analytics, artificial intelligence (AI), and machine learning with specific subject matter expertise to uncover actionable insights hidden in an organization’s data,**

**It was created by Guido van Rossum and first released in 1991.**

# **Data Structure in Python**

Data structures are essential for organizing and processing data in programming. Python offers built-in data structures and user-defined data structures, each with unique characteristics and uses. Here are some commonly used data structures in Python:

## **Built-in data structures:**

1. Tuple
2. List
3. Set
4. Dictionary

### **Tuples:**

One of the built-in data structures and it is unlike lists because tuples come with limited functionality. Tuples are immutable they cannot be deleted, added, or modified once they are created.

Example: tuple\_A = (0.2, 3.4, 8.3)

**The elements in the tuple could be any data type like (integer, float, Boolean).**

**The elements can be accessed with square brackets [ ].**

Example: tuple\_B = tuple([0.2, 3.4, 8.3)]

**We also can make a nested tuple:**

Example:

Nested\_tuple = ((3, 4, 5), (7, 8, 9))

* **But we can change the tuple values by:**

1. **Concatenate the tuple by using the (+):**

Example:

Tuple\_M = (4, 5, ‘V’) + (None, 8) + (“Hello”)

1. **Unpacking tuples:**

The number of values on the left side must be the same in the right side, if not there will be a ValueError.

But if we don’t know the number of the values or care about the rest values we can just write (\*rest).

Example:

Numbers = 1, 2, 3, 4, 5, 6

a, b, c, \*rest = numbers

a  Output 1

c  Output 3

\*rest  Output [4, 5, 6]

1. **Using (sorted()) which is a Built-in function:**

Example:

Tuple\_sort = tuple(sorted(tuple\_D))

### **Lists:**

They are commonly used to hold a sequence of elements, such as numbers, strings, objects, or any combination thereof. They provide a way to organize and manage data, making it easier to manipulate and process information.

Creating a list is essentially creating a Python object of a specific type. When you create a list, all the items in the list should be put in square brackets and separated by commas.

1. Lists maintain the order in which elements are added. This means that the position of an element within the list is important and can be used to access specific elements, and the first element index will be zero.
2. Lists are often mutable, meaning that you can modify the elements they contain. You can remove, change, or add elements after you create the list.
3. You can use different data types in the list. For instance, a single list can contain a mix of integers, floats, strings, and other data types.

Example:

list\_A = [0, 1, 2, 3, 4]

list also can be nested, so it can contain any type of object. It can include another list or a sub-list.

Example:

list\_C = [0, ’B’, [2,6,9], ‘’Raneem’’]

You can extract a subset of a list using slicing.

Example:

list\_slice = my\_list[1:4]

The output will give me the items from index 1 to 3

You can remove items from a list using these methods: pop(), remove(), or del.

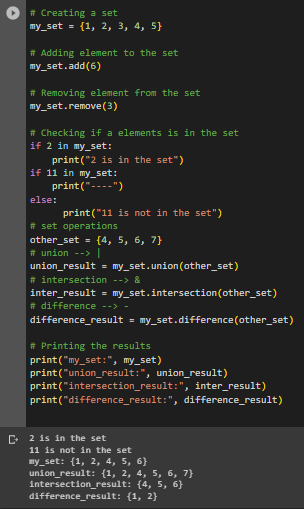
removed\_item = my\_list.pop()

my\_list.remove("hello")

del my\_list[1]¬

You can sort the list by using the sort function.

Example:

List\_sort.sort()

### **Sets:**

**The set can be created by using curly brackets { }**

* A set does not allow duplicate elements. Each element within a set is unique.
* Unlike lists or arrays, sets are not organized in any specific order. The elements do not have an inherent position or index.
* Sets support basic set operations, including union, intersection, and difference. These operations are inspired by mathematical set theory.

(Union): Combines two sets, resulting in a new set that contains all unique elements from both sets.

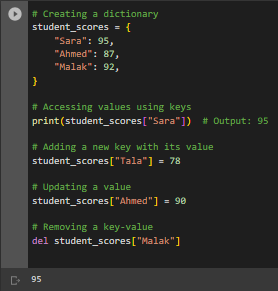
(Intersection): Creates a new set containing only the elements that are present in both input sets.

(Difference): Produces a new set with elements from the first set that are not present in the second set.

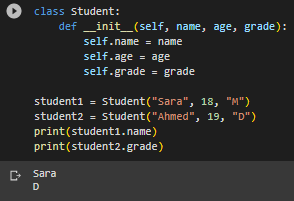
* Sets provide an efficient way to check if a specific element is present in the set. This operation is usually faster than searching for an element in other data structures like lists.
* Elements can be added to a set, and existing elements can be removed. When adding an element that already exists in the set, the set remains unchanged.

### **Dictionary:**

**dictionary is a fundamental data structure used to store and organize data in a way that allows for efficient retrieval and manipulation. It's often referred to as mapping because it has keys with values each key in a dictionary maps to a specific value.**

* **** A dictionary consists of a collection of key-value pairs. The key is a unique identifier, and the value is the associated data. Keys are usually strings, but they can be of any data type, such as integers or even other data structures.
* Dictionaries are usually unordered, which means that the key-value pairs are not stored in a specific order. This is different from lists or arrays, where elements are indexed by their position in order.
* Dictionaries are mutable, which means you can remove, add, or update key-value pairs after the dictionary is created. This makes dictionaries versatile for storing and modifying data dynamically.
* Dictionaries are defined using curly braces {}.

## **User-defined Data Structure:**

****

**In programming, a user-defined data structure is a customized method of classifying and storing data that is created by the programmer. User-defined data structures. Unlike built-in data structures like lists, arrays, dictionaries, and sets, are made by the programmer based on their specific needs.**

# **Python libraries:**

Python libraries are collections of pre-written code that offer a variety of tools and methods to help programmers finish their work without having to start from scratch. These libraries include modules, classes, and functions that make the development process easier and faster by providing pre-made answers to typical programming problems.

**Popular Python libraries:**

|  |  |
| --- | --- |
| **provides a potent N-dimensional array object and is the core Python library for numerical computing. On GitHub, it has almost 18,000 comments and a 700-person active community. It is a general-purpose array-processing software that offers capabilities for working with high-performance multidimensional objects known as arrays. By offering these multidimensional arrays as well as functions and operators that work effectively on these arrays, NumPy partially overcomes the slowness issue.** | A screenshot of a computer program  Description automatically generated |

## **NumPy:**

**Features:**

* Provides fast, precompiled functions for numerical routines.
* Array-oriented computing for better efficiency.
* Supports an object-oriented approach.
* Compact and faster computations with vectorization.

**Applications:**

* Extensively used in data analysis.
* Creates powerful N-dimensional array.

## **Pandas:**

|  |  |
| --- | --- |
| **is essential to the life cycle of data science. Along with NumPy in matplotlib, it is the most well-known and commonly used Python module for data research. It is widely used for data analysis and cleansing, with almost 17,00 comments on GitHub and an active community of 1,200 contributors. Pandas offers quick, adaptable data structures, like data frame CDs, that make it simple and natural to work with structured data, and it is used for general data wrangling and data cleaning.** | A screenshot of a computer program  Description automatically generated |

**Features:**

• Contains high-level data structures and manipulation tools.

• High-level abstraction.

• Enables you to create your own function and run it.

• Eloquent syntax and rich functionalities that give you the freedom to deal with missing data.

## A close up of a logo Description automatically generated**SciKit-Learn:**

**a library for machine learning that offers practically all the algorithms I would require. NumPy and SciPy can interpolate Scikit-learn data.**

Example importing KNeighborsClassifier from SciKit-Learn:

from sklearn.neighbors import KNeighborsClassifier

classifier = KNeighborsClassifier(n\_neighbors=5)

classifier.fit(X\_train , y\_train)

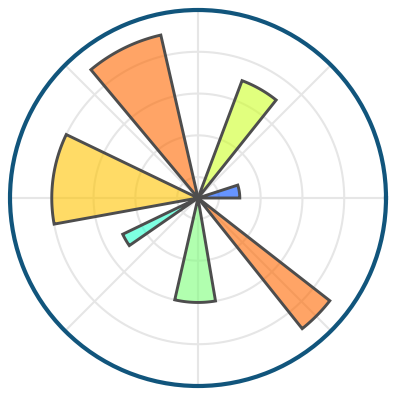
y\_pred\_KNN = classifier.predict(X\_test)

A chart with numbers and a line

Description automatically generated with medium confidence

**And also I use it to convert Categorical values (‘A’, ‘B’) into Numeric values (0, 1).**

# **Plotting and Visualization libraries:**



## **Matplotlib:**

**It is a popular Python library for creating static interactive visualizations in various formats. You can use a lot of tools for creating plots, charts, radar, and graphs to help users visualize data and present their findings effectively. It was created by John D. Hunter in 2003. Matplotlib is usually used in fields such as data science, scientific research, engineering, and more.**

**Features:**

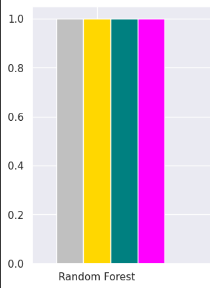
* High-Quality Plots: Matplotlib produces high-quality, publication-ready plots and you can create box plots, bar charts, histograms, radar charts, and more.
* Matplotlib allows you to customize your plots, including line styles, colors, markers, labels, titles, fonts, and more. This makes it highly versatile for different visualization needs.
* Plots help to understand trends, and patterns and make correlations with information.

1. **Box Plot:**

It uses boxes and lines to depict the distributions of one or more groups of numeric data.

|  |  |
| --- | --- |
| Outliers very far from the data num  Upper Whisker  Upper Quartile  Median  Lower Quartile  Lower Whisker |  |

1. **Bar Chart:**

Bar chart or “bar graph” is a common and widely used data visualization technique for displaying categorical data.

* **It could be in two shapes: Vertical or Horizontal.**
* **It could be used for comparing the values of different groups. It makes it easy to identify which category has the highest or lowest values.**
* **What does it contain?**

Bars: The rectangular bars represent the data values.

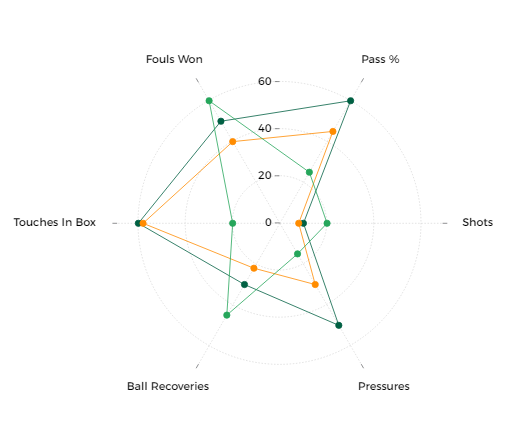
Axes: has two axes, the x-axis (horizontal) and the y-axis (vertical).

Labels: Bar charts include labels for the categories or groups along the x-axis.

Legend: If you are comparing multiple datasets in a single chart.

1. **Radar Chart:**

It is a data visualization technique used to display data in a two-dimensional plane, and it looks like a spider net or a star.

* **The critical elements of a radar chart are center, radial axes, axis labels, scale, gridlines, data points, data lines, and colors:**

1. Center: This is the starting point where all the radial axes extend.

2. Radial axes: lines come from the central point, representing the compared categories.

3. Axis labels: Text labels positioned at the end of each radial axis.

4. Scale: A graduated scale along each radial axis that indicates the values for each variable.

5. Gridlines: Circular or polygonal lines connecting the scale markings on each radial axis help visualize the distribution and relationship of values across different variables.

6. Data points: The actual values of each variable are plotted on the corresponding radial axis.

7. Data lines: Lines connecting the data points for each variable until it closes again.

8. Fill color: The area enclosed by the data lines or polygons can be filled with color. You can use different colors to compare multiple things on the same chart.

# **What is Python?**

Python is a high-level programming language known for its simplicity and readability. It was created by Guido van Rossum in 1991, Python has become one of the most popular programming languages in the world. Its design philosophy emphasizes code readability and a clean, easily understandable syntax.

* Python has an easy-to-read syntax which makes it very good for beginners to start.
* It uses indentation to define code blocks, which enhances code readability.
* Python is an interpreted language so you can run the code without a compiler, this makes it highly accessible for rapid development.
* Python has numerous third-party libraries like NumPy, Pandas, and TensorFlow.
* Python is available on different platforms such as Windows, macOS, and Linux, and that makes it suitable for cross-platform development.
* Python has acquired popularity in data science and machine learning because of libraries like NumPy, pandas, and scikit-learn.

# **Load And Prepare The Data**

|  |  |  |
| --- | --- | --- |
| Column name | Description | Justification |
| Event | There are no null values | I did not do anything |
| Time | I use the Mean function to fill it | maintain the statistical distribution of the data in that column, It provides a reasonable estimate of what the missing value might be |
| S\_Node | There are no null values | I did not do anything |
| Node\_id | I Dropped that column | because the correlation between them is 1.0 |
| Rest\_Energy | use the Mean function to fill it | maintain the statistical distribution of the data in that column, It provides a reasonable estimate of what the missing value might be |
| Trace\_Level | I Dropped that column | I drop ‘Trace\_Level’ column because it has only one value which will not help in the models |
| Mac\_Type\_Pckt | I Dropped that column | I drop 'Mac\_Type\_Pckt' column because it has only one value which will not help in the models |
| Source\_IP\_Port | There are no null values | I did not do anything |
| Des\_IP\_Port | I Dropped that column | I drop ‘Des\_IP\_Port’ column because it has only one value which will not help in the models |
| Packet\_Size | I fill the null values with 48.0 | Because 48.0 is repeated 41276 time |
| TTL | There are no null values | I did not do anything |
| Hop\_Count | There are no null values | I did not do anything |
| Broadcast\_ID | I Dropped that column | because the correlation between them is 1.0 |
| Dest\_Node\_Num | Giving it the previous value | Because it is a value distributed in groups |
| Dest\_Seq\_Num | Giving it the previous value | Because it is a value distributed in groups |
| Src\_Node\_ID | There are no null values | I did not do anything |
| Src\_Seq\_Num | There are no null values | I did not do anything |
| Behaviour | There are no null values | I did not do anything |
| Type | I Dropped that column | drop ‘type' column because it has only one value which will not help in the models |

**Visualization between the Behaviour and the Dest\_Node\_Num columns:**

**A graph of bar graph

Description automatically generated with medium confidence**

|  |  |  |
| --- | --- | --- |
| Model Name | Description | Picture for how it works |
| Random Forest | **It can build classification or regression models, it works by creating multiple decision trees, and the final output is determined by the most chosen among the trees.** |  |
| KNN | **It’s a classifier algorithm used in data mining and machine learning where it works in how similar data from other data is.**  **First, we calculate the distance we have we will see which has the smallest distance, and then we see which data appears more so our new data will have its value.** |  |
| DTC | **It can build classification or regression models, it works by break down the data set into smaller subsets, It can handle both categorical and numerical data.** **The final result is a tree with decision nodes ( has two or more branches) and leaf nodes (decision).** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Approach no.** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| **RF** | **0.99254** | **0.9907397526722063** | **0.9674703092531949** | **0.9772125066242713** |
| **KNN** | **0.9572** | **0.9122916206325382** | **0.8422984418105965** | **0.8758507316677564** |
| **DTC** | **0.99982** | **0.9996686152269421** | **0.9993185912328331** | **0.9994931924856644** |

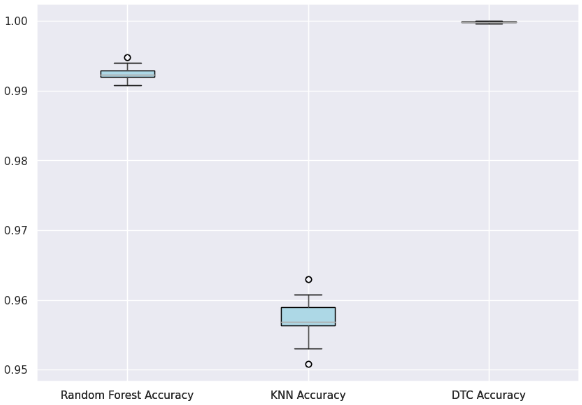
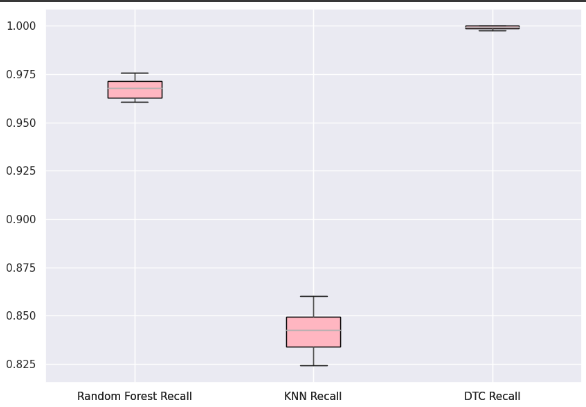
# **Compare the different models:**

# **Visualization**

## **Box Plot**

It uses boxes and lines to depict the distributions of one or more groups of numeric data.

|  |  |
| --- | --- |
| Importing the mat library |  |
| Making Three dictionary for the models |  |
| Get the Data Frame ready |  |
| Creating the box plot and customize its size and labels |  |

A graph with lines and dots

Description automatically generated with medium confidence

A graph with a green line

Description automatically generated

## **Bar Chart**

Bar chart or “bar graph” is a common and widely used data visualization technique for displaying categorical data.

|  |  |
| --- | --- |
| Importing the mat library | A black and white sign with white text  Description automatically generated |
| Making Three pref of the average for the models |  |
| Goin them in one data frame |  |
| customize the bars and add the legend |  |

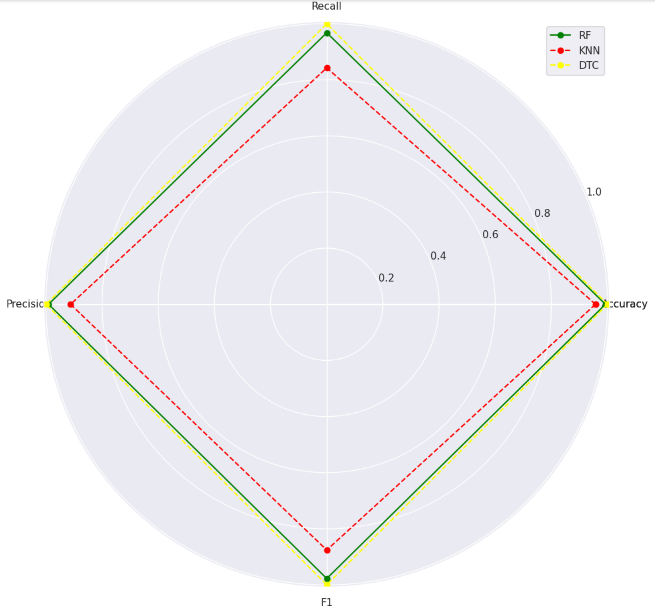
A graph of different colored bars

Description automatically generated

## **Radar Chart**

It is a data visualization technique used to display data in a two-dimensional plane, and it looks like a spider net or a star.

|  |  |
| --- | --- |
| Importing the mat library | A black and white sign with white text  Description automatically generated |
| transposes the Data Frame df\_pref and stores the result in df\_pref2 |  |
| ‘measure’ contains the performance measure names, and the data is converted into a list and extracted from ‘df\_pref2’ for the three models. |  |
| I create evenly spaced angles around a circle, and create a polar subplot ‘ax’ is added to figure. |  |
| Using ‘ax.plot’ to create lines for each model with different colors and labels. |  |



# **Thar Choice of Data Structure**

## **NumPy**

|  |  |
| --- | --- |
| I used the NumPy library to create empty NumPy arrays to store accuracy, recall, precision, and f1-score values, and then append values to them. | A black background with white text  Description automatically generated |
| Appending the accuracy of the test of (X,y) | A screen shot of a computer code  Description automatically generated |
| I used the NumPy library to calculate the means of lists or arrays |  |
| I create an array ‘x’ using the NumPy arange function, len(df\_pref) shows the length of the array, which is based on the number of items in the df\_pref DataFrame. This array is used to specify the x-coordinates of the bars in the bar plot. |  |

## **Pandas**

|  |  |
| --- | --- |
| I used Pandas library to create a Data Frame several times by giving the ‘’index’ a value and the ‘column’ a value. |  |
| I use it to concatenate them horizontally (along the columns) into a single Data Frame named df\_pref. |  |

## **Sklearn**

**a library for machine learning that offers practically all the algorithms I would require.**

|  |  |
| --- | --- |
| I used sklearn to import labelEncoder to change the value from string to integer. |  |
| I used it to import the Decision Tree Classifier |  |

# **The effectiveness of different models**

## **Random Forest:**

Average Accuracy: 0.99254

Average Recall: 0.96747

Average Precision: 0.99074

Average F1-Score: 0.97721

Random Forest is a highly effective model with a very high accuracy (99.25%), good recall (96.75%), high precision (99.07%), and a strong F1-score (97.72%). It appears to perform very well across all metrics.

## **K-Nearest Neighbors (KNN):**

Average Accuracy: 0.9572

Average Recall: 0.84230

Average Precision: 0.91229

Average F1-Score: 0.87585

KNN also demonstrates effectiveness, but it has lower accuracy (95.72%) compared to Random Forest. It has a decent recall (84.23%) and precision (91.23%) but a slightly lower F1-score (87.59%) compared to Random Forest. KNN may be a good choice if computational efficiency is not a concern, and you prioritize precision.

## **Decision Tree Classifier:**

Average Accuracy: 0.99982

Average Recall: 0.99932

Average Precision: 0.99967

Average F1-Score: 0.99949

The Decision Tree Classifier is performing exceptionally well. It has the highest accuracy (99.98%) among the three models, and its recall (99.93%), precision (99.97%), and F1-score (99.95%) are also outstanding.

**What is the best model for me?**

The choice of the best model depends on your specific use case, but all three models are effective, but the Decision Tree Classifier appears to be the most accurate.

# **Recommendations**

**Feature selection in machine learning**

1. To reduce the dimensionality of feature space.

2. To speed up a learning algorithm.

3. To improve the predictive accuracy of a classification algorithm.

**Common techniques for feature selection**

**Filter methods:** These include the Chi-square test, which assesses relationships between categorical variables.

**Wrapper methods:** An iterative approach like forward selection that adds features that improve model performance until no further improvement is observed.

**Embedded methods:** Tree-based techniques like Random Forest and Gradient Boosting, which provide feature important scores for selection.

**Data** is the foundation of any machine learning project. Ensure my data is accurate, relevant, and representative of the problem you are trying to solve. Address issues like **data imbalance**.

# **References**

Wikipedia. (n.d.). Random forest. Retrieved from <https://en.wikipedia.org/wiki/Random_forest>

DataCamp. (n.d.). 21 Essential Python Tools. Retrieved from <https://www.datacamp.com/tutorial/21-essential-python-tools>

Analytics Vidhya. (2020). Feature Selection Techniques in Machine Learning. Retrieved from <https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/>

IBM. (n.d.). Data Science. Retrieved from <https://www.ibm.com/topics/data-science>

Visme. (n.d.). How to Create a Radar Chart. Retrieved from <https://visme.co/blog/radar-chart/#:~:text=The%20critical%20elements%20of%20a,data%20lines%20and%20fill%20colors>

GeeksforGeeks. (n.d.). Python – Introduction to Matplotlib. Retrieved from <https://www.geeksforgeeks.org/python-introduction-matplotlib/>