**24AI602 - Machine Learning**

**Case Study on KNN and Linear Regression**

**Varadharajan K**

**CB.SC.P2AIE25030**

Link to access my Colab Notebook (Implementation of both KNN and Linear Regression):

<https://colab.research.google.com/drive/1JlCZ2woLxJ-yk4dw5ytqliXtIRrFfiuJ?usp=sharing>

**Drug classification case study using KNN**

**What is a KNN algorithm?**

K-Nearest Neighbours (KNN) is a supervised machine learning algorithm generally used for classification, but can also be used for regression tasks. It works by finding the "k" closest data points (neighbours) to a given input and makes a prediction based on the majority class (for classification) or the average value (for regression). Since KNN makes no assumptions about the underlying data distribution, it is a non-parametric and instance-based learning method.

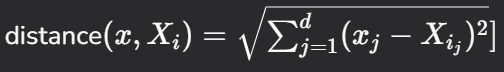
**What is 'K' in K Nearest Neighbour?**

In the k-Nearest Neighbours algorithm, k is just a number that tells the algorithm how many nearby points or neighbours to look at when it makes a decision.

**Distance Metrics Used in the KNN Algorithm**

**Euclidean Distance**

Euclidean distance is defined as the straight-line distance between two points in a plane or space.



**Manhattan Distance**

This is the total distance you would travel if you could only move along horizontal and vertical lines like a grid or city streets.



**Working of the KNN algorithm**

Step 1: Selecting the optimal value of K

Step 2: Calculating distance

Step 3: Finding Nearest Neighbours

Step 4: Voting for Classification or Taking Average for Regression

**Case Study Objective:**

To perform KNN on the patient-drug dataset and predict the outcome of the drug type that is suitable for the patient.

**Dataset:**

The Kaggle dataset is a CSV file consisting of about 200 patient records with details on their Age, Gender, BP, cholesterol, Na\_to\_K, and the drug type(A, B, C, X, Y).

Link to the dataset: [Drug Classification](https://www.kaggle.com/datasets/prathamtripathi/drug-classification/data)

**Implementation:**

It was implemented in Google Colab using Python, with some important libraries like: pandas, seaborn, numpy, scikit-learn and matplotlib.

* The dataset was loaded and checked for any null values and data type mismatches.
* Some exploratory data analysis was performed to look at the data distribution with respect to features.
* Feature data encoding was performed to ensure we can perform KNN on our dataset.
* The dataset was split into test and train to train the KNN model and subsequently test the model’s prediction on the data.
* Oversampling was done on the dataset to avoid overfitting since the number of 'DrugY' is more than other types of drugs.
* The KNN model was implemented with:
  + The model from the scikit-learn package.
  + The model built from scratch with 3 different distance metrics.
* The accuracies of the models were computed and compared.
* The models were run on different K values and plotted on a graph to identify which K value the model had the best accuracy.
* The confusion matrix and the classification reports were also generated to get the best inference of the models.

**Inferences and Results:**

The KNN model using scikit-learn gave an accuracy of 83.33% with a K value of 20. The same predictive accuracy was achieved with the KNN model that was built from scratch with the Euclidean distance metric. While Manhattan and Cosine distance metrics gave accuracies of 93.33% and 85% respectively, with a K value of 20.

The highest accuracy recorded for various values of K was 95%. Overall, the performance of the KNN model was good, and the model can rightly predict the outcome of the drug type that is suitable for the patient.

**Electricity cost prediction using Linear Regression**

Linear regression is a type of supervised machine-learning algorithm that learns from the labelled datasets and maps the data points with the most optimised linear functions, which can be used for prediction on new datasets. It assumes that there is a linear relationship between the input and output, meaning the output changes at a constant rate as the input changes. This relationship is represented by a straight line.

**Why Linear Regression is Important?**

* Simplicity and Interpretability
* Predictive Ability
* Basis for Other Models
* Efficiency
* Widely Used
* Analysis

**What is the Best Fit Line in Linear Regression?**

The best-fit line is the straight line that most accurately represents the relationship between the independent variable (input) and the dependent variable (output). It is the line that minimises the difference between the actual data points and the predicted values from the model.

**Types of Linear Regression**

When there is only one independent feature, it is known as Simple Linear Regression or Univariate Linear Regression.



When there is more than one feature, it is known as Multiple Linear Regression or Multivariate Regression.



**Evaluation Metrics for Linear Regression:**

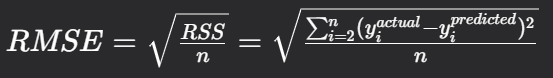
**Mean Squared Error (MSE)** is an evaluation metric that calculates the average of the squared differences between the actual and predicted values for all the data points. The difference is squared to ensure that negative and positive differences don't cancel each other out.



**Mean Absolute Error** is an evaluation metric used to calculate the accuracy of a regression model. MAE measures the average absolute difference between the predicted values and actual values.



The square root of the residuals' variance is the **Root Mean Squared Error**. It describes how well the observed data points match the expected values or the model's absolute fit to the data.



**R-squared** is a statistic that indicates how much variation the developed model can explain or capture. It is always in the range of 0 to 1. In general, the better the model matches the data, the greater the R-squared number.

**Lasso Regression (L1 Regularisation)**

Lasso Regression is a technique used for regularising a linear regression model. It adds a penalty term to the linear regression objective function to prevent overfitting.



**Ridge Regression (L2 Regularisation)**

Ridge regression is a linear regression technique that adds a regularisation term to the standard linear objective. Again, the goal is to prevent overfitting by penalising large coefficients in the linear regression equation. It is useful when the dataset has multicollinearity, where predictor variables are highly correlated.



**Case Study Objective:**

To perform Linear Regression on the electricity dataset, which encompasses a variety of factors influencing energy consumption and expenditure across different types of sites, and predict its cost.

**Dataset:**

The Kaggle dataset is a CSV file consisting of about 10000 records with details on site area, water consumption, recycling rate, utilisation rate, air quality index, issue resolution time, resident count, and electricity cost.

Link to the dataset: [Electricity cost prediction dataset](https://www.kaggle.com/datasets/shalmamuji/electricity-cost-prediction-dataset/data)

**Implementation:**

* The dataset was loaded and checked for any null values and data type mismatches.
* Some exploratory data analysis was performed to look at the data distribution with respect to features.
* Feature data encoding was performed to ensure we can perform Linear Regression on our dataset.
* The dataset was split into test and train to train the Linear Regression model and subsequently test the model’s prediction on the data.
* The feature importance plot was depicted with the coefficient value, which showed which feature had the greatest impact on the electricity.
* The model’s performance was computed with RMSE, R², and the MAE values to get the best inferences.
* The ridge and lasso regressions were also performed by creating a pipeline for the same.

**Inferences and Results:**

The Linear regression model produced values of 312.06, 0.92, 245.65 for RMSE, R², and MAE. While ridge regression model achieved the same metrics as that of Linear regression, lasso regression showed slightly better performance with marginally lower RMSE and MAE, and a slightly higher R² compared to linear and ridge regression. All three models exhibited very similar performance on the dataset.

Overall, the performance of the Regression model was good, and the model can rightly predict the electricity cost across different site types, encompassing a variety of factors.

**What is Vibe coding?**

Vibe coding is an emerging software development practice that uses artificial intelligence (AI) to generate functional code from natural language prompts, accelerating development, and making app building more accessible, especially for those with limited programming experience.

The term, coined by AI researcher Andrej Karpathy in early 2025, describes a workflow where the primary role shifts from writing code line-by-line to guiding an AI assistant to generate, refine, and debug an application through a more conversational process. This frees you up to think about the big picture, or the main goal of your app, while the AI handles writing the actual code.

In practice, vibe coding is generally applied in two main ways:

"Pure" vibe coding: In its most exploratory form, a user might fully trust the AI's output to work as intended. As Karpathy framed it, this is akin to "forgetting that the code even exists," making it best suited for rapid ideation or what he called "throwaway weekend projects," where speed is the primary goal.

Responsible AI-assisted development: This is the practical and professional application of the concept. In this model, AI tools act as a powerful collaborator or "pair programmer." The user guides the AI but then reviews, tests, and understands the code it generates, taking full ownership of the final product.

**What is Psephology and who is a Psephologist?**

Psephology is the statistical study and analysis of electoral history, polling data, and voter behavior to understand voting patterns and predict future election outcomes.

A psephologist is an expert in this field who analyzes historical voting figures, public opinion polls, demographic trends, and other data to explain electoral results and forecast future ones.

**What they do?**

1. **Analyze Data:**

Psephologists meticulously examine past voting records, campaign finance information, and current opinion polls.

1. **Identify Trends:**

They look for patterns in voter preferences, variations in electoral turnout, and demographic voting trends.

1. **Explain Results:**

They use this analysis to explain why particular election results occurred.

1. **Predict Outcomes:**

They develop predictive theories for future elections using statistical models and data analysis.

**Tools and knowledge required:**

**Statistical Analysis:** A strong background in statistics is essential.

**Demographics:** Understanding the characteristics of different populations is crucial.

**Political Science:** Knowledge of electoral systems and voting behavior is necessary.

**Data Analysis Skills:** The ability to interpret and make sense of complex data sets.

Who employs them?

**Political Parties:** To understand their electoral support and strategize for future campaigns.

**Polling Firms:** To develop and validate their polling methodologies and results.

**Political Consultants:** To provide data-driven advice to candidates and parties.

**Media:** To provide insights and analysis during elections.