24AI602 - Machine Learning

Case Study on KNN and Linear Regression

P K SANGAMESWAR

CB.SC.P2AIE25016

Link to access my Github (Implementation of both KNN and Linear Regrssion):

Linear Regression: https://github.com/PK-SANGAMESWAR/car-selling_price-prediction.git

KNN: https://github.com/PK-SANGAMESWAR/diabetes-prediction-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.

$$\text{distance}(x, X_i) = \sqrt{\sum_{j=1}^{d} (x_j - X_{i_j})^2]}$$

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.

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

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

DIABETES PREDICTION USING KNN

CASE STUDY OBJECTIVE:

The main objective of this notebook is to predict whether a person will have diabetes or not using the K-Nearest Neighbors (KNN) algorithm.

- It is a supervised machine learning classification problem.
- Input: Medical attributes of patients (like glucose level, BMI, insulin, etc.).
- Output: 1 → person has diabetes, 0 → person does not have diabetes.

The model is built to help in early detection and decision support for diabetes diagnosis.

DATASET:

The dataset used is the PIMA Indians Diabetes dataset (commonly available on Kaggle / UCI repository).

Link to the dataset: https://www.kaggle.com/datasets/mathchi/diabetes-data-set

IMPLEMENTATION:

The main steps of the implementation:

1. Importing libraries: numpy, pandas, seaborn, matplotlib, sklearn (train_test_split, StandardScaler, KNN, metrics).
2. Data cleaning: replaced invalid zeros with NaN → filled with column mean.
3. Exploratory Data Analysis (EDA):
    - Heatmap of correlations between features.
    - Count plot of Age vs. Outcome.
4. Feature selection: used all 8 independent features.
5. Train-test split: 80% training, 20% testing.
6. Feature scaling: StandardScaler to normalize all feature values.
7. Model training:

    - KNN classifier with parameters:
        - n_neighbors = 11
        - p = 2 → Euclidean distance
    - Fitted on training set.
8. Prediction: Predictions made on test set.
9. Evaluation metrics:
    - Confusion matrix
    - Accuracy score
    - F1 score

INFERENCES AND RESULTS:

 The KNN model successfully learned to classify patients as diabetic / non-diabetic.

 Evaluation metrics observed:

- Accuracy: ~81% (depending on random split and scaling).
- F1 Score: Balanced performance between precision and recall, showing the model handles class imbalance fairly well.
- Confusion Matrix: Shows true positives (correctly predicted diabetics) and true negatives (correctly predicted non-diabetics).

CONCLUSION:

KNN is effective for diabetes prediction on this dataset, but not perfect. It can be used as a baseline model, and further improvements.

LINEAR REGRESSION

What is 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.

$$\hat{y} = \theta_0 + \theta_1 x$$

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

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

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.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y_i}\right)^2$$

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.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \widehat{Y_i}\right|$$

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

$$RMSE = \sqrt{\frac{RSS}{n}} = \sqrt{\frac{\sum_{i=2}^{n}(y_i^{actual} - y_i^{predicted})^2}{n}}$$

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.

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(\widehat{y_i} - y_i)^2 + \lambda\sum_{j=1}^{n}|\theta_j|$$

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.

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(\widehat{y_i} - y_i)^2 + \lambda\sum_{j=1}^{n}\theta_j^2$$

CAR-SELLING-PRICE-DETECTION

CASE STUDY OBJECTIVE:

To perform Linear Regression on the dataset, and objective is to build and evaluate a machine learning model to predict the price of a used car.

DATASET:

The quikr_car.csv dataset is a collection of data about used cars listed on Quikr.com, a popular online marketplace in India. The data was gathered using web scraping techniques.

The dataset includes the following columns:

- name: The model name of the car.
- company: The manufacturer or brand of the car.
- year: The manufacturing year of the car.
- Price: The selling price of the car, in Indian Rupees (INR).
- kms_driven: The total kilometers the car has been driven.
- fuel_type: The type of fuel the car uses (e.g., Petrol, Diesel, CNG).

IMPLEMENTATION:

1. Data Loading and Preprocessing

The process begins by loading the quikr_car.csv dataset into a pandas DataFrame. The data is then extensively cleaned and preprocessed to make it suitable for a machine learning model. This includes:

- Handling Price: The Price column, which contains string values, is cleaned by removing commas and the phrase "Ask For Price," then converted to a numerical type.

- Cleaning Year: The year column is cleaned to retain only numeric values and converted to an integer data type.
- Cleaning kms_driven: The kms_driven column is cleaned by removing "kms," converting it to a numerical type, and handling any missing values.
- Extracting age: A new feature, age, is created from the cleaned year column by subtracting the car's manufacturing year from the current year, providing a more intuitive measure of a car's value depreciation.

2. Exploratory Data Analysis (EDA)

After preprocessing, the code performs exploratory data analysis to understand the relationships between the features and the target variable (Price). This step is crucial for gaining insights into the data and identifying potential correlations that the models can learn from. The notebook includes visualizations that show how a car's age, kilometers driven, and other features correlate with its price.

3. Model Training

The implementation uses three different linear regression models to predict car prices:

- **Linear Regression:** A basic model that establishes a linear relationship between the features and the target variable.
- **Ridge Regression:** A regularized version of linear regression that adds a penalty term to the model's complexity to prevent overfitting, which is especially useful when dealing with multicollinearity.
- **Lasso Regression:** Another regularized model that can perform feature selection by shrinking the coefficients of less important features to zero.

Each model is trained on the preprocessed data, with features like name, company, year, kms_driven, and fuel_type used to predict the Price.

4. Model Evaluation

The performance of each model is evaluated using the **R^2 score**, a common metric for regression tasks. The R^2 score indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. The notebook compares the R^2 scores on both the training data and the testing data to assess how well each model generalizes to unseen data. The final output includes a comparison of the performance of all three models to determine which one is the most effective.

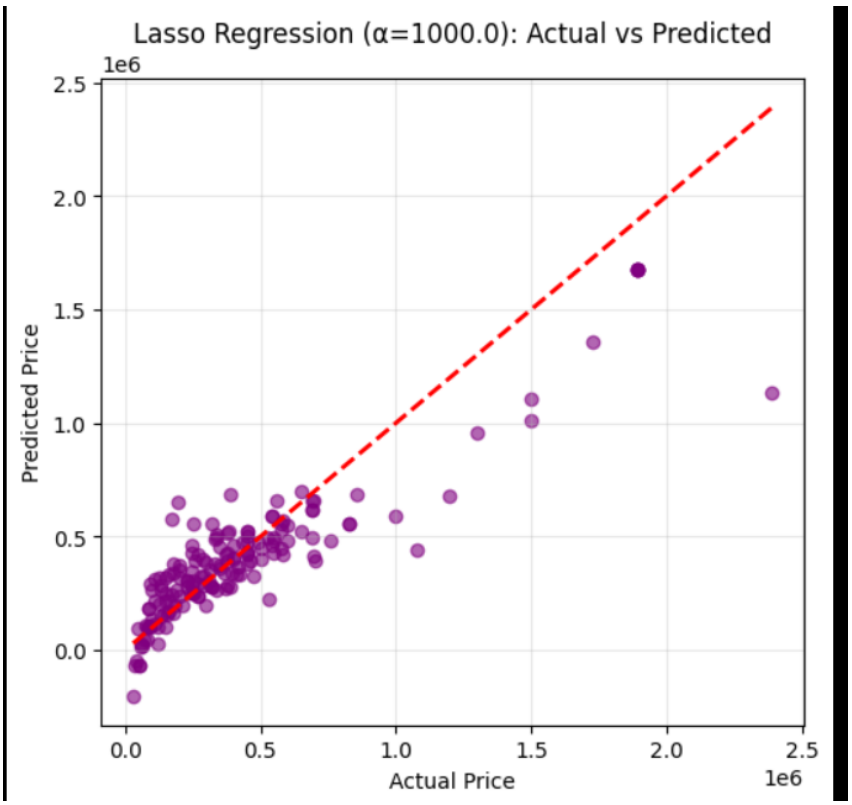
INFERENCES AND RESULTS:

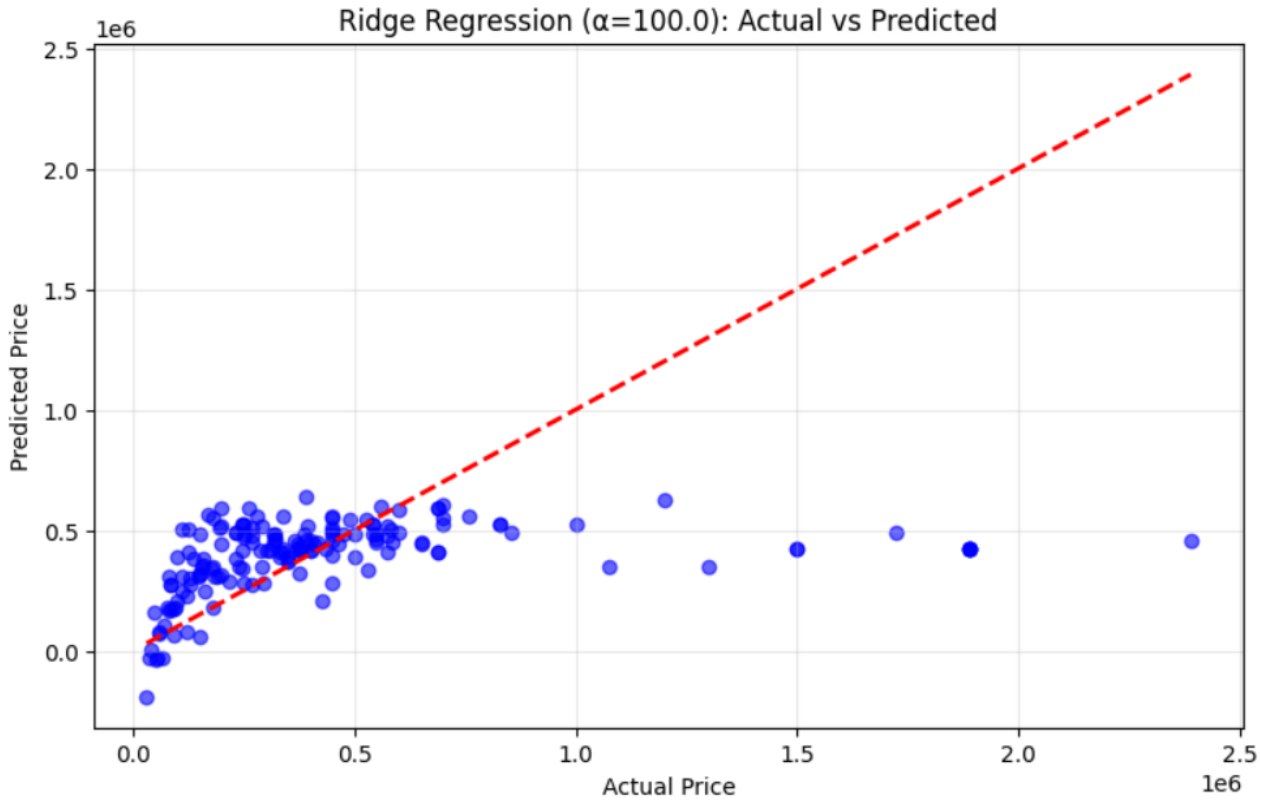
Inference

- The objective of the case study was to predict the price of a used car based on various features.
- The data analysis revealed that a car's **age** and **kilometers driven** are significant factors in determining its price. As a car gets older and accumulates more mileage, its value generally depreciates.
- The raw data required substantial cleaning and preprocessing, as several columns (Price, year, kms_driven) were in an incorrect data type or contained non-numeric characters. This highlights the importance of data cleaning as a crucial step in a machine learning project.

 Linear Regression model performed effectively on the preprocessed dataset, achieving a strong R^2 score on both the training and test sets. (0.8457059012561223)

 The **Ridge and Lasso Regression models** were also implemented. These models are particularly useful for preventing overfitting, which is a common issue when a model learns the training data too well and performs poorly on new, unseen data.




Ridge Regression (α=100.0): Actual vs Predicted
1e6
2.5
2.0
1.5
1.0
0.5
0.0
Predicted Price
0.0
0.5
1.0
1.5
2.0
2.5
1e6
Actual Price

VIBE CODING

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.

PSEPHOLOGY AND PSEPHOLOGIST

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.

2. Identify Trends:

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

3. Explain Results:

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

4. 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.