

# OASIS INFOBYTE INTERNSHIP (Data Science)

## Task 3: Car Price Prediction with Machine Learning

**Name:** Palak Parmar

**College:** Bansal Group Of Institute

**Submission Date:** 06/11/2025

### Introduction:

This beginner-level project demonstrates how to predict car selling prices using a simple Machine Learning model. We use a small dataset with common features (Year, Present Price, Kms Driven, Owner) and apply Linear Regression. The code is kept minimal and easy to run for beginners.

### Libraries to install (run once in terminal):

```
pip install pandas  
pip install scikit-learn  
pip install matplotlib
```

### Code (copy and run in Python):

```
# Beginner-friendly Car Price Prediction (no external files required)

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error

# Sample dataset created in code to avoid file path problems
data = {
    'Year': [2014, 2016, 2012, 2018, 2015, 2013, 2017, 2011, 2019, 2010],
    'Present_Price': [5.59, 9.54, 3.75, 11.25, 6.87, 4.45, 8.99, 2.30, 12.50, 1.99],
    'Kms_Driven': [27000, 52000, 43000, 15000, 34000, 50000, 22000, 61000, 12000, 80000],
    'Owner': [0, 1, 0, 0, 1, 0, 0, 2, 0, 1],
    'Selling_Price': [4.5, 7.8, 3.0, 9.5, 6.0, 3.8, 8.0, 2.2, 10.5, 1.5]
}

df = pd.DataFrame(data)

# Features and target
X = df[['Year', 'Present_Price', 'Kms_Driven', 'Owner']]
y = df['Selling_Price']

# Split into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train model
model = LinearRegression()
model.fit(X_train, y_train)
```

```
# Predict
y_pred = model.predict(X_test)

# Evaluate and print results
print("R2 Score:", round(r2_score(y_test, y_pred), 3))
print("Mean Absolute Error:", round(mean_absolute_error(y_test, y_pred), 3))
print("Test Actuals:", list(y_test.values))
print("Test Predictions:", [round(x,2) for x in y_pred])
```

Expected Output (example): - R2 Score: a value close to 1.0 indicates better fit (example: 0.85) - Mean Absolute Error: small number (example: 0.5) - Printed lists showing actual vs predicted selling prices for test samples. This code uses an internal sample dataset so it will run without external CSV files, ensuring the PDF is beginner-friendly and runs without file errors.

Conclusion: This simple example shows the full workflow: create data, select features, split, train a model, predict, and evaluate. Because the dataset is defined inside the code, beginners can copy-paste and run immediately without file path or encoding issues.