Project Report On

Used Car Price Prediction



Submitted in partial fulfillment for the award of

Post Graduate Diploma in Big Data Analytics (PG-DBDA) From Know-IT(Pune)

Guided by:

Anay Tamhankar

Trupti Joshi

Prasad Deshmukh

Submitted By:

Abhishek Patel (230943025001) Ajinkya Gaiki (230943025003) Nikhil Gaikwad (230943025017)

Vaibhav Gurav (230943025021)

**CERTIFICATE**

TO WHOMSOEVER IT MAY CONCERN

This is to certify that

Abhishek Patel (230943025001) Ajinkya Gaiki (230943025003) Nikhil Gaikwad (230943025017)

Vaibhav Gurav (230943025021)

Have successfully completed their project on

Used Car Price Prediction

Under the guidance of Anay Tamhankar, Trupti Joshi, and Prasad Deshmukh

## **ACKNOWLEDGEMENT**

Our project used car price prediction was a great learning experience for us and we are submitting this work to CDAC Know-IT (Pune).

We all are very glad to mention Anay Tamhankar, Trupti Joshi and Prasad Deshmukh for their valuable guidance while working on this project. Their guidance and support helped us overcome various obstacles during the project work.

We are highly grateful to Mr. Vaibhav Inamdar Manager (Know IT), CDAC, for his guidance and support whenever necessary while doing this course Post Graduate Diploma in Big Data

Analytics (PG-DBDA) through CDAC ACTS, Pune.

Our most heartfelt thanks go to Mr. Prasad Deshmukh (Course Coordinator, PG-DBDA) who gave all the required support and kind coordination to provide all the necessities like required hardware, internet facility, and extra Lab hours to complete the project and throughout the course up to the last day here in CDAC Know-IT, Pune.

**TABLE OF CONTENTS**

ABSTRACT

1. INTRODUCTION
2. DATA COLLECTION AND FEATURES
3. SYSTEM REQUIREMENTS
   1. SOFTWARE REQUIREMENTS
   2. HARDWARE REQUIREMENTS
4. FUNCTIONAL REQUIREMENTS
5. ARCHITECTURE
6. PR-EPROCESSING
7. EDA
8. MACHINE LEARNING ALGORITHMS
9. DATA VISUALIZATION AND REPRESENTATION
10. CONCLUSION AND FUTURE SCOPE
11. REFERENCES

### ABSTRACT

In the used car market, buyers and sellers often lack reliable methods to determine fair prices for

cars. According to a survey, approximately 50% of people prefer to buy used cars because new

ones can be really expensive, with high taxes adding to the cost. Therefore, there is a need for a

Price Prediction model that effectively determines the worthiness of the car using a variety of

dependent features. This project addresses the above challenge by developing a machine-learning

model that predicts car prices based on various features. Using the dataset's features and the

regression model, the price of the car will be predicted. Our model provides transparency

in pricing, allowing both sellers and buyers to understand the factors affecting the price.

Therefore, sellers receive a reasonable value for their cars and buyers make informed decisions.

### INTRODUCTION

### 

### 

### Determining the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle’s price on the market. The focus of this project is developing machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases. We implement and evaluate various learning methods on a dataset consisting of the sale prices of different cars and models. We will compare the performance of various machine learning algorithms like Linear Regression, Ridge Regression, Lasso Regression, Gradient Boosting Regressor, Decision Tree Regressor, Random Forest and choose the best out of it. Depending on various parameters we will determine the price of the car. Regression Algorithms are used because they provide us with continuous value as an output and not a categorized value because of which it will be possible to predict the actual price a car rather than the price range of a car. User Interface has also been developed which acquires input from any user and displays the Price of a car according to user’s inputs.

# Dataset Collection and Features

#### Data Sources:

#### The dataset used in this project was obtained from Kaggle, a platform for data

#### science and machine learning enthusiasts.

#### Data Structure:

#### The data structure of a vehicle’s dataset containing dependent features of cars typically follows a

#### tabular format, where each row represents a single car listing and each column represents a

#### specific attribute or feature of the car. The dependent features, also known as target variables or

#### labels, are typically those attributes that you want to predict or analyze based on other independent

#### features in the dataset.

#### Dataset Size: The dataset, sourced from Kaggle, contains about 450,000 records of used cars, offering insights into various attributes like car age, odometer reading, manufacturer, model, and more.

#### Features / Attribute:

**Odometer**: This typically represents the mileage or distance traveled by the vehicle.

**Car Age:** This column might represent the age of the car, usually calculated from the year of manufacture to the current year.

**Manufacturer:** The manufacturer is the company that built the vehicle.

**Model:** The model refers to the specific version or model of the vehicle produced by

the manufacturer.

**Condition:** This column describes the condition of the vehicle, which could include categories

such as "new," "used," "like new," "fair," etc.

**Cylinders:** The number of cylinders in the vehicle's engine. Common values include 4, 6, 8, etc.

**Fuel:** The type of fuel used by the vehicle, such as gasoline, diesel, hybrid, electric, etc.

**Title Status:** This column indicates the status of the vehicle's title, which could include "clean," "rebuilt," "salvage," etc.

**Transmission:** The type of transmission installed in the vehicle, such as automatic, manual,

or other variants.

**Drive:** This column describes the drive configuration of the vehicle, which could include

options like

**Type:** The type of vehicle, such as sedan, SUV, truck, hatchback, coupe, etc.

**Paint Color:** The color of the vehicle's exterior paint.

**Region:** This column might represent the geographic region where the vehicle is located

or being sold.

### SYSTEM REQUIREMENTS

**Hardware Requirements**

1. Computer: A computer with sufficient processing power and memory to run data processing and analysis tasks. A modern multicore processor and at least 8 GB of RAM are recommended.
2. Storage: Adequate storage space to store the generated dataset and any additional datasets if required. An SSD (Solid State Drive) is recommended for faster data access.
3. Internet Connection: A stable internet connection for downloading and installing software packages and libraries, as well as for any online resources needed during the project.

**Software Requirements**

1. Operating System: Windows 10 or higher
2. Python: The project heavily relies on Python for data generation, analysis, and machine learning. Ensure Python is installed on your system.
3. Python Libraries: Install the following Python libraries and dependencies using package managers like pip:

Pandas: For EDA

Matplotlib and Seaborn: data visualization

PySpark: For preprocessing and ML.

1. Apache Spark: For Preprocessing of dataset and ML.

1. Integrated Development Environment (IDE): Choose a Python friendly IDE, such as Jupyter Notebook, or your preferred text editor.

6.Visualization Software

Tableau: If you plan to visualize and analyze data with Tableau, install Tableau public.

### 

### FUNCTIONAL REQUIREMENTS

**1.Python:**

* Python is a general purpose and high-level programming language.
* It is use for developing desktop GUI applications, websites and web applications.
* Python allows to focus on core functionality of the application by taking care of common programming tasks.
* Python is derived from many other languages, including ABC, Modula3, C, C++, Algol68, Small Talk, and Unix shell and other scripting languages.

**2. Apache Spark:**

* What is Spark: Apache Spark is an opensource distributed computing system designed for processing large volumes of data.
* Key Features: Spark provides a number of key features that make it well-suited for processing big data, including in memory processing, support for various data sources and formats, fault tolerance, and scalability.
* Spark also provides a range of APIs, including SQL, streaming, machine learning, and graph processing, making it a versatile platform for a wide range of use cases.

**3.Tableau:**

* Data visualization is the graphical representation of information and data.
* It helps create interactive elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.
* It helps create interactive graphs and charts in the form of dashboards and worksheets to gain business insights.
* Tableau is widely used for Business Intelligence but is not limited to it.
* All of this is made possible with gestures as simple as drag and drop.

# ARCHITECTURE



Fig: System Architecture of Used car price prediction model.

**PRE - PROCESSING**

**1.Data Loading:** The dataset is loaded into a PySpark DataFrame from a CSV file. This DataFrame will be used for subsequent preprocessing steps.

**2.Column Removal:** Certain columns that are not relevant to the price prediction task or contain mostly null values are dropped from the DataFrame.

**3.Duplicate Removal:** Duplicate records in the DataFrame are removed to ensure data integrity and avoid bias in the analysis.

**4.Handling Missing Values:** Null values in specific columns (e.g., region, price, year, model, odometer, manufacturer, transmission, title status, fuel) are dropped. Categorical columns with missing values are filled with a placeholder value ('unknown').

**5.Categorical Encoding:** Top manufacturers, regions, and models are identified based on their frequency in the dataset. Less frequent manufacturer, region, and model values are grouped into an 'others' category to reduce the dimensionality of categorical features.

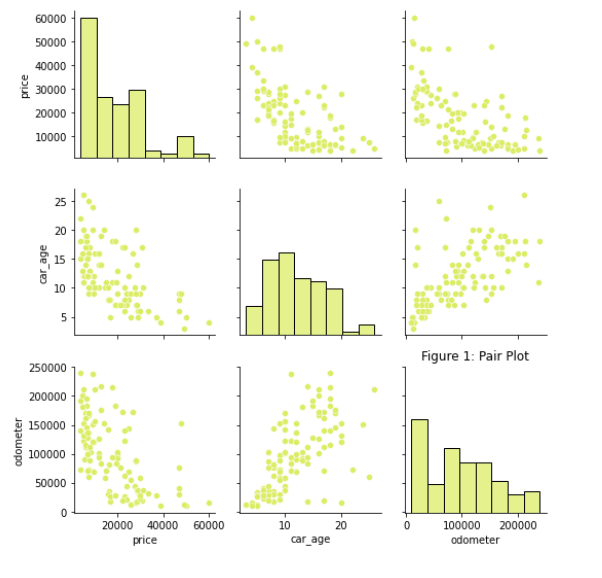
**6.Outlier Detection and Removal:** Outliers in the 'price' and 'odometer' columns are detected using the interquartile range (IQR) method. Records with 'price' and 'odometer' values outside the calculated upper and lower limits are filtered out.

**7.Year Filtering:** Records with 'year' values before 1996 and after 2022 are removed, likely to remove outliers or invalid data points.

**8.Feature Engineering:** A new feature 'car\_age' is created based on the 'year' column to represent the age of the car at the time of prediction.

**9.Final Cleanup:** Unnecessary columns, such as 'year', are dropped from the DataFrame, leaving only the relevant features for the prediction task

**EDA**

****

****

**MACHINE LEARNING ALGORITHMS**

**1.Linear regression:**

Linear regression is a simple yet powerful machine learning algorithm used to predict a continuous target variable based on one or more input features. PySpark, the Apache Spark library for Python, provides a distributed and scalable platform for big data processing.

Linear regression is useful for finding relationship between multiple continuous variables

There are multiple independent variables and single independent variable

**2.Lasso Regression:**

Lasso Regression The “LASSO” stands for Least Absolute Shrinkage and Selection Operator. Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

**3. Ridge Regression:**

A Ridge regressor is basically a regularized version of Linear Regressor.

The regularized term has the parameter ‘alpha’ which controls the regularization of the model i.e helps in reducing the variance of the estimates.

**4. Decision Trees:**

are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions.

Decision tree is an algorithm to solve classification problems. The decision tree algorithm uses a tree structure and uses layered inference to achieve the final classification. The decision tree is composed of the following elements:

1. Root node: contains the complete set of samples

2. Internal node: corresponding characteristic attribute test

3. Leaf node: represents the result of the decision

**5.Random Forest:**

Random Forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. Random Forest is suitable for situations when we have a large dataset, and interpretability is not a major concern. Decision trees are much easier to interpret and understand. Since a random forest combines multiple decision trees, it becomes more difficult to interpret.

1. **Gradient Boosting:**

Gradient Boosting is a machine learning technique that builds an ensemble of weak learners, typically decision trees, to create a strong predictor. The algorithm starts by fitting a simple model to the data, such as a decision tree with one or two levels. The residuals from this model are then used to train a second model, which is added to the ensemble. This process is repeated many times, with each new model trained on the residuals of the previous models. The final predictor is the sum of all the models in the ensemble.

**Hyper parameter tunning:**

We cannot directly feed categorical data into Machine Learning (ML) algorithms. We must provide a numerical representation to ML models of the categorical features of a dataset. While working in python, we generally use label encoder, get dummies, and OneHotEncoder for converting the categorical features. However, in PySpark, we can perform the conversion in different ways.

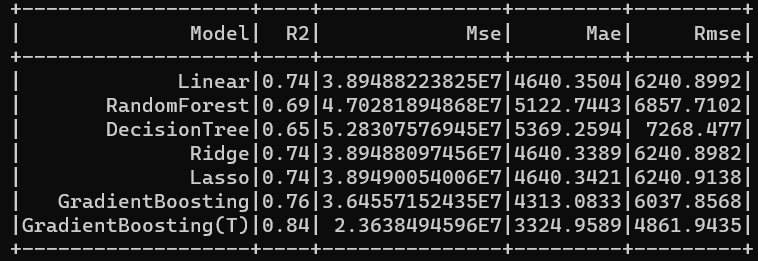
In Pyspark, there are two methods available that we can use for the conversion process: String Indexer and

OneHotEncoder.

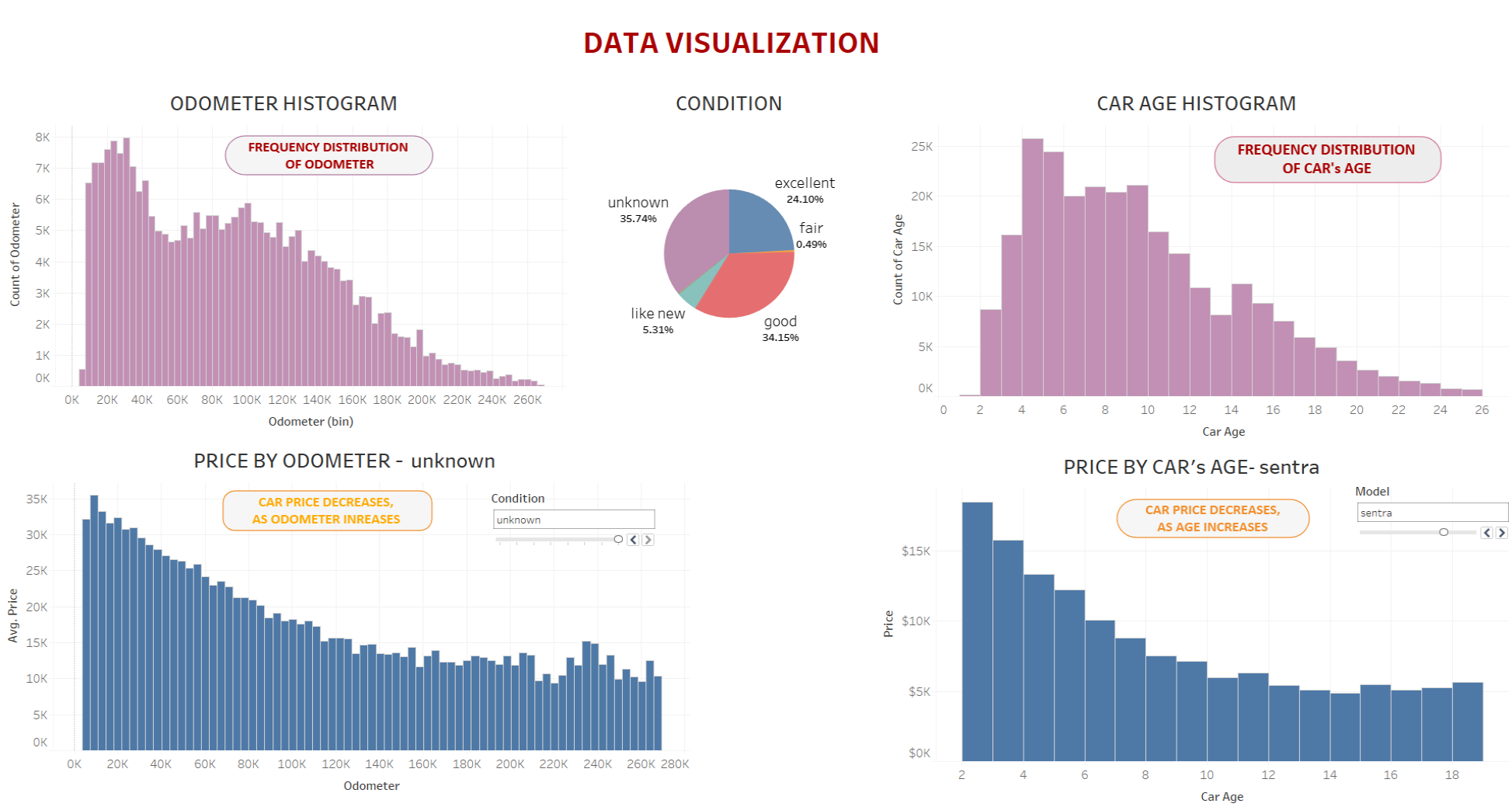
When a feature contains only two categories/ groups, in that case, we can directly apply the String Indexer method for conversion. The String Indexer method is equivalent to Label Encoder for the scikit learning package by python.

However, when a feature contains more than two groups, in that case, we cannot directly apply OneHotEncoder as we do in the python version. In pyspark, the OneHotEncoder requires the input into a numerical format, thus, before fitting the categorical data into the OneHotEncoder in pyspark, we must convert the feature into a numerical format using the available String Indexer in the pyspark.

**MODEL PERFORMANCE**

****

# TABLEAU DASHBOARD



**CONCLUSION**

**1.Data Understanding and Cleaning:** The dataset consisted of approximately 450,000 records of used cars. Initial data exploration revealed the presence of duplicate records, missing values, and outliers. Columns irrelevant to the price prediction task were removed, and missing values were handled appropriately.

**2.Feature Engineering:** Categorical features such as manufacturer, model, and region were processed to reduce dimensionality and improve model performance. New features, such as 'car\_age', were engineered to provide additional insights into the dataset.

**3.Outlier Detection and Removal:** Outliers in the 'price' and 'odometer' columns were identified using the interquartile range (IQR) method and filtered out to ensure model robustness.

**4.Modeling Approach:** The next steps in the project would involve selecting an appropriate machine learning algorithm for training the predictive model. Techniques such as regression analysis or ensemble methods like Gradient Boosted Trees (GBT) could be explored to predict

used car prices accurately.

**5.Evaluation and Deployment:** After training the model, it would be evaluated using suitable performance metrics such as mean absolute error (MAE) or

root mean square error (RMSE). The trained model could then be deployed into production, allowing users to input car features and receive predicted prices.

**6.Future Directions:** Further refinement of the model could involve hyperparameter tuning, feature selection, and cross-validation to improve predictive accuracy. Integration of additional data sources or advanced techniques like natural language processing (NLP) for analyzing car descriptions could

enhance model performance.

**FUTURE SCOPE**

**1.Incorporating More Features:** Integrate additional relevant features such as vehicle history reports, maintenance records, and ownership history to improve prediction accuracy.

**2.Geospatial Analysis:** Incorporate geospatial analysis to account for regional variations in used car prices, considering factors such as location-specific demand, supply dynamics, and economic conditions.

**3.Deep Learning Models:** Explore deep learning architectures such as recurrent neural networks (RNNs), long short-term memory networks (LSTMs), or attention mechanisms to capture sequential patterns and dependencies in the data.

**4.Deployment and Scalability:** Deploying the used car price prediction model on AWS opens up several opportunities for scalability, reliability, and accessibility.

**5.Marketplace Integration:** Integrate the prediction model into online used car marketplaces or platforms to provide real-time pricing insights and guidance to sellers and buyers, enhancing the overall user experien

**References**

1. Apache Spark. [https://spark.apache.org/]
2. Python. [[https://ww](http://www.python.org/)w.py[thon.](http://www.python.org/)org/]
3. scikit-learn. [https://scikit-learn.org/]