

**Vinitha Jeevarathnam**

May 30, 2022

**Module 6 – Practical Application 2**

**Text

Description automatically generated**

**Logo

Description automatically generated**

Table of Contents

[**1.** **CRISP-DM Framework** 3](#_Toc104837533)

[**2.** **Business Understanding** 4](#_Toc104837534)

[**3.** **Data Understanding** 5](#_Toc104837535)

[**4.** **Data Preparation** 7](#_Toc104837536)

[**5.** **Modeling** 9](#_Toc104837537)

[**6.** **Evaluation** 10](#_Toc104837538)

[**7.** **Deployment** 14](#_Toc104837539)

## **CRISP-DM Framework**

The CRoss-Industry Standard Process for Data Mining provides a model and framework for AI/ML data mining. For this practical application, I have adopted several steps from the CRISP-DM framework.

Diagram

Description automatically generated

## **Business Understanding**

##### Determine Business Objectives

The dataset provided for this practical application is from Kaggle that contains information on approx. 4.2 million used cars. I need to use this dataset to accomplish the following objectives,

1. Identify the factors that affect the price
2. Identify the factors valued by customers with a used car
3. Provide the recommendations to a used car dealership

##### Assess Situation

* **Assumptions**
  + The data provided for this application is of high quality and minimal effort is needed to cleanup
  + Availability of system resources to perform data processing of approx. 4.2 data records
* **Constraints**
  + None

##### Determine Data Mining Goals

* Maintain as much as data after data cleanup activities
* Leverage Target Encoder and Iterative Imputer like BayesianRidge to convert categorical columns to numerical columns and smartly fill ‘NaN’ values

##### Produce Project Plan

A project plan is not needed for this application.

## **Data Understanding**

##### Collect Initial Data

* Import all the necessary packages needed to perform functions related to pandas, modeling (sklearn), and plotting
* Import the data using read\_csv
* The dataset contains 426880 records and 18 columns

##### Describe and Explore Data

* 4 out of the 18 are numerical features and the remaining are categorical features
* Since **id and VIN** are not helpful with modeling, these features are dropped from the dataframe
* **price:** Min(0); Max(3736928711); Mean(75199.03); Median (13950.0); std(12182282.17)
* **manufacturer:** [nan, 'gmc', 'chevrolet', 'toyota', 'ford', 'jeep', 'nissan', 'ram', 'mazda', 'cadillac', 'honda', 'dodge', 'lexus', 'jaguar', 'buick', 'chrysler', 'volvo', 'audi', 'infiniti', 'lincoln', 'alfa-romeo', 'subaru', 'acura', 'hyundai', 'mercedes-benz', 'bmw', 'mitsubishi', 'volkswagen', 'porsche', 'kia', 'rover', 'ferrari', 'mini', 'pontiac', 'fiat', 'tesla', 'saturn', 'mercury', 'harley-davidson', 'datsun', 'aston-martin', 'land rover', 'morgan']
* **model:** [nan, 'sierra 1500 crew cab slt', 'silverado 1500', ..., 'gand wagoneer', '96 Suburban', 'Paige Glenbrook Touring']
* **condition:** [nan, 'good', 'excellent', 'fair', 'like new', 'new', 'salvage']
* **cylinders:** [nan, '8 cylinders', '6 cylinders', '4 cylinders', '5 cylinders', 'other', '3 cylinders', '10 cylinders', '12 cylinders']
* **fuel:** [nan, 'gas', 'other', 'diesel', 'hybrid', 'electric']
* **odometer:** Min(0); Max(10000000.0); Mean(98043.33); Median(85548.0); std(213881.50)
* **title\_status:** [nan, 'clean', 'rebuilt', 'lien', 'salvage', 'missing', 'parts only']
* **transmission:** [nan, 'other', 'automatic', 'manual']
* **drive:** [nan, 'rwd', '4wd', 'fwd']
* **size:** [nan, 'full-size', 'mid-size', 'compact', 'sub-compact']
* **type:** [nan, 'pickup', 'truck', 'other', 'coupe', 'SUV', 'hatchback', 'mini-van', 'sedan', 'offroad', 'bus', 'van', 'convertible', 'wagon']
* **paint\_color:** [nan, 'white', 'blue', 'red', 'black', 'silver', 'grey', 'brown', 'yellow', 'orange', 'green', 'custom', 'purple']
* **region:** Cities across United States of America
* **state:** 51 states in United States of America
* **year:** [nan, 1900 to 2022]

##### Verify Data Quality

* Almost all columns have missing values (NaN)

A picture containing text

Description automatically generated

* All numerical features have outliers

Table

Description automatically generated

## **Data Preparation**

##### Select and Clean Data

* Used **IQR** to remove outliers from **price, year, and odometer**
* For categorical values with minimum values, I have converted the non-numerical values to numerical ones,

|  |  |  |
| --- | --- | --- |
| **Features** | **New Values** | **Comments** |
| **cylinders** | 0, 3, 4, 5, 6, 8, 10, 12 | Replaced ‘cylinders’ with ‘’ |
| **transmission** | 0 – other and NaN  1 – Automatic  2 – Manual | 0 assigned to ‘other’ and NaN |
| **drive** | 0 – NaN  1 – 4wd and fwd  2 - rwd | 0 assigned to NaN |
| **fuel** | 'gas', 'other', 'diesel', 'hybrid', 'electric' | ‘other’ assigned to NaN |
| **type** | 'pickup', 'truck', 'other', 'coupe', 'SUV', 'hatchback', 'mini-van', 'sedan', 'offroad', 'bus', 'van', 'convertible', 'wagon' | ‘other’ assigned to NaN |
| **model** | 'sierra 1500 crew cab slt', 'silverado 1500', 'silverado 1500 crew', ..., '1500 z71', 'ATI', '96 Suburban' | ‘other’ assigned to NaN |
| **title\_status** | 'clean', 'rebuilt', 'lien', 'salvage', 'missing', 'parts only' | ‘other’ assigned to NaN |
| **paint\_color** | 'white', 'blue', 'red', 'black', 'silver', 'grey', nan, 'brown', 'yellow', 'orange', 'green', 'custom', 'purple' | ‘other’ assigned to ‘custom’ and NaN |
| **condition** | 'good', 'excellent', 'fair', 'like new', 'new', 'salvage' | A combination of ‘title\_status’ and ‘price’ is used to conditionally set the value for NaN |

##### Construct and Format Data

* Using numerical **Target Encoder** and **IterativeImputer-BayesianRidge**, all the categorical columns and values are converted to numerical features.

Graphical user interface, application

Description automatically generated

Table

Description automatically generated

* Before we initiate the modeling, a data quality check is performed the following are inferred,
  + Price, year, model, and odometer need to be scaled
  + An IQR needs to be performed again on ‘price’ to remove outliers (as mean and min are showing infinity)
* After scaling the dataframe using **StandardScalar()** method, we now have 349631 records and 16 numerical features

##### Integrate Data

This step is not needed as we are not integrating with other datasets

## **Modeling**

##### Select Modeling

* Supervised regression models are leveraged for this application
* Since we are predicting the price of used cars, we will keep the price as an output variable and everything else as input variables

##### Generate Test Design

* Using **train\_test\_split** method, the data is split into train set (70%) and test set (30%)
* Train set shape – (244741, 15)
* Test set shape – (104890,)

##### Build and Assess Model

The following supervised regression ML models are used in this application,

* Linear Regression
* Ridge Grid Regression
* Lasso Grid Regression
* KNN Regression

A summary of the modeling results is saved in a dataframe,

Graphical user interface, text, chat or text message

Description automatically generated

## **Evaluation**

##### Evaluate Results

* Both **Linear Regression and Ridge Regression** performed well with lower MSEs and RMSEs with a high R2 Score.
* **Year** of the car has the highest feature importance (coeff) in all the models and influences the price of the used car
* Apart from the modeling, the following is inferred from plotting the dataframe using seaborn and matplotlib,
  + The **price** of used cars has significantly gone up in 2021 and the inventory of **cars (count) is low** in the same year. Additionally, we can also classify the 2021 used cars are mostly in **‘new’ condition.**

Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

* + The **price** of used cars is influenced by the condition of the car. Cars with a **‘new’ condition** have high selling point than cars in ‘fair’ condition.
  + The **price** of used cars is influenced by the drive type of the car. Cars with **‘RWD’ (rear-wheel) drive** have a high selling price.

Chart, treemap chart

Description automatically generated

* + The **price** of used cars is influenced by the transmission type of the car. Cars with **‘automatic’ transmission** have a high selling price.

Chart, treemap chart

Description automatically generated

* + The **price** of used cars is influenced by the manufacturer of the car. **High-end car manufacturers** like **‘aston-martin’, ‘tesla’** have a high selling price for used cars. The **condition** of a specific manufacturer's car also influences the price of the used cars.

Chart, histogram

Description automatically generated

A picture containing text, writing implement, pencil, stationary

Description automatically generated

* + The **price** of used cars is influenced by the **fuel type** of the car. Cars utilizing **‘diesel’ and ‘electric’** fuel types have a high selling point for used cars

Chart, bar chart

Description automatically generated

* + The **price** of used cars is influenced by the **cylinders** of the car. Cars with **8 and 12 cylinders** have a high selling point for used cars.

Chart, bar chart, box and whisker chart

Description automatically generated

* + The **price** of used cars is influenced by the style type of car. Cars styles like a **truck, pickup, and offroad** have a high selling point for used cars.

Chart, bar chart

Description automatically generated

##### Determine Next Steps

* Explore additional modeling algorithms like decision trees, random forest, and bagging models to see how their MSE, RME, and R2s perform
* Instead of manually manipulating the values of categorical features, leverage **Target Encoder**, and **IterativeImputer-BayesianRidge** to automate the numerical value creation
* For now, I will provide the details that I have obtained in section 6.1 to the used car dealer to predict used car sales pricing.

## **Deployment**

This step is not needed for this practical application and can be skipped.