Kmit

SONET

Data Science

(Level-I)

Ajeet K. Jain, M. Narsimlu

(ML TEAM)- SONET, KMIT, Hyderabad

Session – 33

This session deals with

Case Study-2

Kmit SONET

Regression

DATA SCIENCE

Predicting price of pre-owned cars

Problem statement

Storm Motors is an e-commerce company who act as mediators between parties interested in selling and buying pre-owned cars.

For the year 2015-2016, they have recorded data about the seller and car including-

- ***Specification details***
- ***Condition of car***
- ***Seller details***
- ***Registration details***
- ***Web advertisement details***
- ***Make and model information***
- ***Price***

Storm Motors wishes to develop an algorithm to predict the price of the cars based on various attributes associated with the car.

Regression

Storm Motors wishes to develop an algorithm to predict the price of the cars based on associated with the car.

Framework

- Solution conceptualization (contd.)
  - Filter data based on logical checks
    - Price, year of registration, power
  - Reduced number of data
- Method identification
  - Linear regression
  - Random forest

Regression

Framework

- Realization of solution
  - Assumption checks using regression diagnostics
  - Evaluate performance metrics
  - If assumptions are satisfied and solutions are acceptable then model is good
  - If performance metrics are not reasonable then a single model is not able to capture the variation in price as a whole

Data Description

```
import pandas as pd
data_cars=pd.read_csv("cars_sampled.csv")
print(data_cars)
cars_data=data_cars.copy()
print(cars_data.info())
print(cars_data.describe())
```

Data Description

```
#formating float data
pd.set_option("display.float_format",lambda x:"%.3f"%x)
print(cars_data.describe())
pd.set_option("display.max_columns",500)
print(cars_data.describe())
```

Data Cleaning

```
#formating float data
pd.set_option("display.float_format",lambda x:"%.3f"%x)
print(cars_data.describe())
pd.set_option("display.max_columns",500)
print(cars_data.describe())
```

```
#Dropping unwanted columns
col=["name","dateCrawled","dateCreated","postalCode","lastSeen"]
cars_data=cars_data.drop(columns=col,axis=1)
print(cars_data)
```

```
#Removing duplicate records
cars_data.drop_duplicates(keep="first",inplace=True)
print(cars_data)
```

Data Cleaning

```
#Data Cleaning
print(cars_data.isnull().sum())
yearwise_count=cars_data["yearOfRegistration"].value_counts().sort_index()
print(yearwise_count)
print(sum(cars_data["yearOfRegistration"]>2018))
print(sum(cars_data["yearOfRegistration"]<1950))
sns.regplot(x="yearOfRegistration",y="price",scatter=True,fit_reg=False,
            data=cars_data)
#working range 1950-2018
```

Data Cleaning

```
#variable price
price_count=cars_data["price"].value_counts().sort_index()
#sns.distplot(cars_data["price"])
print(cars_data["price"].describe())
sns.boxplot(y=cars_data["price"])
print(sum(cars_data["price"]>150000))
print(sum(cars_data["price"]<100))
#working range 100-150000
```

Data Cleaning

```
#variable powerPS
power_count=cars_data["powerPS"].value_counts().sort_index()
sns.distplot(cars_data["powerPS"])
print(cars_data["powerPS"].describe())
sns.boxplot(y=cars_data["powerPS"])
print(sum(cars_data["powerPS"]>500))
print(sum(cars_data["powerPS"]<10))
#working range 10-500
```

Data Cleaning

```
#working range of Data


cars_data=cars_data[(cars_data.yearOfRegistration<=2018)
                    &(cars_data.yearOfRegistration>=1950)
                    &(cars_data.price>=100)
                    &(cars_data.price<=150000)
                    &(cars_data.powerPS>=10)
                    &(cars_data.powerPS<=500)]
print(cars_data)
```

Reducing variables

```
#combining yearOfRegistration and monthOfRegistration
cars_data["monthOfRegistration"]/=12
#creating new variable Age by adding yearOfRegistration
#and monthOfRegistration
cars_data["Age"]=(2018-cars_data["yearOfRegistration"])+cars_data["monthOfRegistra
cars_data["Age"]=round(cars_data["Age"],2)
print(cars_data["Age"].describe())
```

Reducing variables

```
#dropping yearOfRegistration and monthOfRegistration
cars_data.drop(columns=["yearOfRegistration","monthOfRegistration"],axis=1)
```

```
#Data visualization of Age,Price,powerPS
sns.distplot(cars_data["Age"])
sns.boxplot(cars_data["Age"])
sns.distplot(cars_data["price"])
sns.boxplot(cars_data["price"])
sns.distplot(cars_data["powerPS"])
sns.boxplot(cars_data["powerPS"])
```

Conclusion

You are aware of

Regression case study

We will proceed with

Case Study

Kmit

SONET

DATA SCIENCE

THANK YOU