# Analysis of Camera Dataset using Machine Learning Algorithms

TEAM DATA PIRATES
Pawan Prasad P, Phani Kumar Vedurumudi, G U Deepak, Jayanth O

PES UNIVERSITY EC CAMPUS
Hosur Rd, Konappana Agrahara, Electronic City, Bengaluru, Karnataka-560100

## ABSTRACT

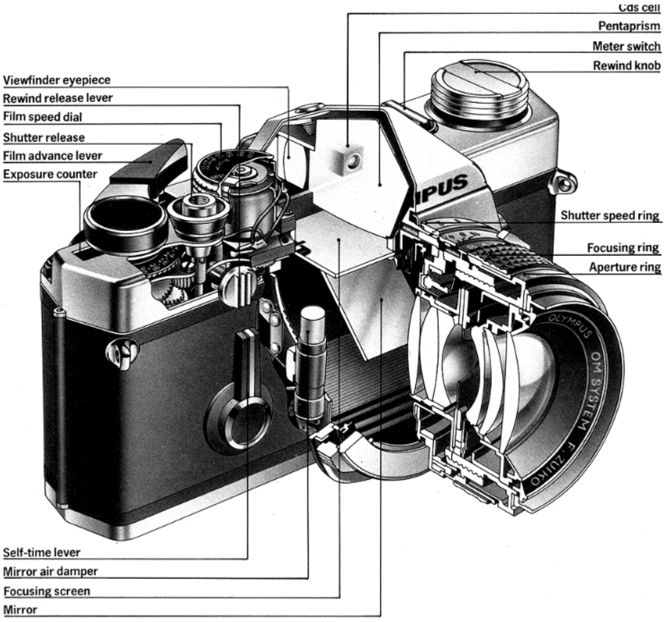
***Camera is used in every day to day life to capture the wonderful memories of our life. A camera is an optical instrument that captures a visual image. As we know that the evolution of camera from the ancient time to the recent days camera has a dramatical changes in the features like lens, aperture, focal length, price etc. We have taken the dataset to check the radical changes in the price over a period from 1997-2002. There are many varieties of cameras from olden days camera to Digital camera. Our main focus is on the price prediction of camera based on features of camera models using Machine Learning Algorithms like Linear Regression, Lasso-Ridge Regression, KNN and SVR Models. We basically used Regression Models because our dataset has continuous data. The features with more correlation are used for analysis. The overall Accuracy rate or prediction rate can reach upto 70% but generally it is between 50 – 70%, because the camera dataset doesn't have much correlation among the features.***

***Keywords—Camera, features, continuous data Period, price prediction, Machine Learning Algorithms, Linear Regression, Lasso-Ridge Regression, KNN, SVR, Accuracy rate.***

## I. INTRODUCTION

A camera is an optical instrument that captures a visual image. At a basic level, cameras are sealed boxes (camera body) with a small hole (aperture) that allows light through to capture a image on light-sensitive surface (usually photographic film or a digital sensor).

Our Dataset has different kinds of camera models with their features like Model, Release date, Max resolution, Low resolution, Effective pixels, Zoom wide (W), Zoom tele (T), Normal focus range, Macro focus range, Storage included, Weight (inc. batteries), Dimensions and Price. The Model feature is having categorical data and all other features are having continuous data. Due to **continuous data** we have used Regression Models to predict the price of the models. Our dataset has 13 features and 1038 observations.



The main focus of our project is to predict the price of camera based on features of camera models using Machine Learning Algorithms like Linear Regression, Lasso-Ridge Regression, KNN and SVR Models. We took problem related to price prediction because we have nowadays share market predictions which is a great problem in this tech world.

## II. PROBLEM STATEMENT

Our Project objective is to predict the price of the model based on the given features of the camera model. We have used the various Machine Learning Algorithms Models like KNN (k-nearest neighbours), Linear Regression and SVR.

## III. Related work

We have done the literature survey of nearly 12 recent journal Papers in which the authors have used various theory to use the camera in various AI fields like Computer Vision and AI in Security. But we have used different kinds of models to do the price prediction which is our main motive and we have chosen the price prediction which is a regression problem and due to more no. of types of models we cannot find the model type with all other features of camera which consists continuous data (value). Due to continuous data of all the features we have selected various ML Regression Models related to our dataset which is more precise and gives the correct prediction of price. The author's of many journal Papers have tried only one ML model for their project but we have used the best model out of various Regression models like SVR (Support Vector Regressor, KNN (k-nearest neighbours), Linear Regression Model and Lasso-Ridge Regression Model. The author of all this journal papers have used Neural Network, Human Tracking Algorithm, Camera Calibration Algorithm and many more. We have took all the useful techniques from the previous works. Our main motive is to find the prediction of prices of new data and our dataset have only the continuous value so we have took the various Regression Models to solve our problem. The scope of our problem is basically to give a clear cut prediction of the price of camera models and use the best model to deal with this problem.

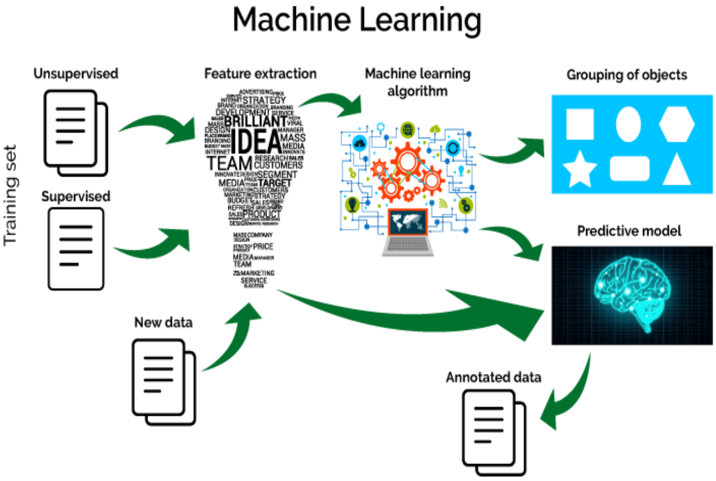
## IV. Model building and Testing

The Model Building Process is very crucial part and it is done in a systematic manner. We have used various Supervised Machine Learning Models for the price prediction and the tool which we have used for all the process of model building and testing is the **scikit learn** which is a very good tool for ML.

MACHINE LEARNING PROCESS
Clean, Prepare & Manipulate Data
Test Data
Get Data
3
Improve
2
4
1
Train Model
5

The Model is build using the following steps:

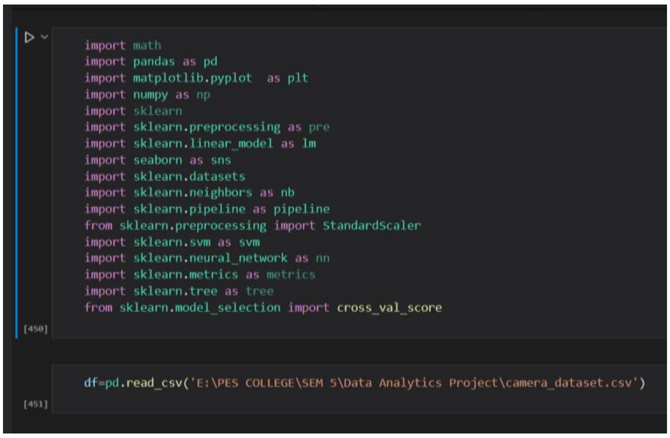
- ❖ Loading the dataset from the CSV File
- ❖ Exploratory Data Analysis (EDA)
- ❖ Preprocessing the Data
- ❖ Data Visualization
- ❖ Building Regression Model
- ❖ Cross Validation (Testing and Training (80-20 split) and Performance Metrics
- ❖ Model Evaluation
- ❖ Comparison Analysis



### ❖ Loading the dataset from the CSV File

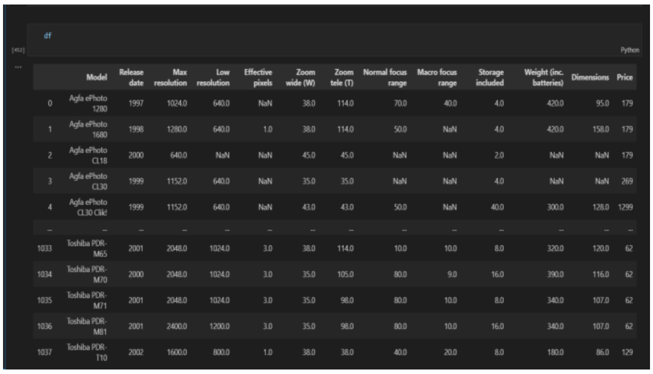
We have used the basic and advance libraries of the python programming language to build our model. Some of them are numpy, pandas, matplotlib, seaborn and sklearn etc.

Our Data has been collected from Kaggle's dataset which is online and free dataset. 1000 Cameras Dataset have been gathered and cleaned up by Petra Isenberg, Pierre Dragicevic and Yvonne Jansen.



```
import math
import pandas as pd
import matplotlib.pyplot  as plt
import numpy as np
import sklearn
import sklearn.preprocessing as pre
import sklearn.linear_model as lm
import seaborn as sns
import sklearn.datasets
import sklearn.neighbors as nb
import sklearn.pipeline as pipeline
from sklearn.preprocessing import StandardScaler
import sklearn.svm as svm
import sklearn.neural_network as nn
import sklearn.metrics as metrics
import sklearn.tree as tree
from sklearn.model_selection import cross_val_score
```

```
df=pd.read_csv('E:\PES COLLEGE\SEM 5\Data Analytics Project\camera_dataset.csv')
```



```
df
```

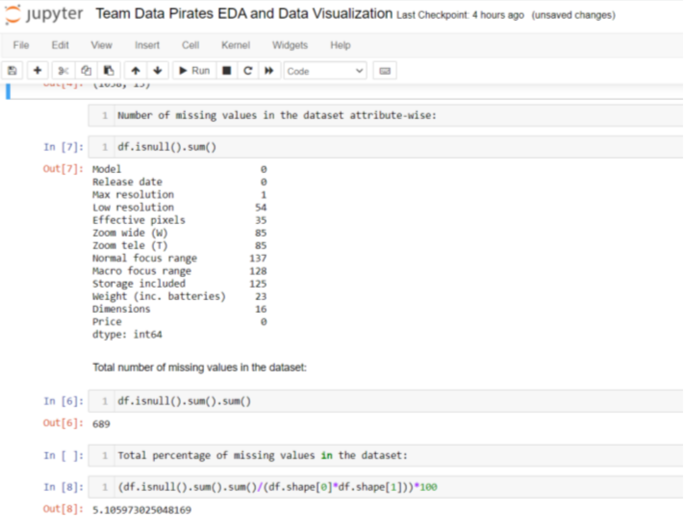
| | Model | Release date | Max resolution | Low resolution | Effective pixels | Zoom wide (W) | Zoom tele (T) | Normal focus range | Macro focus range | Storage included | Weight (inc. batteries) | Dimensions | Price |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Agfa ePhoto 1280 | 1997 | 1024.0 | 640.0 | NaN | 38.0 | 114.0 | 70.0 | 40.0 | 4.0 | 420.0 | 95.0 | 179 |
| 1 | Agfa ePhoto 1680 | 1998 | 1280.0 | 640.0 | 1.0 | 38.0 | 114.0 | 50.0 | NaN | 4.0 | 420.0 | 158.0 | 179 |
| 2 | Agfa ePhoto CL18 | 2000 | 640.0 | NaN | NaN | 45.0 | 45.0 | NaN | NaN | 2.0 | NaN | NaN | 179 |
| 3 | Agfa ePhoto CL30 | 1999 | 1152.0 | 640.0 | NaN | 35.0 | 35.0 | NaN | NaN | 4.0 | NaN | NaN | 269 |
| 4 | Agfa ePhoto CL30 Clik! | 1999 | 1152.0 | 640.0 | NaN | 43.0 | 43.0 | 50.0 | NaN | 40.0 | 300.0 | 128.0 | 1299 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1033 | Toshiba PDR-M65 | 2001 | 2048.0 | 1024.0 | 3.0 | 38.0 | 114.0 | 10.0 | 10.0 | 8.0 | 320.0 | 120.0 | 62 |
| 1034 | Toshiba PDR-M70 | 2000 | 2048.0 | 1024.0 | 3.0 | 35.0 | 105.0 | 80.0 | 9.0 | 16.0 | 390.0 | 116.0 | 62 |
| 1035 | Toshiba PDR-M71 | 2001 | 2048.0 | 1024.0 | 3.0 | 35.0 | 98.0 | 80.0 | 10.0 | 8.0 | 340.0 | 107.0 | 62 |
| 1036 | Toshiba PDR-M81 | 2001 | 2400.0 | 1200.0 | 3.0 | 35.0 | 98.0 | 80.0 | 10.0 | 16.0 | 340.0 | 107.0 | 62 |
| 1037 | Toshiba PDR-T10 | 2002 | 1600.0 | 800.0 | 1.0 | 38.0 | 38.0 | 40.0 | 20.0 | 8.0 | 180.0 | 86.0 | 129 |

## ❖ Exploratory Data Analysis (EDA)

This Dataset contains all the details of different Camera Models features and their values which helps to analysis various real-world problems. It has both Categorical and continuous data.

1. Dataset name: Camera Dataset
2. Source: Kaggle
3. No. of Observations: 1038
4. No. of Columns: 13
5. Percentage of Missing Values: 5%

The no. of Missing Values in the Dataset.



jupyter
Team Data Pirates EDA and Data Visualization Last Checkpoint: 4 hours ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

```
Number of missing values in the dataset attribute-wise:
```

```
In [7]: df.isnull().sum()
```

```
Out[7]: Model                        0
        Release date                 0
        Max resolution               1
        Low resolution              54
        Effective pixels            35
        Zoom wide (W)               85
        Zoom tele (T)               85
        Normal focus range         137
        Macro focus range          128
        Storage included           125
        Weight (inc. batteries)     23
        Dimensions                  16
        Price                        0
        dtype: int64
```

Total number of missing values in the dataset:

```
In [6]: df.isnull().sum().sum()
Out[6]: 689
```

```
In [ ]: Total percentage of missing values in the dataset:
```

```
In [8]: (df.isnull().sum().sum()/(df.shape[0]*df.shape[1]))*100
Out[8]: 5.105973025048169
```

This Dataset contains a combination of numerical and categorical features:

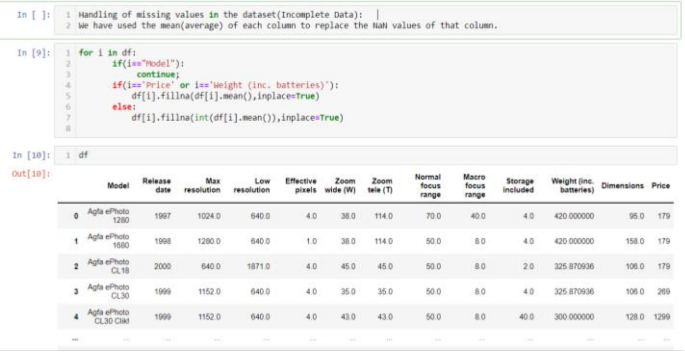
Categorical:

- Model

Numerical:

- Release date
- Max resolution
- Low resolution
- Effective pixels
- Zoom wide (W)
- Zoom tele (T)
- Normal focus range
- Macro focus range
- Storage included
- Weight (inc. batteries)
- Dimensions
- Price

Removing the Missing or Incomplete values in the dataset.



```
In [ ]: Handling of missing values in the dataset(Incomplete Data):
        We have used the mean(average) of each column to replace the NaN values of that column.
```

```
In [9]: for i in df:
                if(i=="Model"):
                    continue;
                if(i=='Price' or i=='Weight (inc. batteries)'):
                    df[i].fillna(df[i].mean(),inplace=True)
                else:
                    df[i].fillna(int(df[i].mean()),inplace=True)
```

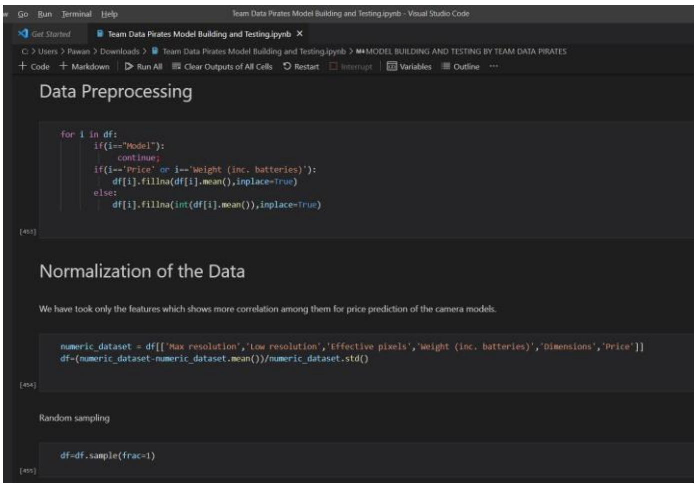
```
In [10]: df
```

Out[10]:

| | Model | Release date | Max resolution | Low resolution | Effective pixels | Zoom wide (W) | Zoom tele (T) | Normal focus range | Macro focus range | Storage included | Weight (inc. batteries) | Dimensions | Price |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Agfa ePhoto 1280 | 1997 | 1024.0 | 640.0 | 4.0 | 38.0 | 114.0 | 70.0 | 40.0 | 4.0 | 420.000000 | 95.0 | 179 |
| 1 | Agfa ePhoto 1680 | 1998 | 1280.0 | 640.0 | 1.0 | 38.0 | 114.0 | 50.0 | 8.0 | 4.0 | 420.000000 | 158.0 | 179 |
| 2 | Agfa ePhoto CL18 | 2000 | 640.0 | 1871.0 | 4.0 | 45.0 | 45.0 | 50.0 | 8.0 | 2.0 | 325.870936 | 106.0 | 179 |
| 3 | Agfa ePhoto CL30 | 1999 | 1152.0 | 640.0 | 4.0 | 35.0 | 35.0 | 50.0 | 8.0 | 4.0 | 325.870936 | 106.0 | 269 |
| 4 | Agfa ePhoto CL30 Clikl | 1999 | 1152.0 | 640.0 | 4.0 | 43.0 | 43.0 | 50.0 | 8.0 | 40.0 | 300.000000 | 128.0 | 1299 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

## ❖ Preprocessing the Data

The data of the dataset shows less correlation among the attributes. So we have selected only those attributes which has more correlation for model building and we have done random sampling for changing the data randomly for training and testing. We have used Normalization and Standardization which done in the Pipeline of sklearn.



Go
Run
Terminal
Help
Team Data Pirates Model Building and Testing.ipynb - Visual Studio Code
Get Started
Team Data Pirates Model Building and Testing.ipynb
C: > Users > Pawan > Downloads > Team Data Pirates Model Building and Testing.ipynb > M↓MODEL BUILDING AND TESTING BY TEAM DATA PIRATES
+ Code
+ Markdown
Run All
Clear Outputs of All Cells
Restart
Interrupt
Variables
Outline

Data Preprocessing

```
for i in df:
        if(i=="Model"):
            continue;
        if(i=='Price' or i=='Weight (inc. batteries)'):
            df[i].fillna(df[i].mean(),inplace=True)
        else:
            df[i].fillna(int(df[i].mean()),inplace=True)
```

[453]

Normalization of the Data

We have took only the features which shows more correlation among them for price prediction of the camera models.

```
numeric_dataset = df[['Max resolution','Low resolution','Effective pixels','Weight (inc. batteries)','Dimensions','Price']]
df=(numeric_dataset-numeric_dataset.mean())/numeric_dataset.std()
```

[454]

Random sampling

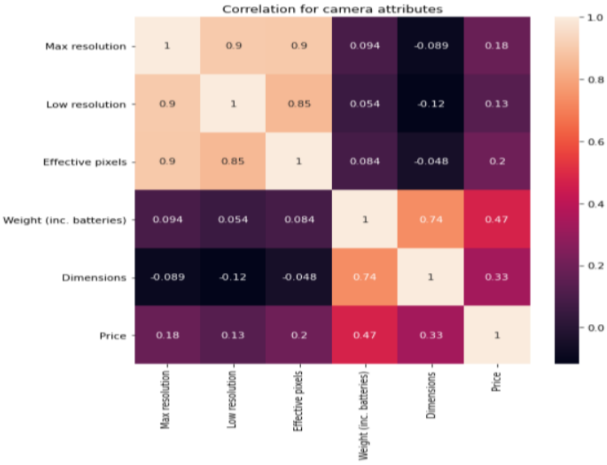
```
df=df.sample(frac=1)
```

[455]

## ❖ Data Visualization

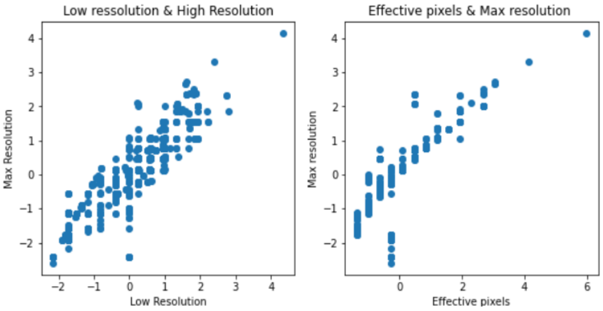
### Heat map:

The Heat map shows the correlation between the features. These features are used for the model building and testing.



### Scatter Plot:

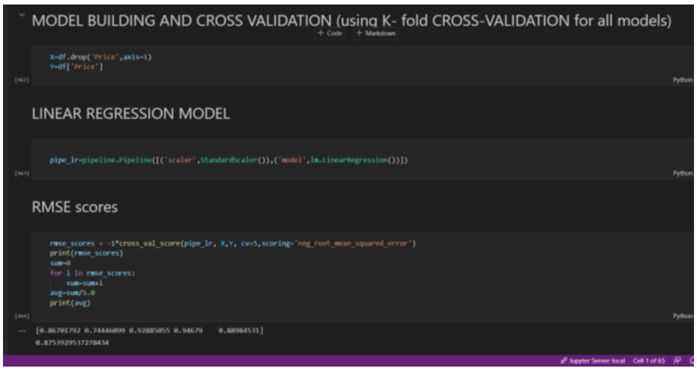
This plot shows the distribution of the data in the dataset. The variation or distance from each point to another point.



## ❖ Building Regression Model and Cross Validation and Evaluation

We have built many models and I would like to address only two models among them that is the Linear Regression Model and KNN Model. We have did the 5-fold Cross Validation which is like 80-20 split of data. The Evaluation of Model is done using various Performance Metrics like RMSE, MAE and Rsquare. The Model is built using sklearn by creating a Pipeline which is a model and which would preprocess the data using Standard scalar. We have used Performance Metrics also to evaluate the Model. Please check out our Github link below.

### Linear Regression Model and Cross validation:



MODEL BUILDING AND CROSS VALIDATION (using K- fold CROSS-VALIDATION for all models)

```
X=df.drop('Price',axis=1)
Y=df['Price']
```

LINEAR REGRESSION MODEL

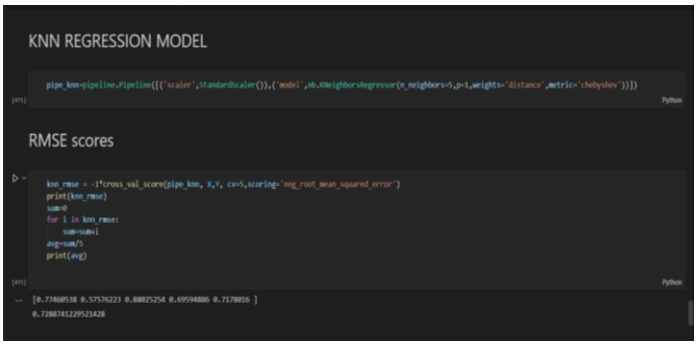
```
pipe_lr=pipeline.Pipeline([('scaler',StandardScaler()),('model',lm.LinearRegression())])
```

RMSE scores

```
rmse_scores = -1*cross_val_score(pipe_lr, X,Y, cv=5,scoring='neg_root_mean_squared_error')
print(rmse_scores)
sum=0
for i in rmse_scores:
    sum=sum+i
avg=sum/5.0
print(avg)
```

```
[0.86701792 0.74446099 0.92885055 0.94679    0.88984531]
0.8753929537278434
```

### KNN Model and Cross Validation:



KNN REGRESSION MODEL

```
pipe_knn=pipeline.Pipeline([('scaler',StandardScaler()),('model',nb.KNeighborsRegressor(n_neighbors=5,p=1,weights='distance',metric='chebyshev'))])
```

RMSE scores

```
knn_rmse = -1*cross_val_score(pipe_knn, X,Y, cv=5,scoring='neg_root_mean_squared_error')
print(knn_rmse)
sum=0
for i in knn_rmse:
    sum=sum+i
avg=sum/5
print(avg)
```

```
[0.77460538 0.57576223 0.88025254 0.69594886 0.7178016 ]
0.7288741229521428
```

## ❖ Comparison Analysis or Result

From the RMSE, MSE and Rsquare scores of the above all the Regression models we can make a inference that KNN model is better than all other models.

RMSE of Linear Regression - 0.8756

R^2 of Lasso and Ridge Regression - 0.1269

RMSE of KNN - 0.7293

RMSE of SVR - 0.9456

But we can clearly see that RMSE value of KNN Model is less compare to other models. In this Model evaluation we have consider the normalized data and we came to conclusion that KNN is the best model for prediction.

## V. Conclusion

The Dataset which we have used consist of less correlation among the features so our accuracy is around 50 to 70% which is given by KNN Model. We have not split the data explicitly to test the model instead we have used the 5-fold cross validation test which automatically splits the data in 80-20 split and based various Performance Metrics scores we have came to an conclusion that KNN Model is best for our dataset. We have also performed the tuning of parameters of the models by eliminating the less correlated features (attributes) from the dataset and then trained the model. The other Regression models are not so good for our dataset and we can conclude that KNN Model is best for our problem that is Price Prediction of the camera model.

## VI. References

[1] https://scikit-learn.org/stable/
[2] https://scikit-learn.org/stable/modules/cross_validation.html
[3] https://www.jstor.org/stable/25750706
[4] https://ieeexplore.ieee.org/abstract/document/709612/
[5] https://www.sciencedirect.com/science/article/pii/S0379073820304631

**Entire Project Details:**

- https://github.com/Pawantech-App/DATA-ANALYTICS-PROJECT-
- https://drive.google.com/drive/folders/1kb9kgaIb6HjDvNBAdP683aatBqJ9bX1M?usp=sharing

**CONTRIBUTION:**

Team Members:

- Pawan Prasad P - PES2UG19CS280
- Phani KumarVedurumudi - PES2UG19CS281
- G U Deepak - PES2UG19CS124
- Jayanth O - PES2UG19CS163