# Machine Learning Powered Webapp for Prediction of Used Cars Prices

VALASA TARUN[1], PALLIKONDA RITHIN[2], S CHANDRA SEKHAR[3] AND BEJJAM VASUNDHARA DEVI[4]

[1, 2] *Undergraduate Student, Department of Computer Science and Engineering, Sreenidhi Institute of Science and Technology, Hyderabad*

[3, 4] *Assistant Professor, Department of Computer Science and Engineering, Sreenidhi Institute of Science and Technology, Hyderabad*

***Abstract— Due to the pandemic situation the the rate of inflation is on surge and people from all over the world are experiencing this phenomenon, where the value of money we presently have is decreasing without spending them correctly. In this situation people are showing interest in buying or getting hands on whatever product they get for a cheaper price. Due to this effect used cars or pre-owned cars are on high demand because they cost less price and the depreciation of used cars is slow compared to a newly bought car which is way pricier and need more cost to maintain. In this paper we are going to use machine learning techniques to predict the price of used or pre-owned cars so as to help the people to sell their car for a fair price and avoid hassle of going to a physical showroom or a office of an organization to check their price of the car.***

***Indexed Terms— Pre-owned, price prediction, Machine Learning, Nodejs, Mongodb, javascript.***

## I. INTRODUCTION

Due to the pandemic the way we buy things has took a 180 degree turn so , as we cannot step out our houses , but we need our things to live our life smoothly and joyfully. As humans we need to keep our flow going on . Many people who have never bought something online have shopped on amazon or other e commerce websites for the first time in their life and the experience was also good according to the news statistics of many e commerce companies as they stated. So we thought of finding a new way of buying and selling vehicles from the home itself. This brings us to develop our project Using the latest technology stack which is popular and has good community support when we get stuck or any error occurs we can find the some pre-solved solutions on the internet without having any problems. Our project is to create a marketplace where people can see used cars and able to buy them from the website itself and contact the owner of the listed car through our web app and there is also an admin user for to look like example app statistics and monitoring the usage of certain user and has a doughnut shaped chart which portray the statistics. In this project we have also added a interesting feature which is needed in this time that is machine learning to predict the cost of the used car if some body wants to sell it on our web app. This machine learning model has been developed from the data set extracted from Kaggle and it is cleaned to remove waste i.e. not useful data rows and generated into another cleaned data set for training the model . As the provided data is with labels we used supervised machine learning and specifically chose the Linear Regression which gave us like close to ninety two percent(92%) accuracy .

## II. LITERATURE SURVEY

Our data set which we have selected is collected from web scraping the quickr cars website using python script using the beautiful soup library. The dataset has Around 20,000 rows and it has 6 major attributes or columns to consider for out usecase.

Car price prediction is a interesting field or subject which has a large scope because give rise to many other applications such as prediction of house prices or used bike prices for future iterations.

[1] In this study the findings were that the usage or buying of cars has shot up by 2.7 % from the year 2013 and most likely that number is going to increase over the coming decades and also more people want their own mode of transport rather than public transport.

In this paper the author [2] tells us about the model built using the SVM algorithm has performed well and can be used for solving this problem buit also found out that SVM cannot give good predictions for data consisting more than one dimension of freedom.

In [3] gongie built a model using the popular ANN (Artificial neuron networks) which have advanced features like back propagation and adjusting weights to make more accurate predictions. But here too the ANN cannot give good accuracy because it is best at solving or understanding non linear data than continuous or liner data in this case.

Further , In [4] the author applied all the regression and classification algorithms and conducted a comparative study and pick the best one to get the best accuracy , he found out that naïve bayes and decision tree algorithms cannot even get close to 70% due to lack of more data in the dataset.

## III. MATERIALS AND METHODOLOGY

### *A. Data extraction*

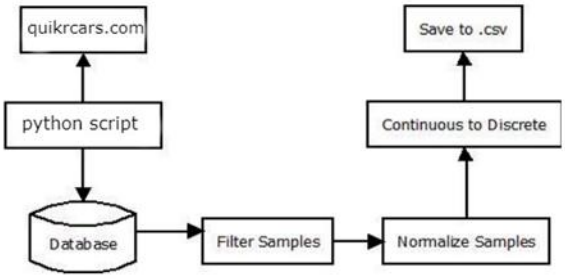


Fig 1 Data Extraction Diagram

### *B. Looking at extracted data*

| | name | company | year | Price | kms_driven | fuel_type |
|---|---|---|---|---|---|---|
| 0 | Hyundai Santro Xing XO eRLX Euro III | Hyundai | 2007 | 80,000 | 45,000 kms | Petrol |
| 1 | Mahindra Jeep CL550 MDI | Mahindra | 2006 | 4,25,000 | 40 kms | Diesel |
| 2 | Maruti Suzuki Alto 800 Vxi | Maruti | 2018 | Ask For Price | 22,000 kms | Petrol |
| 3 | Hyundai Grand i10 Magna 1.2 Kappa VTVT | Hyundai | 2014 | 3,25,000 | 28,000 kms | Petrol |
| 4 | Ford EcoSport Titanium 1.5L TDCi | Ford | 2014 | 5,75,000 | 36,000 kms | Diesel |

Fig 2 Extracted Data

### *C. Cleaning the extracted Data*

**Cleaned Data**

```
car
```

| | name | company | year | Price | kms_driven | fuel_type |
|---|---|---|---|---|---|---|
| 0 | Hyundai Santro Xing | Hyundai | 2007 | 80000 | 45000 | Petrol |
| 1 | Mahindra Jeep CL550 | Mahindra | 2006 | 425000 | 40 | Diesel |
| 2 | Hyundai Grand i10 | Hyundai | 2014 | 325000 | 28000 | Petrol |
| 3 | Ford EcoSport Titanium | Ford | 2014 | 575000 | 36000 | Diesel |
| 4 | Ford Figo | Ford | 2012 | 175000 | 41000 | Diesel |
| ... | ... | ... | ... | ... | ... | ... |
| 811 | Maruti Suzuki Ritz | Maruti | 2011 | 270000 | 50000 | Petrol |
| 812 | Tata Indica V2 | Tata | 2009 | 110000 | 30000 | Diesel |
| 813 | Toyota Corolla Altis | Toyota | 2009 | 300000 | 132000 | Petrol |
| 814 | Tata Zest XM | Tata | 2018 | 260000 | 27000 | Diesel |
| 815 | Mahindra Quanto C8 | Mahindra | 2013 | 390000 | 40000 | Diesel |

816 rows × 6 columns

Fig 3 Cleaned Data

Removed the data with nan values or null values which will effect in training of the model to get good results in the end.

### *D. Exploratory Data Analysis*

**Checking relationship of Fuel Type with Price**

```
plt.subplots(figsize=(14,7))
sns.boxplot(x='fuel_type',y='Price',data=car)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d353660d60>
```

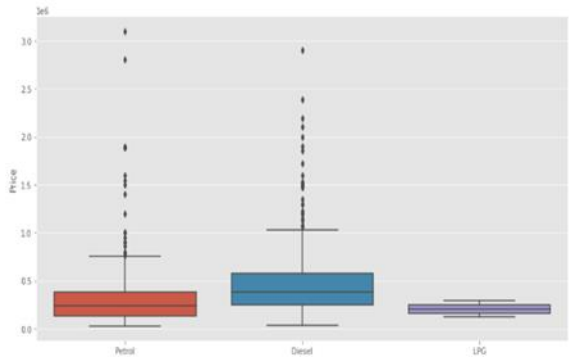


Fig 4 Fuel vs price graph


Fig 5 kilometers driven vs price graph

## IV. MODEL IMPLEMENTATION AND RESULTS

### A. *Model selection*

We have considered three techniques or algorithm for this dataset they are ANN (Artificial neural networks) , Random Forest and Linear Regression to cover all the aspects in machine learning that are deep learning , classification and regression as this a supervised learning problem as the data is labelled.

By splitting the data into 30 , 70 train test split we trained our model on these algorithms and got the following results.

| Algorithm | **Accuracy** | **Error** |
|---|---|---|
| RF | 41.18% | 8.04% |
| ANN | 42.35% | 7.05% |
| LR | 92.00% | 8.01% |

Table 1 Showing accuracies of the model

```
scores=[]
for i in range(1000):
    X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=i)
    lr=LinearRegression()
    pipe=make_pipeline(column_trans,lr)
    pipe.fit(X_train,y_train)
    y_pred=pipe.predict(X_test)
    scores.append(r2_score(y_test,y_pred))
```

Figure 6 Showing LR model is trained in random state for over 1000 times

```
[64]: scores[np.argmax(scores)]

t[64]: 0.920088412025344
```

Figure 7 Showing LR model's accuracy of 92%

### B. *Building web application*

Here we used HTML, CSS, javascript ,Nodejs, Mongodb, python , ajax , ejs languages / technologies to build our webapp

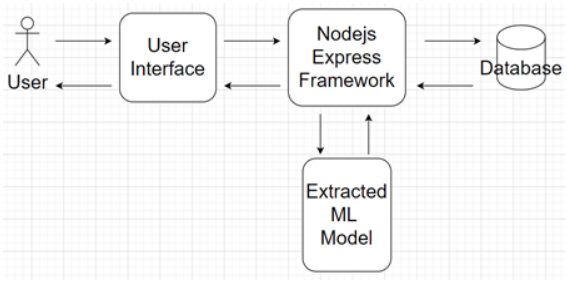


Figure 8 System Architecture of the app

In the above figure we can see there is the user who sees the User Interface with which he or she can interact and give inputs. After giving the inputs the input values goes to the nodejs server and get processed and then stored in the database if needed or directly sent to the extracted machine learning model. The extraction of the machine learning model is possible using a python library called pickle with which we can extract our models and use them else where. The input values are given to the model and the model gives a prediction of the car price as the output to the user which is rendered or shown to the user using the User Interface which is processed by the browser .

The communication between nodejs and our machine learning model is possible by using child process module in nodejs and specifically a function in the child process called spawn which can run any language file and give and receive inputs and outputs respectively.

Using the spawn function we give input and get output from our machine learning model which Linear Regression as shown above.

```
In [65]: pipe.predict(pd.DataFrame(columns=X_test.columns,
         data=np.array(
         ['Maruti Suzuki Swift','Maruti',2019,100,'Petrol']
         ).reshape(1,5)))
Out[65]: array([400707.28215338])
```

Figure 9 Showing the prediction made by the ML model with value of 400707.28215338 rupees from given inputs

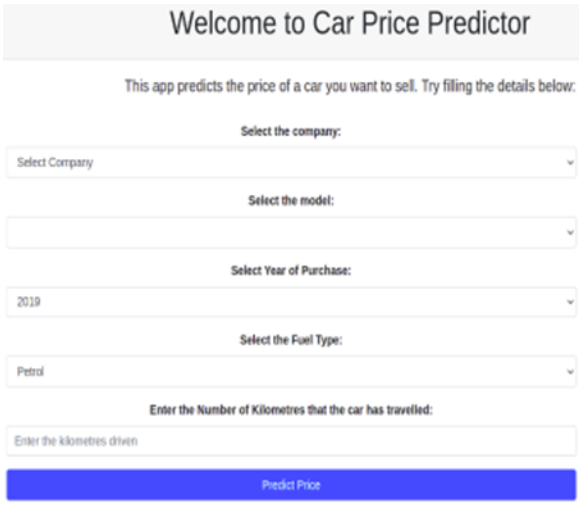


Figure 10 Showing the UI of the web application with some input fields as dropdowns and text boxes above

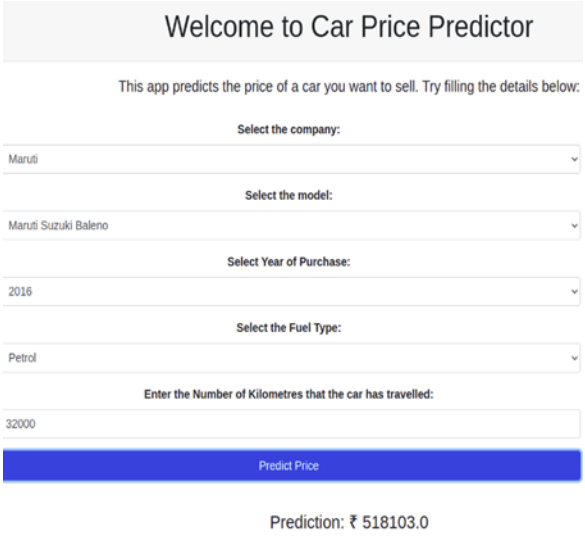


Figure 11 Showing the prediction made by the ML model at the bottom of the page using ajax , after giving the inputs to the UI form and submitting it. (above picture)

## V. CONCLUSION

Our machine learning model we developed using Linear Regression algorithm gave us 92 percent accuracy. We extracted the ML model using the well known pickle library and using child process in nodejs with spawn function to run the extracted ML model and show the prediction using ajax calls and displaying the output in the UI.

This model's accuracy can be further more increased by adding more data to the data set and retraining the ML model and also trying with other machine learning algorithms with hyper parameter tuning.

## REFERENCES

[1] Agencija za statistiku BiH. (n.d.), retrieved from: http://www.bhas.ba . [accessed July 18, 2018.].

[2] Listiani, M. (2009). Support vector regression analysis for price prediction in a car leasing application (Doctoral dissertation, Master thesis, TU Hamburg-Harburg).

[3] Wu, J. D., Hsu, C. C., & Chen, H. C. (2009). An expert system of price forecasting for used cars using adaptive neuro-fuzzy inference. Expert Systems with Applications, 36(4), 7809-7817..

[4] Gongqi, S., Yansong, W., & Qiang, Z. (2011, January). New Model for Residual Value Prediction of the Used Car Based on BP Neural Network and Nonlinear Curve Fit. In Measuring Technology and Mechatronics Automation (ICMTMA), 2011 Third International Conference on (Vol. 2, pp. 682-685). IEEE

[5] Ho, T. K. (1995, August). Random decision forests. In Document analysis and recognition, 1995., proceedings of the third international conference on (Vol. 1, pp. 278-282). IEEE.

[6] Feng Wang; Xusong Zhang; Qiang Wang "Prediction of Used Car Price Based on Supervised Learning Algorithm", International Conference on Networking, Communications and Information Technology, 2021.

[7] Richardson M., 2009. Determinants of Used Car Resale Value, Thesis(BSc).,the Colorado College.

[8] [8]3.2.4.3.1.sklearn.ensemble.RandomForestClassifier — scikit-learn 0.19.2 documentation. (n.d.). Retrieved from: http://scikitlearn.org/stable/modules/generated/sklearn.ensemble .RandomForestClassifier.html [accessed: August 30, 2018].

[9] [9] Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. (2001). Support vector clustering. Journal of machine learning research, 2(Dec), 125-137.