Car Price Prediction Project

Submitted by:

Kshitij Chawla

ACKNOWLEDGMENT

In this project different libraries and methods are used that are available in python which helped in completion of the project:

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PowerTransformer.html

http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

http://scikit-learn.org/stable/modules/model_evaluation.html

http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

https://seaborn.pydata.org/generated/seaborn.countplot.html

https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

https://www.analyticsvidhya.com/blog/2020/10/how-to-choose-evaluation-metrics-for-classification-model/

https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

INTRODUCTION

- Business Problem Framing

With the covid 19 impact in the market, we have seen lot of changes in the car market. Now some cars are in demand hence making them costly and some are not in demand hence cheaper. One of our clients works with small traders, who sell used cars. With the change in market due to covid 19 impact, our client is facing problems with their previous car price valuation machine learning models. So, they are looking for new machine learning models from new data. We have to make car price valuation model. This project contains two phase-

Data Collection Phase

You have to scrape at least 5000 used cars data. You can scrape more data as well, it’s up to you. more the data better the model

In this section You need to scrape the data of used cars from websites (Olx, cardekho, Cars24 etc.) You need web scraping for this. You have to fetch data for different locations. The number of

columns for data doesn’t have limit, it’s up to you and your creativity. Generally, these columns are Brand, model, variant, manufacturing year, driven kilometers, fuel, number of owners, location and at last target variable Price of the car. This data is to give you a hint about important variables in used car model. You can make changes to it, you can add or you can remove some columns, it completely depends on the website from which you are fetching the data. Try to include all types of cars in your data for example- SUV, Sedans, Coupe, minivan, Hatchback.

Note – The data which you are collecting is important to us. Kindly don’t share it on any public platforms.

Model Building Phase

After collecting the data, you need to build a machine learning model. Before model building do all data pre-processing steps. Try different models with different hyper parameters and select the best model.

Follow the complete life cycle of data science. Include all the steps like.

1. Data Cleaning
2. Exploratory Data Analysis
3. Data Pre-processing
4. Model Building
5. Model Evaluation
6. Selecting the best model

- Conceptual Background of the Domain Problem

  The project will require knowledge and practice in building Graphs /plots and analysing them to get the relationship between dataset, Knowledge of Different Learning Models to build and predict the required output. Basic Data science concepts to increase the quality of the dataset and Python Knowledge (Coding Language) which will be used to solve the complete Car Prediction project. Understanding of calculating R2 score, RMSE, accuracy, skewness and basic mathematics/statistical approaches will help to build an accurate model for this project.

- Review of Literature

  Used car valuation is the process of finding the fair market value of any used car at present time at a given place. This differs as per make, model, trim, KMs driven, year of purchase, current condition, etc. Valuation of used car cannot have a correct price tag; it will always be a fair range of price.

  A car starts depreciating its value once it moves out of the showroom and with time it depreciates almost half of its original price. The owner of the car also does not have the clear understanding of its fair valuation.

- Motivation for the Problem Undertaken

  I wanted to solve the real-life problem using the Technical skills gathered during the course of being a Data Analyst and improving the skill set.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem----

  Regression Models->

  Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (s) (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables. For example, relationship between rash driving and number of road accidents by a driver is best studied through regression.

  Decision Tree –

  It is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs and utility.

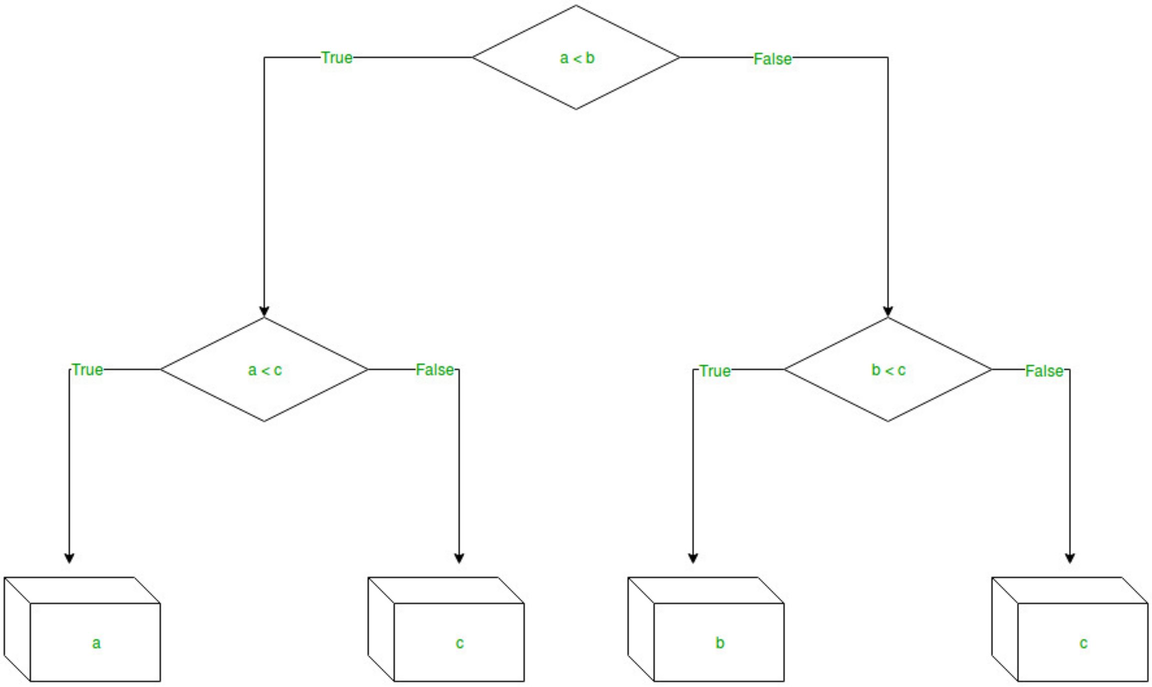
  Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

  The branches/edges represent the result of the node and the nodes have either:

  Conditions [Decision Nodes]

  Result [End Nodes]

  The branches/edges represent the truth/falsity of the statement and takes makes a decision based on that in the example below which shows a decision tree that evaluates the smallest of three numbers:



Random Forest –

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Naive Bayes –

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Linear Regression –

Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

When there is a single input variable (x), the method is referred to as simple linear regression. When there are multiple input variables, literature from statistics often refers to the method as multiple linear regression.

SVM –

Supervised Machine Learning Algorithm used for classification and/or regression. It is more preferred for classification but is sometimes very useful for regression as well. Basically, SVM finds a hyper-plane that creates a boundary between the types of data. In 2-dimensional space, this hyper-plane is nothing but a line.

We used different Plots/ graphs to perform EDA on the dataset->

1) Box Plot: It is a type of chart that depicts a group of numerical data through their quartiles. It is a simple way to visualize the shape of our data. It makes comparing characteristics of data between categories very easy.
2) Count Plot: IT is kind of like a histogram or a bar graph for some categorical area. It simply shows the number of occurrences of an item based on a certain type of category
3) Heat Map: It contains values representing various shades of the same colour for each value to be plotted. Usually the darker shades of the chart represent higher values than the lighter shade. For a very different value a completely different colour can also be used.

4) Scatter Plot: A scatter plot is a diagram where each value in the data set is represented by a dot. The Matplotlib module has a method for drawing scatter plots

- Data Sources and their formats

Below are the fields present in our dataset with the information what these fields describe

COLUMN	DATA_TYPE
myear	float64
fuel	object
km	object
owner	object
transmission	object
brand	object
model	object
variant	object
price	object
city	object

- Data Pre-processing Done

1) First we checked the data set dimensions (We have 7151 rows and 11 columns)

```
: df.shape
: (7151, 11)
```

2) Then we checked whether there is any repeating data available

```
duplicate = df.duplicated()
print(duplicate.sum())
df[duplicate]
0
```

3) We checked the outliers using the Box Plot and replaced the outliers with more appropriate values. Removal of outliers

can also be done but taking the Data Loss percentage into consideration It is better to replace the outlier

- Hardware and Software Requirements and Tools Used

1) Software: Jupyter Notebook - To code and build the project in python
2) Libraries:
    a) numpy - To perform basic math operations
    b) pandas - To perform basic File operations
    c) Matplotlib - To plot Different Graphs/ Plots
    d) Seaborn - Advance library to enhance the quality of graphs/plots
    e) warnings - To ignore the unwanted warnings raised while interpreting the code
    f) sklearn - To build the Prediction models
    g) imblearn - To balance our dataset distribution

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  We used different approaches from checking the dataset quality to building the model. We checked the null values and repeated rows in the dataset. For checking the Outliers, we used Box Plot and to remove the outliers we used IQR method. Then we moved to next step of checking data distribution and skewness. To scale the data, we used MinMax Scaler method and to remove the skewness we first checked the log and square root method but skewness of the dataset was not getting removed from it so we performed the Power transform to remove skewness.. We started building different models and checked their R2 score and selected the best suited model to perform Hyper tuning on. We got Random Forest Algo with the best result and after performing Hyper tuning we finalized the model.

- Testing of Identified Approaches (Algorithms)

  1) Linear Regression
  2) Decision Tree
  3) Elastic Net
  4) Lasso
  5) Random Forest
  6) Ridge

- ## Run and Evaluate selected models

Linear Regression

In [70]:
```
from sklearn import metrics
regr = LinearRegression()
regr.fit(x_train, y_train)
pred=regr.predict(x_test)
print('R2 score',r2_score(y_test, pred))
print('MAE:', metrics.mean_absolute_error(y_test, pred))
print('MSE:', metrics.mean_squared_error(y_test, pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.6202674492744558
MAE: 0.48815064071651004
MSE: 0.3945003290810261
RMSE: 0.6280926118662964
```

Ridge

In [71]:
```
rr = Ridge(alpha=0.01)
rr.fit(x_train, y_train)
pred=rr.predict(x_test)
print('R2 score',r2_score(y_test, pred))
print('MAE:', metrics.mean_absolute_error(y_test, pred))
print('MSE:', metrics.mean_squared_error(y_test, pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.6202673953963928
MAE: 0.4881508000019625
MSE: 0.39450038505440405
RMSE: 0.6280926564245152
```

Lasso

In [72]:
```
model_lasso = Lasso(alpha=0.01)
model_lasso.fit(x_train, y_train)
pred=model_lasso.predict(x_test)
print('R2 score',r2_score(y_test, pred))
print('MAE:', metrics.mean_absolute_error(y_test, pred))
print('MSE:', metrics.mean_squared_error(y_test, pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.6194658467308831
MAE: 0.48988157757124656
MSE: 0.3953331059041543
RMSE: 0.6287552034807778
```

Elastic Net

In [73]:
```
model_enet = ElasticNet(alpha = 0.01)
model_enet.fit(x_train, y_train)
pred=model_enet.predict(x_test)
print('R2 score',r2_score(y_test, pred))
print('MAE:', metrics.mean_absolute_error(y_test, pred))
print('MSE:', metrics.mean_squared_error(y_test, pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.6197703017133225
MAE: 0.4893521332974955
MSE: 0.39501681068391786
RMSE: 0.6285036282185791
```

Decision Tree Regression

In [74]:
```
from sklearn.tree import DecisionTreeRegressor
from sklearn import metrics
dtr=DecisionTreeRegressor()
dtr.fit(x_train,y_train)
pred=dtr.predict(x_test)
print('R2 score', r2_score(y_test, pred))
print('MAE:', metrics.mean_absolute_error(y_test, pred))
print('MSE:', metrics.mean_squared_error(y_test, pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.8446425341046835
MAE: 0.17782074133578957
MSE: 0.1613993093396768
RMSE: 0.4017453289581309
```

Random Forest Regression

In [75]:
```
from sklearn.ensemble import RandomForestRegressor
rdr = RandomForestRegressor()
rdr.fit(x_train,y_train)
pred1=rdr.predict(x_test)
print('R2 score',r2_score(y_test, pred1))
print('MAE:', metrics.mean_absolute_error(y_test, pred1))
print('MSE:', metrics.mean_squared_error(y_test, pred1))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test,pred1)))
```

```
R2 score 0.9065771666954345
MAE: 0.16004338581527627
MSE: 0.09705604223792372
RMSE: 0.31153818744725936
```

- Key Metrics for success in solving problem under consideration

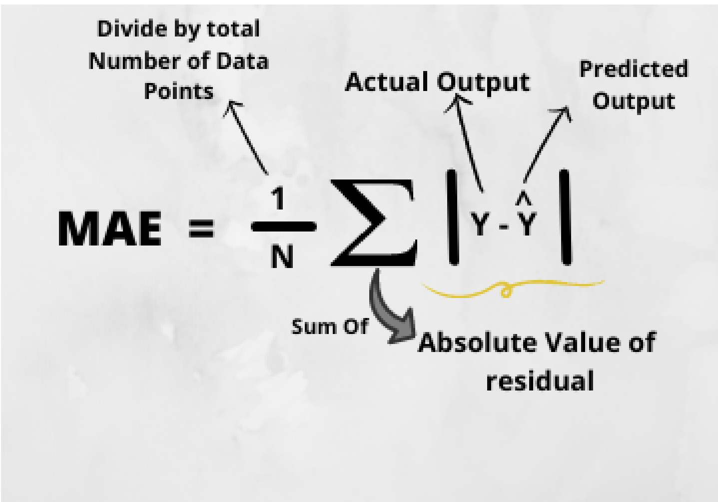
1) Mean Absolute Error(MAE)

MAE is a very simple metric which calculates the absolute difference between actual and predicted values.

To better understand, let's take an example you have input data and output data and use Linear Regression, which draws a best-fit line.

Now you have to find the MAE of your model which is basically a mistake made by the model known as an error. Now find the difference between the actual value and predicted value that is an absolute error but we have to find the mean absolute of the complete dataset.

so, sum all the errors and divide them by a total number of observations And this is MAE. And we aim to get a minimum MAE because this is a loss.



Advantages of MAE

The MAE you get is in the same unit as the output variable.

It is most Robust to outliers.

Disadvantages of MAE

The graph of MAE is not differentiable so we have to apply various optimizers like Gradient descent which can be differentiable.

```
from sklearn.metrics import mean_absolute_error
print("MAE",mean_absolute_error(y_test,y_pred))
```

Now to overcome the disadvantage of MAE next metric came as MSE.

2) Mean Squared Error(MSE)

MSE is a most used and very simple metric with a little bit of change in mean absolute error. Mean squared error states that finding the squared difference between actual and predicted value.

So, above we are finding the absolute difference and here we are finding the squared difference.

What actually the MSE represents? It represents the squared distance between actual and predicted values. we perform squared to avoid the cancellation of negative terms and it is the benefit of MSE.

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \hat{y}\right)^2}_{\text{The square of the difference between actual and predicted}}$$

Advantages of MSE

The graph of MSE is differentiable, so you can easily use it as a loss function.

Disadvantages of MSE

The value you get after calculating MSE is a squared unit of output. for example, the output variable is in meter(m) then after calculating MSE the output we get is in meter squared.

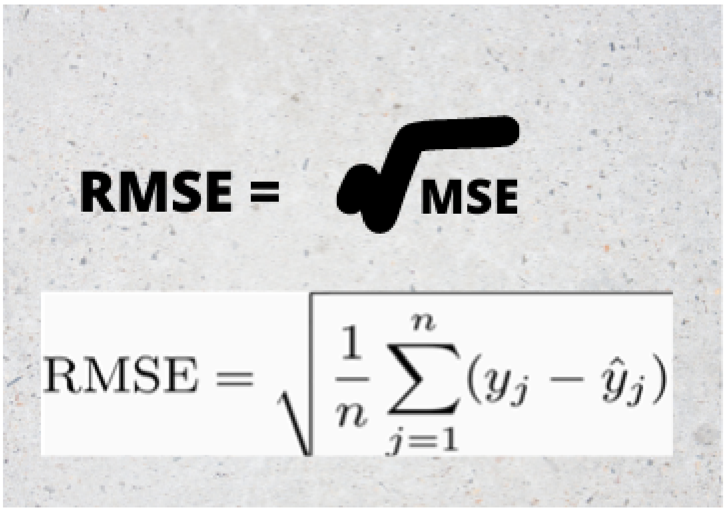
If you have outliers in the dataset then it penalizes the outliers most and the calculated MSE is bigger. So, in short, It is not Robust to outliers which were an advantage in MAE.

```
from sklearn.metrics import mean_squared_error
```

print("MSE",mean_squared_error(y_test,y_pred))

3) Root Mean Squared Error(RMSE)

As RMSE is clear by the name itself, that it is a simple square root of mean squared error.



Advantages of RMSE

The output value you get is in the same unit as the required output variable which makes interpretation of loss easy.

Disadvantages of RMSE

It is not that robust to outliers as compared to MAE.

for performing RMSE we have to NumPy NumPy square root function over MSE.

print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred)))

Most of the time people use RMSE as an evaluation metric and mostly when you are working with deep learning techniques the most preferred metric is RMSE.

4) Root Mean Squared Log Error(RMSLE)

Taking the log of the RMSE metric slows down the scale of error. The metric is very helpful when you are developing a model without calling the inputs. In that case, the output will vary on a large scale.

To control this situation of RMSE we take the log of calculated RMSE error and resultant we get as RMSLE.

To perform RMSLE we have to use the NumPy log function over RMSE.

```
print("RMSE",np.log(np.sqrt(mean_squared_error(y_test,y_pred))))
```

It is a very simple metric that is used by most of the datasets hosted for Machine Learning competitions.

5) R Squared (R2)

R2 score is a metric that tells the performance of your model, not the loss in an absolute sense that how many wells did your model perform.

In contrast, MAE and MSE depend on the context as we have seen whereas the R2 score is independent of context.

So, with help of R squared we have a baseline model to compare a model which none of the other metrics provides. The same we have in classification problems which we call a threshold which is fixed at 0.5. So basically R2 squared calculates how must regression line is better than a mean line.

Hence, R2 squared is also known as Coefficient of Determination or sometimes also known as Goodness of fit.

$$\textbf{R2 Squared} = 1 - \frac{SSr}{SSm}$$

SSr = Squared sum error of regression line

SSm = Squared sum error of mean line

R2 Squared

Now, how will you interpret the R2 score? suppose If the R2 score is zero then the above regression line by mean line is equal means 1 so 1-1 is zero. So, in this case, both lines are overlapping means model performance is worst, It is not capable to take advantage of the output column.

Now the second case is when the R2 score is 1, it means when the division term is zero and it will happen when the regression line does not make any mistake, it is perfect. In the real world, it is not possible.

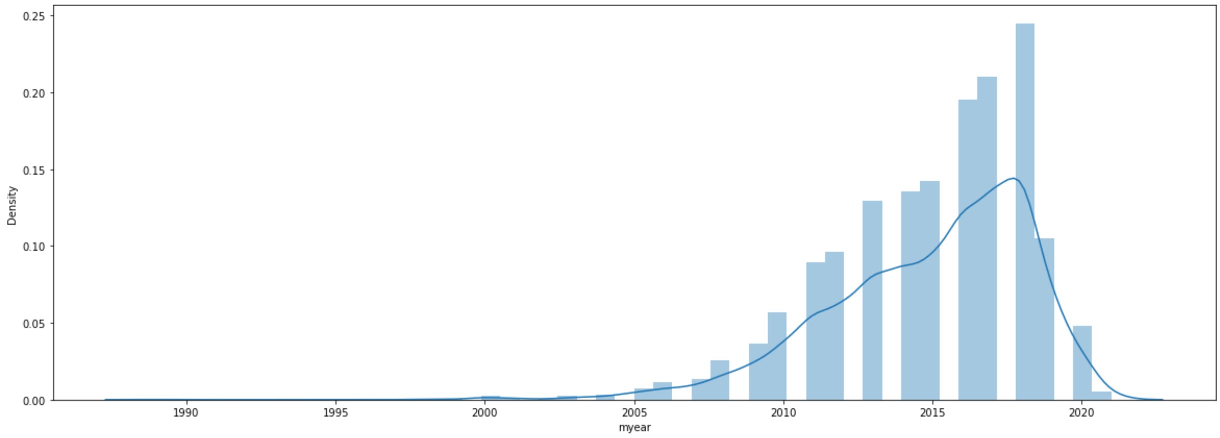
So we can conclude that as our regression line moves towards perfection, R2 score move towards one. And the model performance improves.

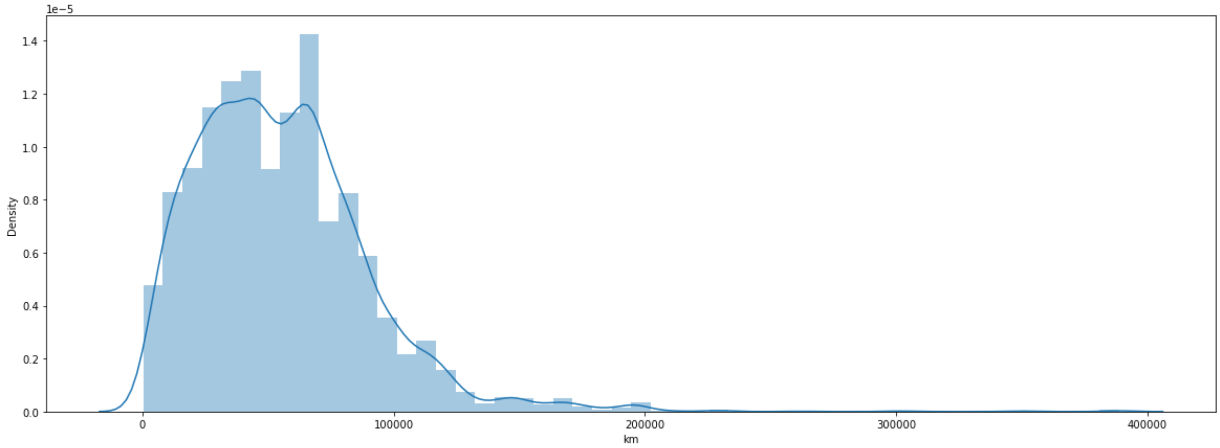
The normal case is when the R2 score is between zero and one like 0.8 which means your model is capable to explain 80 per cent of the variance of data.

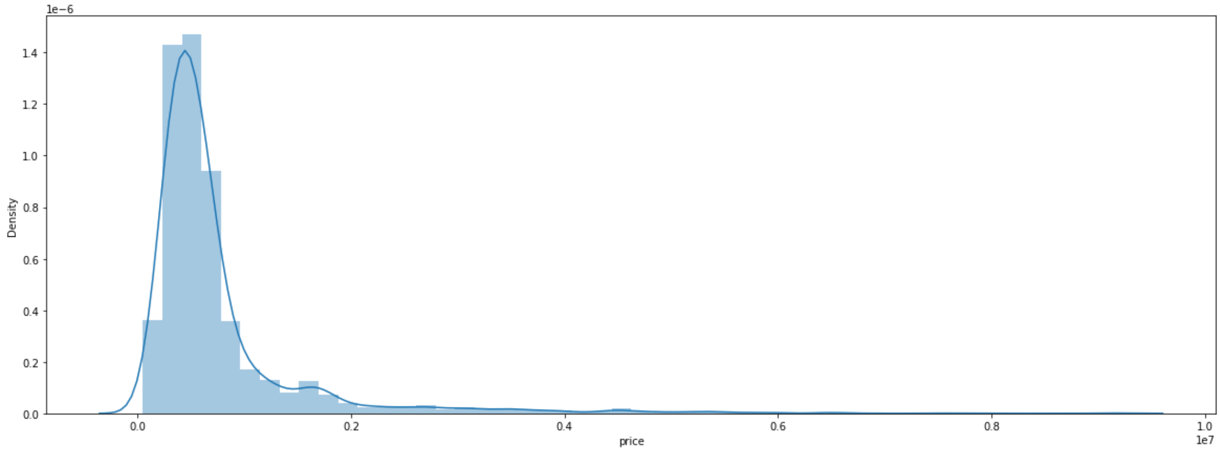
```
from sklearn.metrics import r2_score

r2 = r2_score(y_test,y_pred)

print(r2)
```

- Visualizations

In [14]:
```
counter=1;
for i in range(0,len(continous_columns)):
        plt.figure(figsize=(20,500))
        plt.subplot(60,1,counter)
        counter=counter+1
        sns.distplot(df[continous_columns[i]])
        plt.show()
```





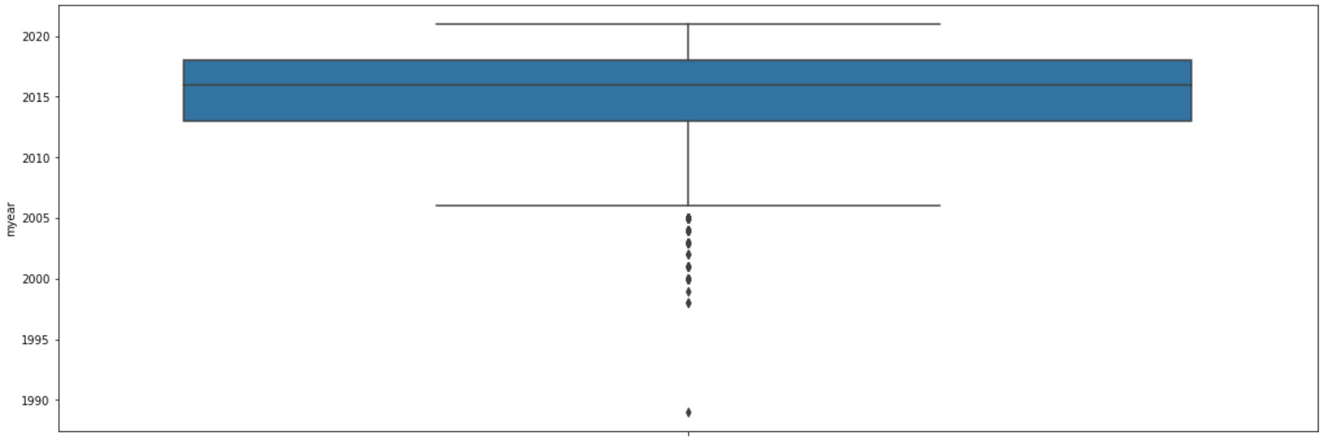


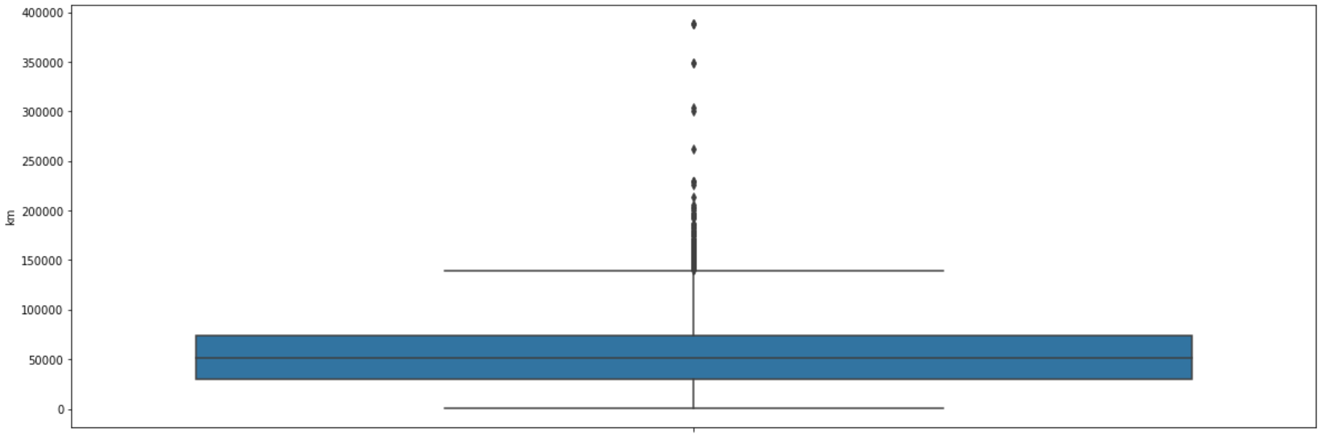
Findings:

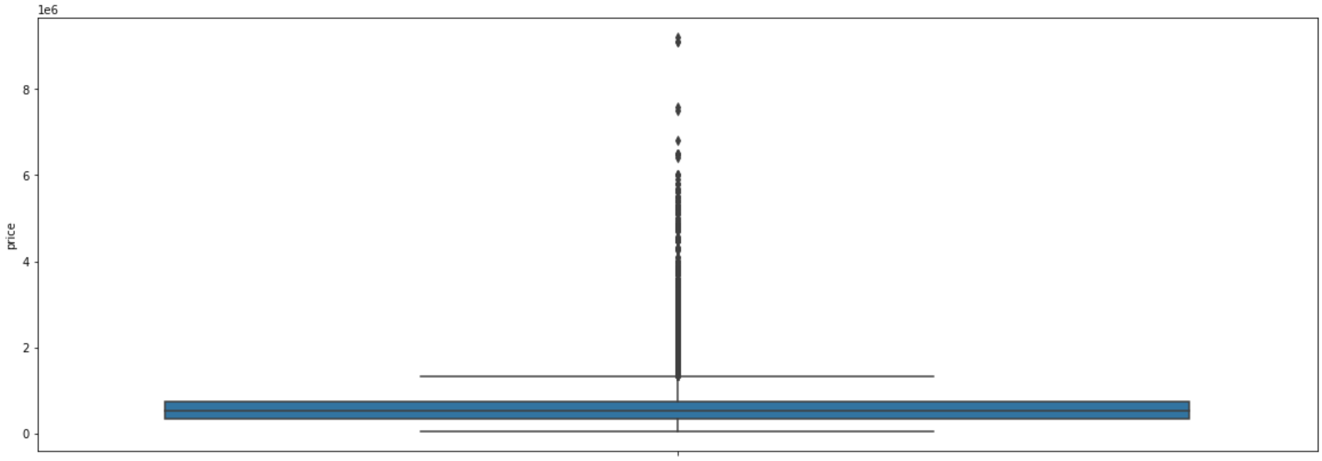
1) most of the cars were maked between 2015-2020

2) most of the cars have travelled less than 100000 and the data in km is not normally distrbiuted

In [15]:
```
counter=1;
for i in range(0,len(continous_columns)):
        plt.figure(figsize=(20,500))
        plt.subplot(60,1,counter)
        counter=counter+1
        sns.boxplot(y=continous_columns[i],hue = continous_columns[i],data=df)
        #sns.boxplot(df[columns[i]])
        plt.show()
```

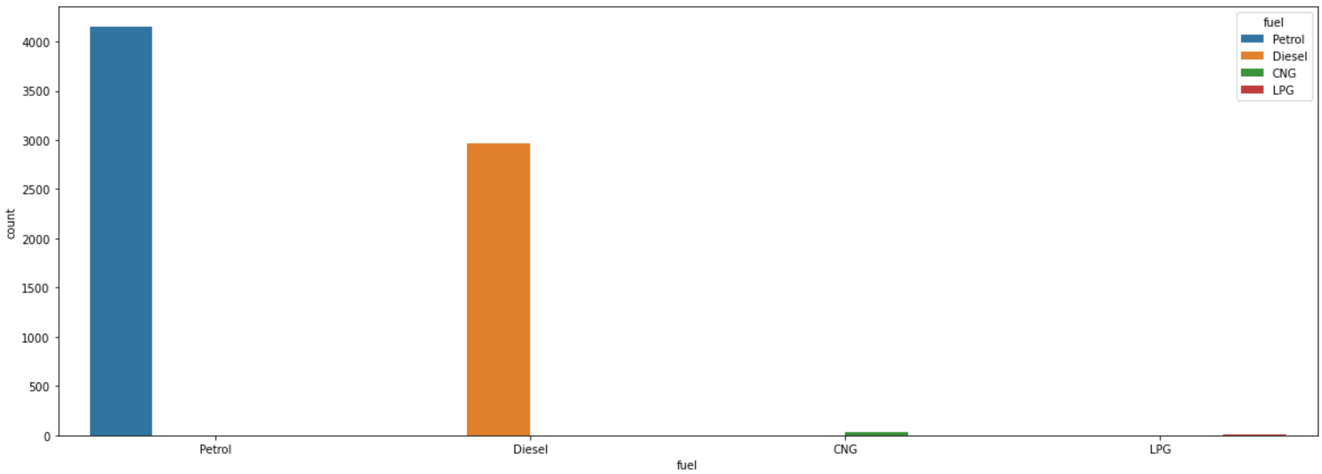


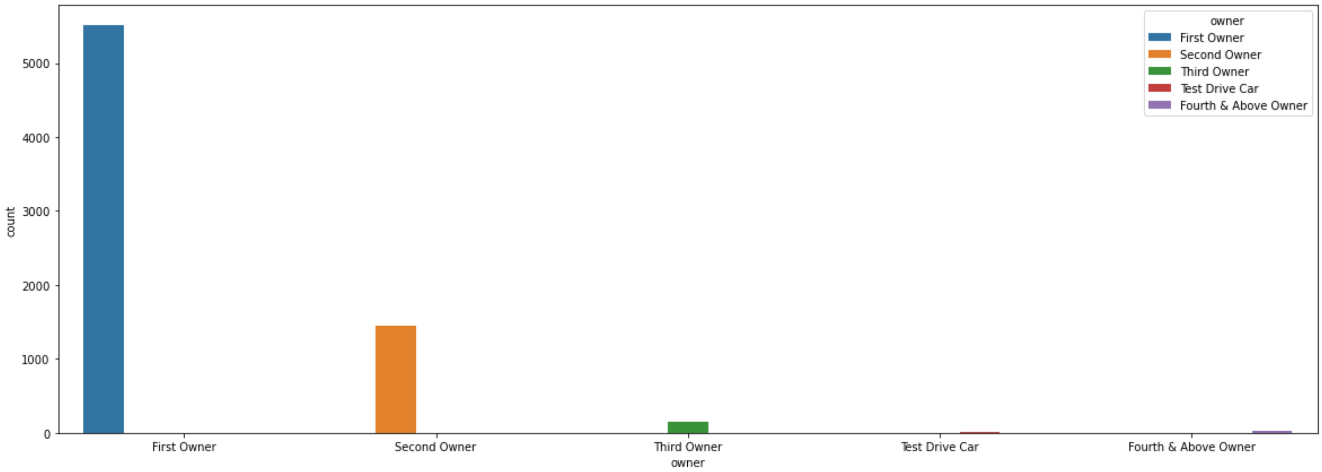


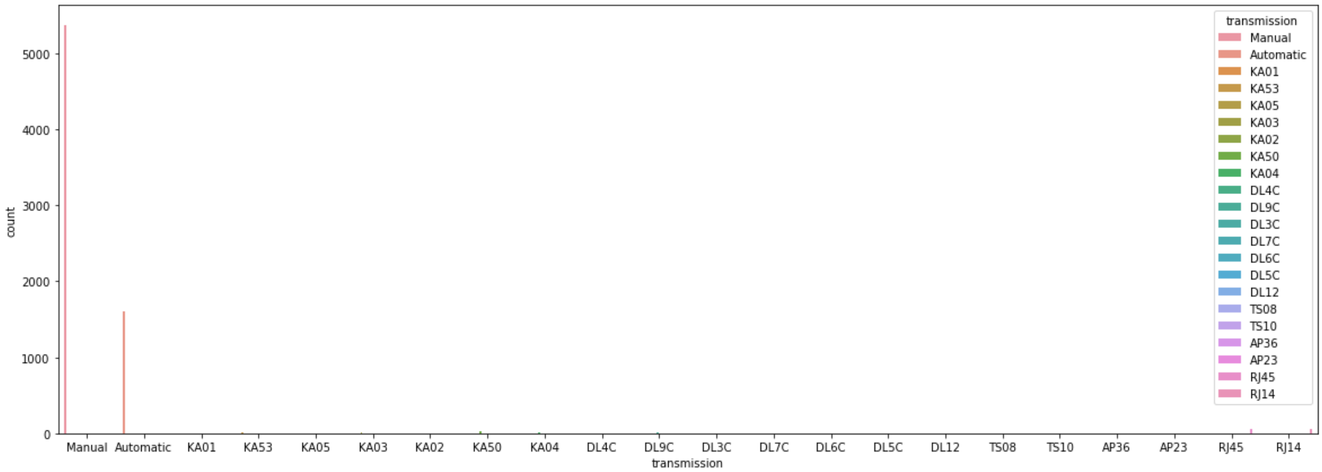


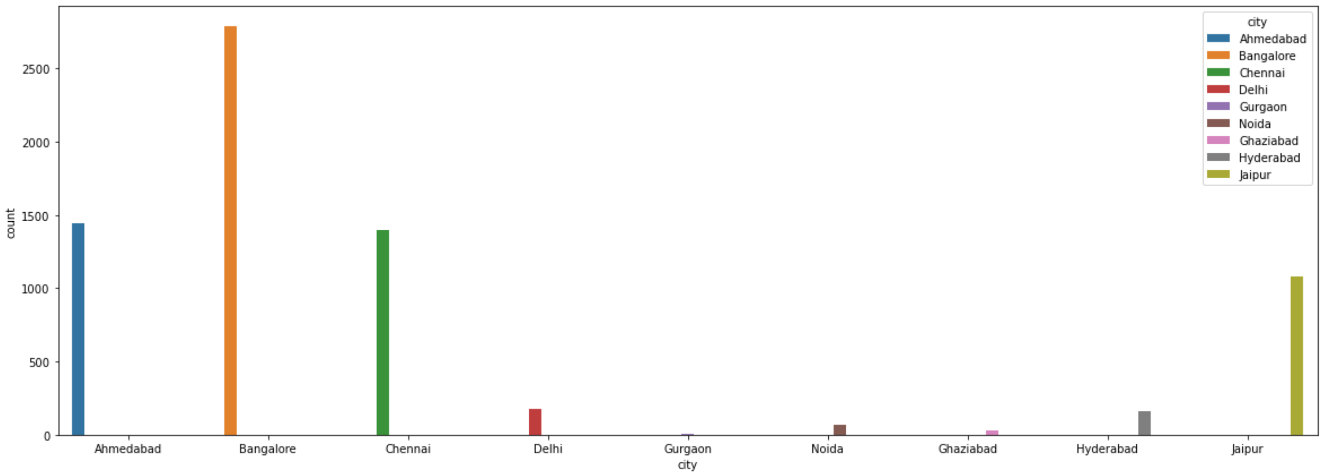
Findings:
columns {myear, price and km} in the dataset have outliers present in them.

```
In [18]: counter=1;

for column in categorical_columns:
        plt.figure(figsize=(20,500))
        plt.subplot(60,1,counter)
        counter=counter+1
        sns.countplot(x=column,hue=column,data=df)
        plt.show()
```







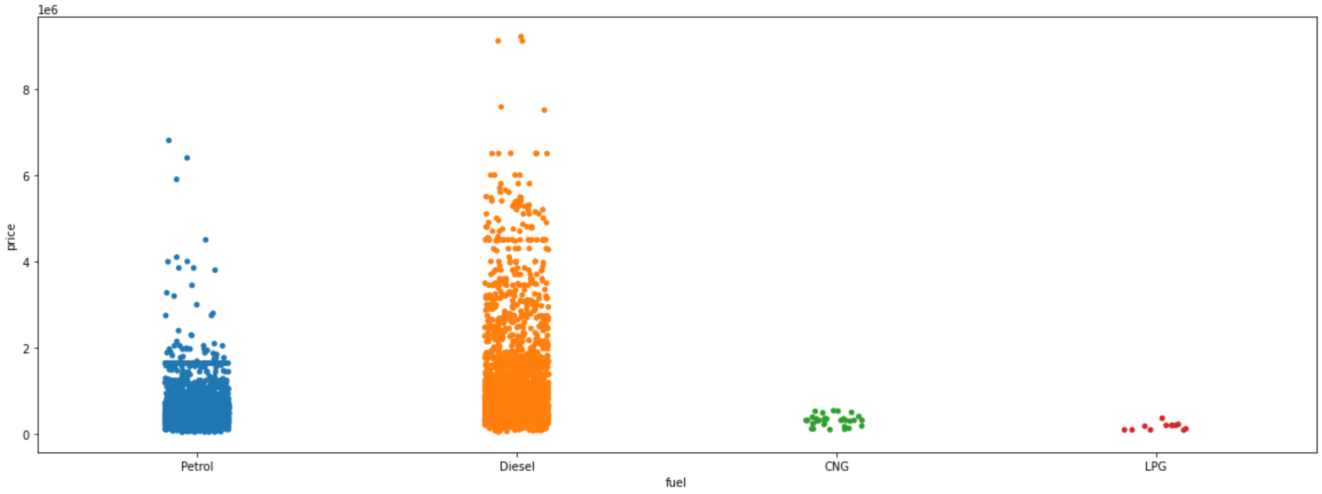


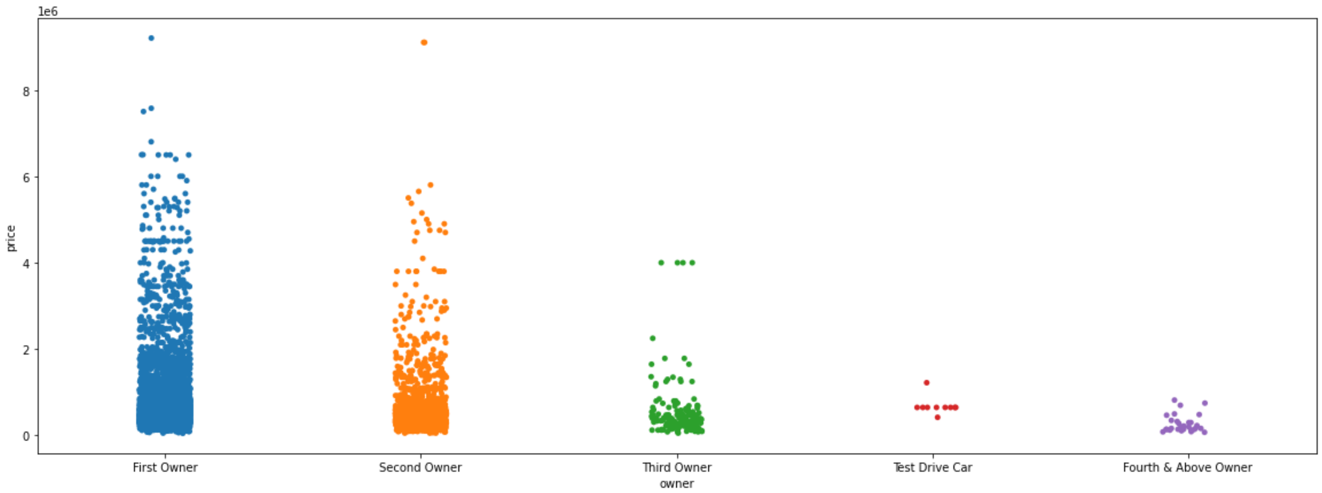
Findings:

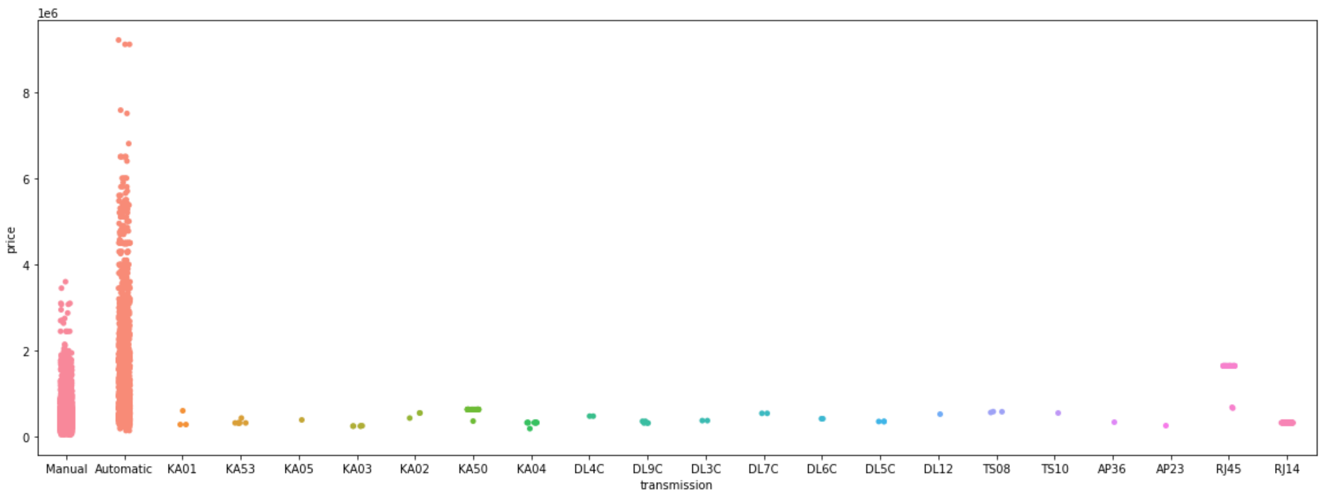
In the dataset we have mostly petrol cars
In the dataset we have mostly 1st owned cars
In the dataset most of the cars are from Bangalore city
In the dataset we have mostly Manual transmission

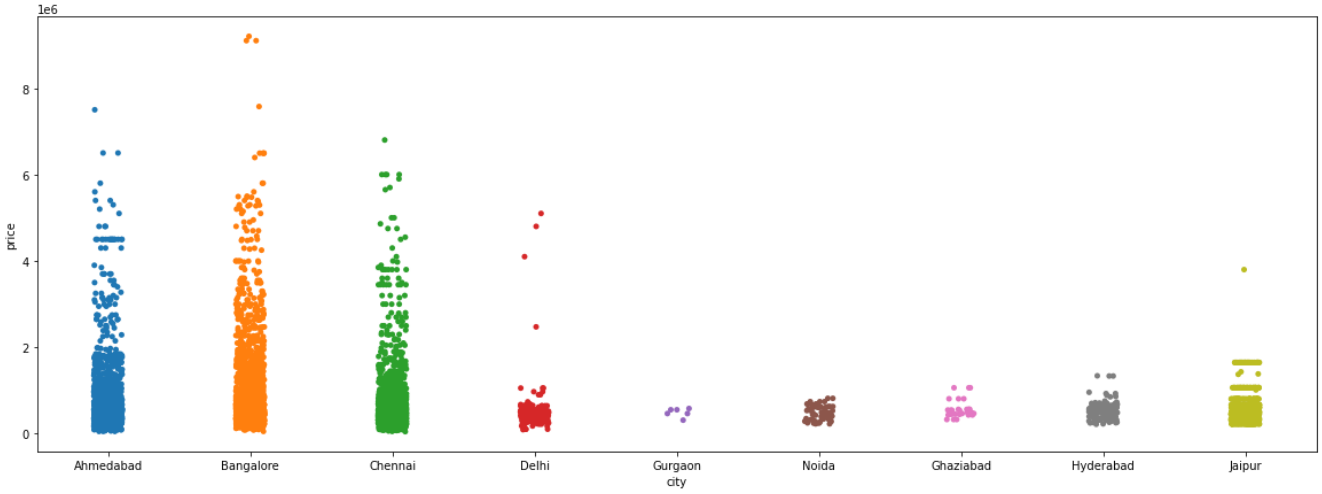
In [24]:

```
counter=1;


for column in categorical_columns:
        plt.figure(figsize=(20,500))
        plt.subplot(60,1,counter)
        counter=counter+1
        sns.stripplot(x=column, y="SalePrice", data=df)
        plt.show()
```









Findings:

1) Automatic transmission cars have higher price rate then other tramissions
2) First owned cars have higher price rate then other
3) Diesel cars have hight price then other fueled cars
4) Cars sold in Bangalore city have higher price rate

Interpretation of the Results

Results:

1) Large amount of null values are present in the dataset
2) Data Set is not normally distributed
3) Dataset have outliers in most of the variables
4) Dataset is not normalized
5) Dataset is skewed
6) Random Forest Algorithm is best suited for the current dataset

CONCLUSION

- Key Findings and Conclusions of the Study

  We found that to predict the Car price using Data Science the best way after performing Data Cleaning is to use Random Forest Algorithm it provides 90% accuracy which is better than other Regression algorithms.

- Learning Outcomes of the Study in respect of Data Science

  In data science, there are various steps involved during Data analysis and cleaning. With the help of various Visualization tools like plots, Graphs we were able to perform the actions and observe different things. Like for finding the outliers we used Box Plot visualization, for finding the skewness and normalization we used Count Plot visualization, for finding skewness we visualized the skewness using Heat Map for the clear picture of how the variables are co-related to each other in the dataset. We used different metrics to check which model best fits the prediction for the dataset.

- Limitations of this work and Scope for Future Work

  Data was unbalanced if data was balanced more accurate and clear picture of the output -> result is dependent on the data

  Neural network classifier which are still unexplored & can be taken for future consideration