

# Internship Report

## Car Sales Purchase Prediction

Name: Ankita Gupta

Course: ML1119

Duration: 2 Weeks

Problem Statement: You are working as a car salesman and you would like to develop a Machine Learning model to predict the total amount that customers are willing to pay.

The model should predict: Car Purchase Amount

---

Github link: <https://github.com/Ankita30-ui/Car-Sales-Purchase-Prediction>

Dataset used

---

The data source used for this project is car.csv The car.csv Data Set contains attributes like: Customer Name, Customer e-mail, Country, Gender, Age, Annual Salary, Credit Card Debt, Net Worth.

Link: <https://drive.google.com/drive/folders/1Dzj0gD6irtFA97BkBlpd3lnwWRPzHBHp?usp=sharing>

Applying algorithms

---

Sequential modelling

---

## 1.Importing the libraries:

```
ML%20%26AJ/Heart%20disease%20prediction/Car_sales_purchasing_pred.ipynb#f

jupyter Car_sales_purchasing_pred Last Checkpoint: a day ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

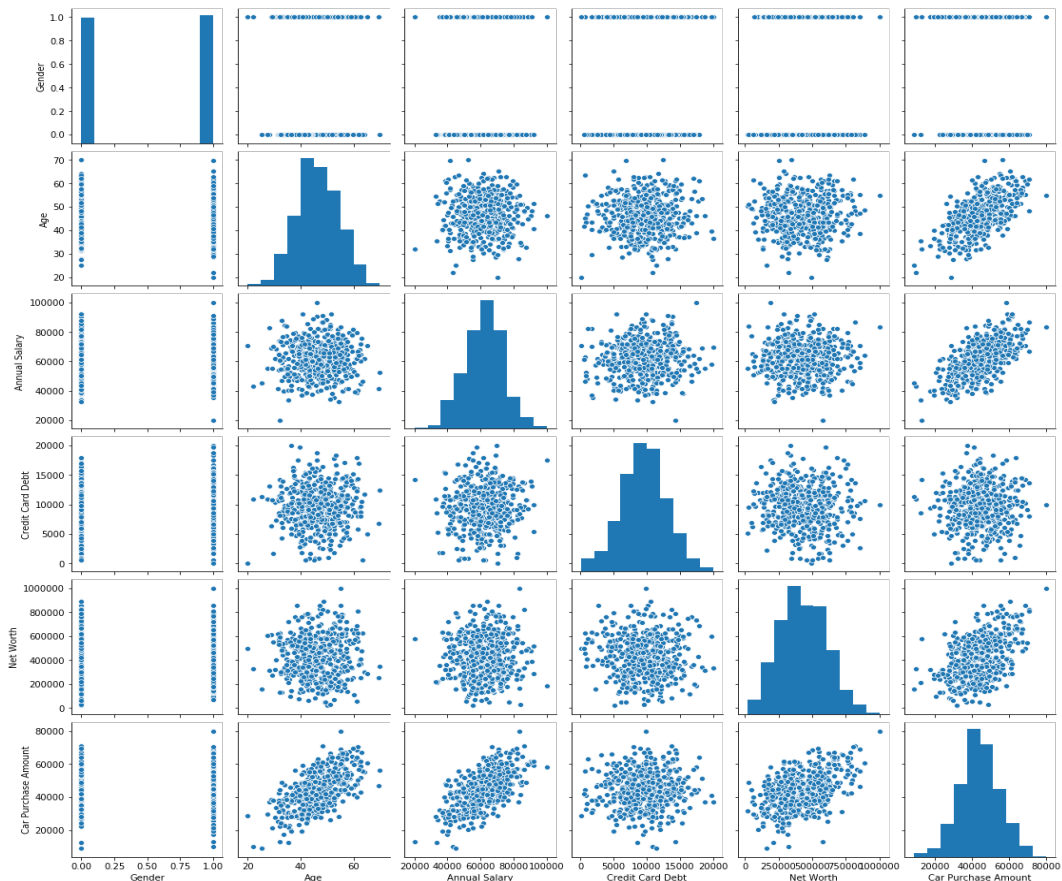
Loading the dataset

In [2]: car_df=pd.read_csv('/home/rupeek/Desktop/ML &AI/Heart disease prediction/Car_Purchasing_Data.csv', encoding='ISO-8859-1')

In [3]: car_df
Out[3]:
```

	Customer Name	Customer e-mail	Country	Gender	Age	Annual Salary	Credit Card Debt	Net Worth	Car Purchase Amount
0	Martina Avila	cubilla.Curae.Phasellus@quisaccusancorvallis.edu	Bulgaria	0	41.851720	62812.09301	11609.380910	238961.2505	35321.45877
1	Harlan Barnes	eu.dolor@diam.co.uk	Belize	0	40.870623	66646.89292	9572.957138	530973.9078	45115.52566
2	Nammi Rodriguez	vulputate.mauris.sagittis@ametconsecteturadip...	Algeria	1	43.152897	53798.55112	11160.355060	638467.1773	42925.70921
3	Jade Cunningham	malesuada@dngrissim.com	Cook Islands	1	58.271369	79370.03798	14426.164850	548599.0524	67422.36313
4	Cedric Leach	felis.utlancorper.viverra@eggetmolislectus.net	Brazil	1	57.313749	59729.15130	5358.712177	560304.0671	55915.46248
...	...	...	...	...	...	...	...	...	...
495	Walter	ligula@Cumsociis.ca	Nepal	0	41.462515	71942.40291	6995.902524	541670.1016	48901.44342
496	Vanna	Cum.sociis.natoque@Sedmolislectus.edu	Zimbabwe	1	37.642000	56039.49793	12301.456790	360419.0988	31491.41457
497	Pearl	penatibus.et@massanorante.com	Philippines	1	53.943497	68888.77805	10611.606860	764531.3203	64147.28888
498	Nell	Quisque.varius@arcuVivamusit.net	Botswana	1	59.160509	49811.99062	14013.034510	337826.6382	45442.15353
499	Maria	Camaron.maria@hotmail.com	malta	1	46.731152	61370.67766	9391.341628	462946.4924	45107.22566

## 3. Visualizing data:



### 3. creating testing and training dataset

```
In [7]: X = car_df.drop(['Customer Name', 'Customer e-mail', 'Country', 'Car Purchase Amount'], axis = 1)

In [8]: print(X)

   Gender  Age  Annual Salary  Credit Card Debt  Net Worth
0      0  41.851720    62812.09301    11609.380910    238961.2505
1      1  40.870623    66646.89292    9572.957136    530973.9078
2      1  43.152897    53798.55112    11160.355060    638467.1773
3      1  58.271369    79370.03798    14426.164850    548599.0524
4      1  57.313749    59729.15130    5358.712177    560304.0071
...    ...    ...    ...    ...    ...
495     0  41.462515    71942.40291    6995.982524    541670.1016
496     1  37.642000    56039.49793    12301.456790    360419.0988
497     1  52.943497    68808.77805    18611.606860    764531.3203
498     1  59.160509    49811.99062    14013.034510    337826.6382
499     1  46.731152    61370.67766    9391.341628    462946.4924

[500 rows x 5 columns]

In [9]: y = car_df['Car Purchase Amount']
y.shape

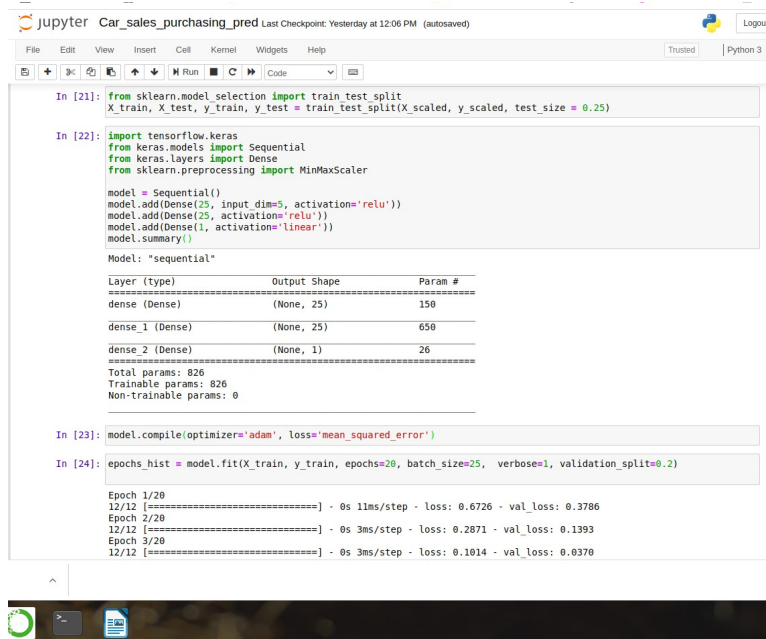
Out[9]: (500,)
```

```
In [10]: from sklearn.preprocessing import MinMaxScaler
scaler_x = MinMaxScaler()
X_scaled = scaler_x.fit_transform(X)

In [11]: scaler_x.data_max_

Out[11]: array([1.e+08, 7.e+01, 1.e+05, 2.e+04, 1.e+06])
```

### 4. Training the model



```
jupyter Car_sales_purchasing_pred Last Checkpoint: Yesterday at 12:06 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [21]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y_scaled, test_size = 0.25)

In [22]: import tensorflow.keras
from keras.models import Sequential
from keras.layers import Dense
from sklearn.preprocessing import MinMaxScaler

model = Sequential()
model.add(Dense(25, input_dim=5, activation='relu'))
model.add(Dense(25, activation='relu'))
model.add(Dense(1, activation='linear'))
model.summary()

Model: "sequential"
Layer (type) Output Shape Param #
=====
dense (Dense) (None, 25) 150
dense_1 (Dense) (None, 25) 650
dense_2 (Dense) (None, 1) 26
=====
Total params: 826
Trainable params: 826
Non-trainable params: 0

In [23]: model.compile(optimizer='adam', loss='mean_squared_error')

In [24]: epochs_hist = model.fit(X_train, y_train, epochs=20, batch_size=25, verbose=1, validation_split=0.2)

Epoch 1/20
12/12 [=====] - 0s 11ms/step - loss: 0.6726 - val_loss: 0.3786
Epoch 2/20
12/12 [=====] - 0s 3ms/step - loss: 0.2871 - val_loss: 0.1393
Epoch 3/20
12/12 [=====] - 0s 3ms/step - loss: 0.1014 - val_loss: 0.0370
```

```

jupyter Car_sales_purchasing_pred Last Checkpoint: Yesterday at 12:06 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
Epoch 1/20
12/12 [=====] - 0s 11ms/step - loss: 0.6726 - val_loss: 0.3786
Epoch 2/20
12/12 [=====] - 0s 3ms/step - loss: 0.2871 - val_loss: 0.1393
Epoch 3/20
12/12 [=====] - 0s 3ms/step - loss: 0.1014 - val_loss: 0.0370
Epoch 4/20
12/12 [=====] - 0s 2ms/step - loss: 0.0296 - val_loss: 0.0126
Epoch 5/20
12/12 [=====] - 0s 3ms/step - loss: 0.0149 - val_loss: 0.0157
Epoch 6/20
12/12 [=====] - 0s 2ms/step - loss: 0.0150 - val_loss: 0.0134
Epoch 7/20
12/12 [=====] - 0s 2ms/step - loss: 0.0128 - val_loss: 0.0111
Epoch 8/20
12/12 [=====] - 0s 3ms/step - loss: 0.0117 - val_loss: 0.0105
Epoch 9/20
12/12 [=====] - 0s 2ms/step - loss: 0.0109 - val_loss: 0.0101
Epoch 10/20
12/12 [=====] - 0s 2ms/step - loss: 0.0100 - val_loss: 0.0097
Epoch 11/20
12/12 [=====] - 0s 2ms/step - loss: 0.0094 - val_loss: 0.0094
Epoch 12/20
12/12 [=====] - 0s 3ms/step - loss: 0.0088 - val_loss: 0.0089
Epoch 13/20
12/12 [=====] - 0s 2ms/step - loss: 0.0081 - val_loss: 0.0082
Epoch 14/20
12/12 [=====] - 0s 3ms/step - loss: 0.0074 - val_loss: 0.0076
Epoch 15/20
12/12 [=====] - 0s 2ms/step - loss: 0.0068 - val_loss: 0.0070
Epoch 16/20
12/12 [=====] - 0s 3ms/step - loss: 0.0064 - val_loss: 0.0065
Epoch 17/20
12/12 [=====] - 0s 2ms/step - loss: 0.0059 - val_loss: 0.0061
Epoch 18/20
12/12 [=====] - 0s 2ms/step - loss: 0.0054 - val_loss: 0.0056
Epoch 19/20
12/12 [=====] - 0s 2ms/step - loss: 0.0050 - val_loss: 0.0053
Epoch 20/20
12/12 [=====] - 0s 3ms/step - loss: 0.0047 - val_loss: 0.0049

```

## 5. Evaluating the model:

```

jupyter Car_sales_purchasing_pred Last Checkpoint: Yesterday at 12:06 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [25]: print(epochs_hist.history.keys())

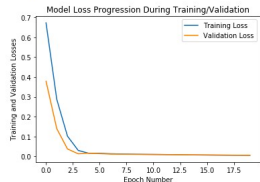
dict_keys(['loss', 'val_loss'])

In [26]: plt.plot(epochs_hist.history['loss'])
plt.plot(epochs_hist.history['val_loss'])

plt.title('Model Loss Progression During Training/Validation')
plt.ylabel('Training and Validation Losses')
plt.xlabel('Epoch Number')
plt.legend(['Training Loss', 'Validation Loss'])

Out[26]: <matplotlib.legend.Legend at 0x7fb7687e2cd0>

```



```

FINAL RESULT

In [27]: # Gender, Age, Annual Salary, Credit Card Debt, Net Worth
X_test=np.array([[1,50,50000,10000,600000]])
y_predict=model.predict(X_test)

In [28]: print('Expected Purchase Amount for Car:',y_predict)

Expected Purchase Amount for Car: [[98529.44]]

```

## 6. Final output:

```

FINAL RESULT

In [27]: # Gender, Age, Annual Salary, Credit Card Debt, Net Worth
X_test=np.array([[1,50,50000,10000,600000]])
y_predict=model.predict(X_test)

In [28]: print('Expected Purchase Amount for Car:',y_predict)

Expected Purchase Amount for Car: [[98529.44]]

```

