In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Make numpy printouts easier to read.
np.set_printoptions(precision=3, suppress=True)
```

In [2]:

```
import tensorflow as tf

from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.layers.experimental import preprocessing

print(tf.__version__)
```

```
2.11.0
```

The Auto MPG dataset

In [3]:

```
url = 'http://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.da
column_names = ['MPG', 'Cylinders', 'Displacement', 'Horsepower', 'Weight',
                'Acceleration', 'Model Year', 'Origin']

raw_dataset = pd.read_csv(url, names=column_names,
                          na_values='?', comment='\t',
                          sep=' ', skipinitialspace=True)
```

In [5]:

```
carsdata = raw_dataset.copy()
carsdata.tail()
```

Out[5]:

	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model Year	Origin
393	27.0	4	140.0	86.0	2790.0	15.6	82	1
394	44.0	4	97.0	52.0	2130.0	24.6	82	2
395	32.0	4	135.0	84.0	2295.0	11.6	82	1
396	28.0	4	120.0	79.0	2625.0	18.6	82	1
397	31.0	4	119.0	82.0	2720.0	19.4	82	1

Clean the Data

In [7]:

```
carsdata.isna().sum()
```

Out[7]:

```
MPG             0
Cylinders       0
Displacement    0
Horsepower      6
Weight          0
Acceleration    0
Model Year      0
Origin          0
dtype: int64
```

In [10]:

```
carsdata = carsdata.replace("?", np.nan)
```

In [12]:

```
carsdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   MPG           398 non-null    float64
 1   Cylinders     398 non-null    int64  
 2   Displacement  398 non-null    float64
 3   Horsepower    392 non-null    float64
 4   Weight        398 non-null    float64
 5   Acceleration  398 non-null    float64
 6   Model Year    398 non-null    int64  
 7   Origin        398 non-null    int64  
dtypes: float64(5), int64(3)
memory usage: 25.0 KB
```

In [14]:

```
carsdata = carsdata.fillna(carsdata.median())
```

In [16]:

```
carsdata["Horsepower"] = carsdata["Horsepower"].astype('float64')
```

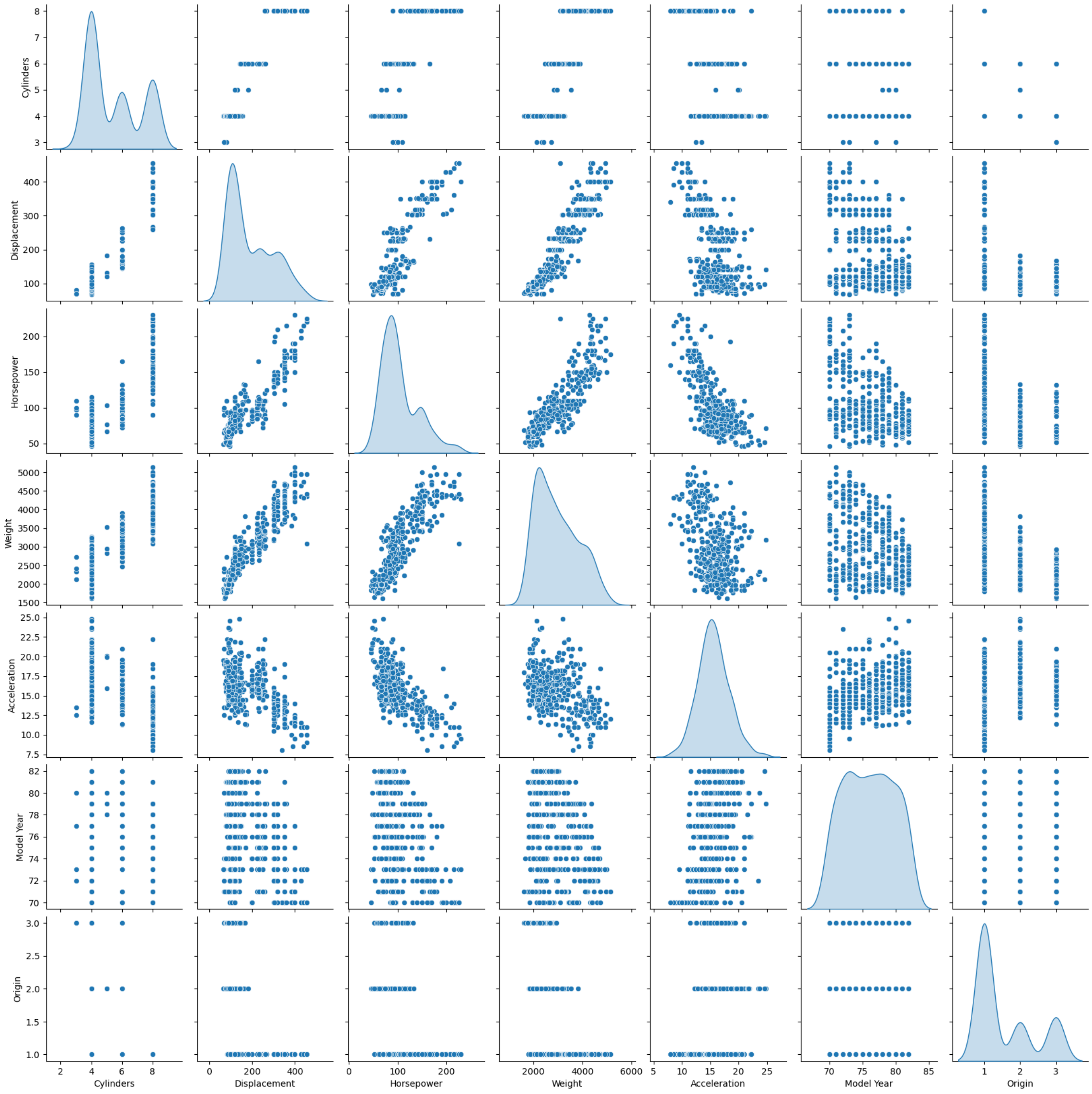
In [19]:

```python
X = carsdata.drop(['MPG'], axis=1)
y = carsdata[['MPG']]

sns.pairplot(X, diag_kind= 'kde')
```

Out[19]:

```
<seaborn.axisgrid.PairGrid at 0x17c658efb50>
```



Scaling

In [23]:

```python
x = carsdata.iloc[:, :-1].values
y = carsdata.iloc[:, -1].values
```

In [25]:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
scaler = sc.fit_transform(x)
```

splitting

In [26]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(scaler,y,random_state=10,test_size=
```

ANN

In [33]:

```
from keras.models import Sequential
from keras.layers import Dense, Flatten, Dropout
```

In [34]:

```
carsANN = Sequential()
```

In [35]:

```
 carsANN.add(Dense(units=8, activation= "relu"))
carsANN.add(Dense(units=3, activation= "relu"))
 carsANN.add(Dense(units=1, activation= "sigmoid"))
```

In [36]:

```
from tensorflow.keras.optimizers.schedules import ExponentialDecay
from keras.optimizers import Adam

carsANN.compile(loss='binary_crossentropy',
                  metrics=['accuracy'],
                  optimizer='Adam'
                  )
```

In [37]:

```
carsANN.fit(x_train,y_train, batch_size=40, epochs=50)
```

```
Epoch 1/50
8/8 [==============================] - 2s 7ms/step - loss: 0.5830 - accura
cy: 0.2673
Epoch 2/50
8/8 [==============================] - 0s 5ms/step - loss: 0.4828 - accura
cy: 0.3176
Epoch 3/50
8/8 [==============================] - 0s 3ms/step - loss: 0.3862 - accura
cy: 0.3522
Epoch 4/50
8/8 [==============================] - 0s 3ms/step - loss: 0.2855 - accura
cy: 0.3742
Epoch 5/50
8/8 [==============================] - 0s 4ms/step - loss: 0.1922 - accura
cy: 0.3868
Epoch 6/50
8/8 [==============================] - 0s 5ms/step - loss: 0.0983 - accura
cy: 0.4151
Epoch 7/50
8/8 [==============================] - 0s 4ms/step - loss: 0.0087 - accura
cy: 0.4843
Epoch 8/50
8/8 [==============================] - 0s 3ms/step - loss: -0.0787 - accur
acy: 0.5472
Epoch 9/50
8/8 [==============================] - 0s 5ms/step - loss: -0.1664 - accur
acy: 0.6069
Epoch 10/50
8/8 [==============================] - 0s 4ms/step - loss: -0.2501 - accur
acy: 0.6258
Epoch 11/50
8/8 [==============================] - 0s 4ms/step - loss: -0.3402 - accur
acy: 0.6289
Epoch 12/50
8/8 [==============================] - 0s 4ms/step - loss: -0.4393 - accur
acy: 0.6289
Epoch 13/50
8/8 [==============================] - 0s 5ms/step - loss: -0.5443 - accur
acy: 0.6289
Epoch 14/50
8/8 [==============================] - 0s 5ms/step - loss: -0.6489 - accur
acy: 0.6289
Epoch 15/50
8/8 [==============================] - 0s 4ms/step - loss: -0.7647 - accur
acy: 0.6289
Epoch 16/50
8/8 [==============================] - 0s 5ms/step - loss: -0.8836 - accur
acy: 0.6289
Epoch 17/50
8/8 [==============================] - 0s 5ms/step - loss: -1.0093 - accur
acy: 0.6289
Epoch 18/50
8/8 [==============================] - 0s 5ms/step - loss: -1.1331 - accur
acy: 0.6289
Epoch 19/50
8/8 [==============================] - 0s 4ms/step - loss: -1.2698 - accur
acy: 0.6289
Epoch 20/50
8/8 [==============================] - 0s 4ms/step - loss: -1.4019 - accur
acy: 0.6289
Epoch 21/50
```

```
8/8 [==============================] - 0s 5ms/step - loss: -1.5453 - accur
acy: 0.6289
Epoch 22/50
8/8 [==============================] - 0s 5ms/step - loss: -1.7009 - accur
acy: 0.6289
Epoch 23/50
8/8 [==============================] - 0s 6ms/step - loss: -1.8512 - accur
acy: 0.6289
Epoch 24/50
8/8 [==============================] - 0s 6ms/step - loss: -2.0151 - accur
acy: 0.6289
Epoch 25/50
8/8 [==============================] - 0s 5ms/step - loss: -2.1933 - accur
acy: 0.6289
Epoch 26/50
8/8 [==============================] - 0s 7ms/step - loss: -2.3852 - accur
acy: 0.6289
Epoch 27/50
8/8 [==============================] - 0s 5ms/step - loss: -2.5835 - accur
acy: 0.6289
Epoch 28/50
8/8 [==============================] - 0s 7ms/step - loss: -2.8123 - accur
acy: 0.6289
Epoch 29/50
8/8 [==============================] - 0s 4ms/step - loss: -3.0373 - accur
acy: 0.6289
Epoch 30/50
8/8 [==============================] - 0s 5ms/step - loss: -3.2917 - accur
acy: 0.6289
Epoch 31/50
8/8 [==============================] - 0s 7ms/step - loss: -3.5575 - accur
acy: 0.6289
Epoch 32/50
8/8 [==============================] - 0s 5ms/step - loss: -3.8469 - accur
acy: 0.6289
Epoch 33/50
8/8 [==============================] - 0s 5ms/step - loss: -4.1449 - accur
acy: 0.6289
Epoch 34/50
8/8 [==============================] - 0s 5ms/step - loss: -4.4840 - accur
acy: 0.6289
Epoch 35/50
8/8 [==============================] - 0s 4ms/step - loss: -4.8299 - accur
acy: 0.6289
Epoch 36/50
8/8 [==============================] - 0s 5ms/step - loss: -5.1848 - accur
acy: 0.6289
Epoch 37/50
8/8 [==============================] - 0s 5ms/step - loss: -5.6059 - accur
acy: 0.6289
Epoch 38/50
8/8 [==============================] - 0s 4ms/step - loss: -6.0411 - accur
acy: 0.6289
Epoch 39/50
8/8 [==============================] - 0s 7ms/step - loss: -6.4765 - accur
acy: 0.6289
Epoch 40/50
8/8 [==============================] - 0s 6ms/step - loss: -6.9852 - accur
acy: 0.6289
Epoch 41/50
8/8 [==============================] - 0s 5ms/step - loss: -7.4907 - accur
```

```
acy: 0.6289
Epoch 42/50
8/8 [==============================] - 0s 5ms/step - loss: -8.0696 - accur
acy: 0.6289
Epoch 43/50
8/8 [==============================] - 0s 4ms/step - loss: -8.6585 - accur
acy: 0.6289
Epoch 44/50
8/8 [==============================] - 0s 5ms/step - loss: -9.3187 - accur
acy: 0.6289
Epoch 45/50
8/8 [==============================] - 0s 5ms/step - loss: -9.9816 - accur
acy: 0.6289
Epoch 46/50
8/8 [==============================] - 0s 5ms/step - loss: -10.7609 - accu
racy: 0.6289
Epoch 47/50
8/8 [==============================] - 0s 4ms/step - loss: -11.5090 - accu
racy: 0.6289
Epoch 48/50
8/8 [==============================] - 0s 3ms/step - loss: -12.2879 - accu
racy: 0.6289
Epoch 49/50
8/8 [==============================] - 0s 5ms/step - loss: -13.1968 - accu
racy: 0.6289
Epoch 50/50
8/8 [==============================] - 0s 5ms/step - loss: -14.0783 - accu
racy: 0.6289
```

Out[37]:

```
<keras.callbacks.History at 0x17c6931ef70>
```

In [38]:

```
pred= carsANN.predict(x_test)
print(pred)
```

```
3/3 [==============================] - 0s 2ms/step
[[1.   ]
 [1.   ]
 [1.   ]
 [1.   ]
 [1.   ]
 [1.   ]
 [1.   ]
 [0.995]
 [1.   ]
 [1.   ]
 [1.   ]
 [1.   ]
 [1.   ]
 [1.   ]
 [0.999]
 [1.   ]
 [1.   ]
 [0.997]
 [0.999]
 [1.   ]
 [0.989]
 [0.999]
 [1.   ]
 [0.998]
 [1.   ]
 [0.999]
 [1.   ]
 [1.   ]
 [1.   ]
 [1.   ]
 [0.996]
 [0.999]
 [1.   ]
 [0.994]
 [1.   ]
 [1.   ]
 [1.   ]
 [0.997]
 [1.   ]
 [0.998]
 [1.   ]
 [1.   ]
 [0.996]
 [0.995]
 [1.   ]
 [1.   ]
 [0.999]
 [0.999]
 [1.   ]
 [0.999]
 [0.995]
 [0.991]
 [1.   ]
 [0.997]
 [1.   ]
 [1.   ]
 [0.997]
 [0.995]
 [1.   ]
 [1.   ]
 [1.   ]
 [0.996]
 [1.   ]
 [0.989]
 [1.   ]
 [0.997]
 [0.999]
 [0.995]
 [0.996]
 [1.   ]
 [0.996]
 [1.   ]
 [0.995]
 [0.997]
 [1.   ]
 [1.   ]
 [0.998]
 [0.995]
 [0.996]
 [0.996]]
```

In [42]:

```
carsANN.evaluate(x_test, y_test)
```

```
3/3 [==============================] - 0s 0s/step - loss: -14.8295 - accur
acy: 0.6125
```

Out[42]:

```
[-14.829455375671387, 0.612500011920929]
```

In [ ]:

```

```