01

What is Predictive Analytics?

Predictive analytics include variety of stattistical techniques from data mining, predictive modelling, and machine learing that analyze current and historical data to make predictions about future or otherwise unknown events

Predictive Analytics

1. Define problem
2. Data Collection
3. Data Preprocessing
4. EDA
5. Model
6. Deploy

02

Type of Predictive Analytics

Applintion of Predictive Analytics

Orgainzation , Businesses, Goverment are turing to predictive analytcs to help solve difficult problems and uncover new opportunities.

> Retail Agriculture Supply Chain E-commerce Manufacturing Health Insurance sales & marketing Sales & Marketing Government & Public sector Banking & Financial Services and more

CATARACT SCREENING

India has about 4.7 million blind people ,and about 66 percent of them lose their eyesight due to cataract. undiagnosed cataract remains a huge problem espescially in rural ares and among

low-income setttlements in urban areas,owing to the lack pf trained professionals and other resourcs. Therefore, cataract is a major public health problem in india ** **bold text The Tamil Nadu e-Governance Agency(TNeGA) has developed ePaarwai, to adddress the resouces constraints in screening a large number of people for cataract. By simply clicling a picture, the app can be used for preliminary screening of the eye

FORECASTING ENERY NEEDS

The Andhra Pradesh Transmission Corportion (APTRANSCO),for the first time in india, released the day-ahead electricity consumption, including a day-ahead electricity demand for every 15 minutes.

ADVISE FARMERS ON CROP DISEASES

Worldwide,crops are affected by various pests and diseses. most of farmlands are owned by small and marginal farmers in india who do not have access to right resources hence,they face a lot of crop damage and related challenges,this directly impacts farmers and their family and eventually economy which is directly dependent on agriculture.

Deep superivsed learing in recent years has been successfully used for pattern indentification from digital images.TNeGA has implemt a solution for two crops paddy and maize for detecting 3 major issues using deep learing -based model, which is trained based on a pre-builts knowlegde base to indentify and pests from digital images.

BREAST CANCER EARLY DETECTIN

According to WHO, 1 in every 12 women have the risk of a breast cancer.Early diagnosis is very critical to decrease mortality,rates. the current gold standard for breast cancer screening, mammmagraphy, requries high capital cost for equipment and experienced radiographers. it is recommmended once every 2 years and only to women above 45 years because it cannot indentify tumors effectively for younger women, and uses of x-ray for scanning,which cna make women more susceptible to cancer if screened multiple times.

NIRAMAI Health Analytix is a Bangalore-based tech startup, which stands for "Non-Invasive Risk Assessment with Machine Intelligence".they have developed a new cancer screening software that uses machine intelligence over thermography images to detect breast cancer at a much earlier stage than traditionl methods or self-examination.

LEAD MANAGMENT FOR INDUSIND BANK

Indusland solution to create, process and track sales leads from generation to conversion.the bank further sought to digitize the customer follow-ups,track workforce productivity, improve lead tracking, and receive real-time insights on various parameters aimed at enhancing the performance of the banks sales agensts.

AI=driven system enabling the auto-alloction of leads to various sales channels and agents. The approprite allocation of tasks and leads was implemented by a combination of linear programming and business rules. the leads were detemined by combining ML algorithms and utilizing data on 8 sourcing channels, 44 financial products, and 150 product varients

E-COMMEREC FRAUD DETECTION

Return order and other frauds are common in E-commerce. this leads to loss of revenue for companies. the use of Machine learing can help early detection and avoid the losses. There are many fraud deteectin machine lesring techniques and can be deployed successfully for prevention and avoidance. this will help the genuie customers with timely service and keep non genuine custorms out of network

04

Terminology of Predictive Analytics

MACHINE LEARNING

Machine learning algorithms build a model based on sample data, knows as training data, in order to make predictions or decisions without beging explicitly programmed to do so.

> **input data & out data= model**

SUPERVISED MACHINE LEARING

Supervised machine learing, is a subcategory of machine learing, uses labeled datasets to train algorithms to classify or predict outcomes accurately

> **y=function(Xi)**

y-Output, Dependent variable, Label, Target

X-Input, Independent variable,feature,Attribute

UN-SUPERVISION MACHINE LEAEING

Unlike supervised learing ,unsupervised learing uses unlabeled data(i.e., only X).From that data, it discovers patterns that help solve for clustering or association problems. this is particularly useful when subject matter experts are unsureof ccommon properties within a data set

SEMI-SUPERVISON MACHINE LEARING

When only part of the given input data has been labeleld it is a semi supervised learning problem.Unsupervised and semi-supervised learing can be useful alternatives when it is time-consuming and costly to gather label data for supervised learning.

05

Hands-on Predictive Analytics with Python

SUPERVISED MACHINE LEARNING: Mileage Prediction

Source:The dataset was used in the 1983 American Statistical Association Exposition.

Attribute Information

1. **mpg:continous(y)**
2. cylinders:multi-valued discrete
3. **displacment:continuous(X)**
4. **horsepower:continuous(X)**
5. **Weight:continous(X)**
6. **acceleration:continuous(X)**
7. model year:multi-valued discrete
8. origin:multi-valued discrete
9. car name:string(unique for each instance)

Import Library

```
import pandas as pd
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

Import data

```
df = pd.read_csv('https://github.com/YBI-Foundation/Dataset/raw/main/MPG.csv')
```

```
df.head()
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origi
0	18.0	8	307.0	130.0	3504	12.0	70	us
1	15.0	8	350.0	165.0	3693	11.5	70	us

```
df.nunique()
```

```
mpg             129
cylinders         5
displacement     82
horsepower       93
weight          351
acceleration     95
model_year       13
origin            3
name            305
dtype: int64
```

Data Preprocessing

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 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    int64
 5   acceleration  398 non-null    float64
 6   model_year    398 non-null    int64
 7   origin        398 non-null    object
 8   name          398 non-null    object
dtypes: float64(4), int64(3), object(2)
memory usage: 28.1+ KB
```

```
df.describe()
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	mc
count	398.000000	398.000000	398.000000	392.000000	398.000000	398.000000	3
mean	23.514573	5.454774	193.425879	104.469388	2970.424623	15.568090	
std	7.815984	1.701004	104.269838	38.491160	846.841774	2.757689	
min	9.000000	3.000000	68.000000	46.000000	1613.000000	8.000000	
25%	17.500000	4.000000	104.250000	75.000000	2223.750000	13.825000	
50%	23.000000	4.000000	148.500000	93.500000	2803.500000	15.500000	
75%	29.000000	8.000000	262.000000	126.000000	3608.000000	17.175000	
max	46.600000	8.000000	455.000000	230.000000	5140.000000	24.800000	

```
df.corr()
```

	mpg	cylinders	displacement	horsepower	weight	acceleration
mpg	1.000000	-0.775396	-0.804203	-0.778427	-0.831741	0.420289
cylinders	-0.775396	1.000000	0.950721	0.842983	0.896017	-0.505419
displacement	-0.804203	0.950721	1.000000	0.897257	0.932824	-0.543684
horsepower	-0.778427	0.842983	0.897257	1.000000	0.864538	-0.689196
weight	-0.831741	0.896017	0.932824	0.864538	1.000000	-0.417457
acceleration	0.420289	-0.505419	-0.543684	-0.689196	-0.417457	1.000000
model_year	0.579267	-0.348746	-0.370164	-0.416361	-0.306564	0.288137

Remove Missing Values

```
df = df.dropna()
```

```
df.info()
```

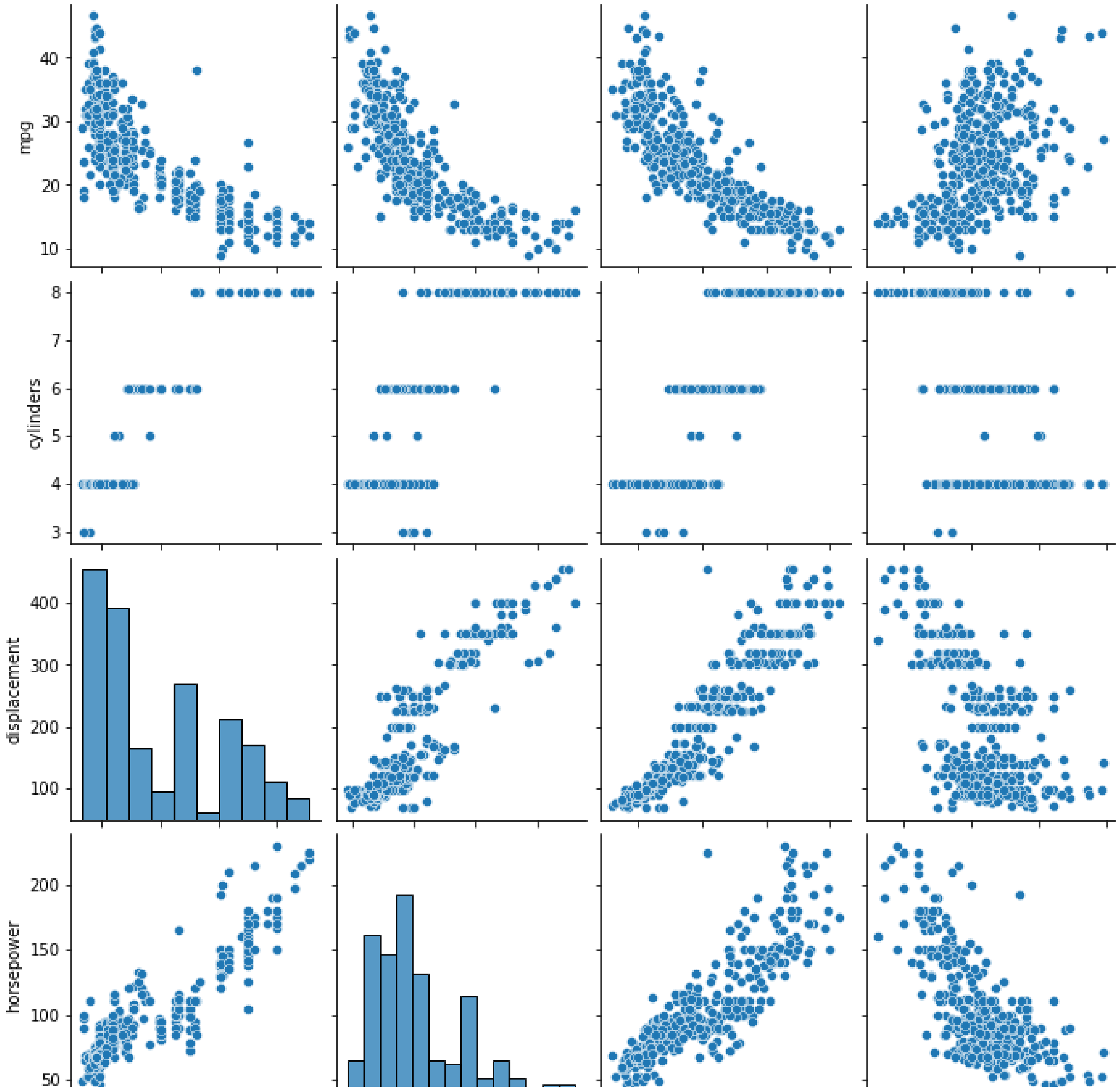
```
<class 'pandas.core.frame.DataFrame'>
```

```
Int64Index: 392 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   mpg           392 non-null    float64
 1   cylinders     392 non-null    int64  
 2   displacement  392 non-null    float64
 3   horsepower    392 non-null    float64
 4   weight        392 non-null    int64  
 5   acceleration  392 non-null    float64
 6   model_year    392 non-null    int64  
 7   origin        392 non-null    object 
 8   name          392 non-null    object 
dtypes: float64(4), int64(3), object(2)
memory usage: 30.6+ KB
```

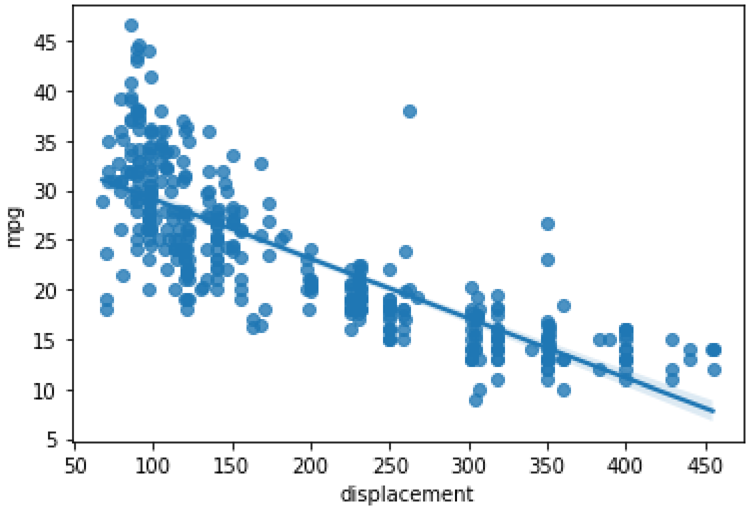
Data Visualization

```
sns.pairplot(df, x_vars= ['displacement','horsepower','weight','acceleration'])
```

```
<seaborn.axisgrid.PairGrid at 0x7ff25710c890>
```



```
sns.regplot(x = 'displacement', y = 'mpg', data = df);
```



Define Target Variable y and Feature X

```
df.columns
```

```
Index(['mpg', 'cylinders', 'displacement', 'horsepower', 'weight',
       'acceleration', 'model_year', 'origin', 'name'],
      dtype='object')
```

```
y = df['mpg']
```

```
y.shape
```

```
(392,)
```

```
X = df[['displacement', 'horsepower','weight','acceleration']]
```

```
X.shape
```

```
(392, 4)
```

```
X
```

```
array([[ 1.07728956,  0.66413273,  0.62054034, -1.285258  ],
       [ 1.48873169,  1.57459447,  0.84333403, -1.46672362],
       [ 1.1825422 ,  1.18439658,  0.54038176, -1.64818924],
       ...,
       [-0.56847897, -0.53247413, -0.80463202, -1.4304305 ],
       [-0.7120053 , -0.66254009, -0.41562716,  1.11008813],
       [-0.72157372, -0.58450051, -0.30364091,  1.40043312]])
```

Scaling Data

```
from sklearn.preprocessing import StandardScaler
```

```
ss = StandardScaler()
```

```
X = ss.fit_transform(X)
```

```
X
```

```
array([[ 1.07728956,  0.66413273,  0.62054034, -1.285258  ],
       [ 1.48873169,  1.57459447,  0.84333403, -1.46672362],
       [ 1.1825422 ,  1.18439658,  0.54038176, -1.64818924],
       ...,
       [-0.56847897, -0.53247413, -0.80463202, -1.4304305 ],
       [-0.7120053 , -0.66254009, -0.41562716,  1.11008813],
       [-0.72157372, -0.58450051, -0.30364091,  1.40043312]])
```

```
pd.DataFrame(X).describe()
```

	0	1	2	3
count	3.920000e+02	3.920000e+02	3.920000e+02	3.920000e+02
mean	-2.537653e-16	-4.392745e-16	5.607759e-17	6.117555e-16
std	1.001278e+00	1.001278e+00	1.001278e+00	1.001278e+00
min	-1.209563e+00	-1.520975e+00	-1.608575e+00	-2.736983e+00
25%	-8.555316e-01	-7.665929e-01	-8.868535e-01	-6.410551e-01
50%	-4.153842e-01	-2.853488e-01	-2.052109e-01	-1.499869e-02
75%	7.782764e-01	5.600800e-01	7.510927e-01	5.384714e-01
max	2.493416e+00	3.265452e+00	2.549061e+00	3.360262e+00

Train Test Split Data

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train ,y_test = train_test_split(X, y, train_size = 0.7)
```

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((274, 4), (118, 4), (274,), (118,))
```

Linear Regression Model

```
from sklearn.linear_model import LinearRegression
```

```
lr = LinearRegression()
```

Random Forest model

```
from sklearn.ensemble import RandomForestClassifier
```

```
rf = RandomForestClassifier()
```

Predict Test Data

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-18-8758101b10f8> in <module>()
----> 1 df.images[0]

NameError: name 'df' is not defined
```

SEARCH STACK OVERFLOW

0s completed at 3:18 AM