INTERNSHIP REPORT

A report submitted in partial fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

By

S.KISHORE

Reg.No: 953621243302

RAMCO INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

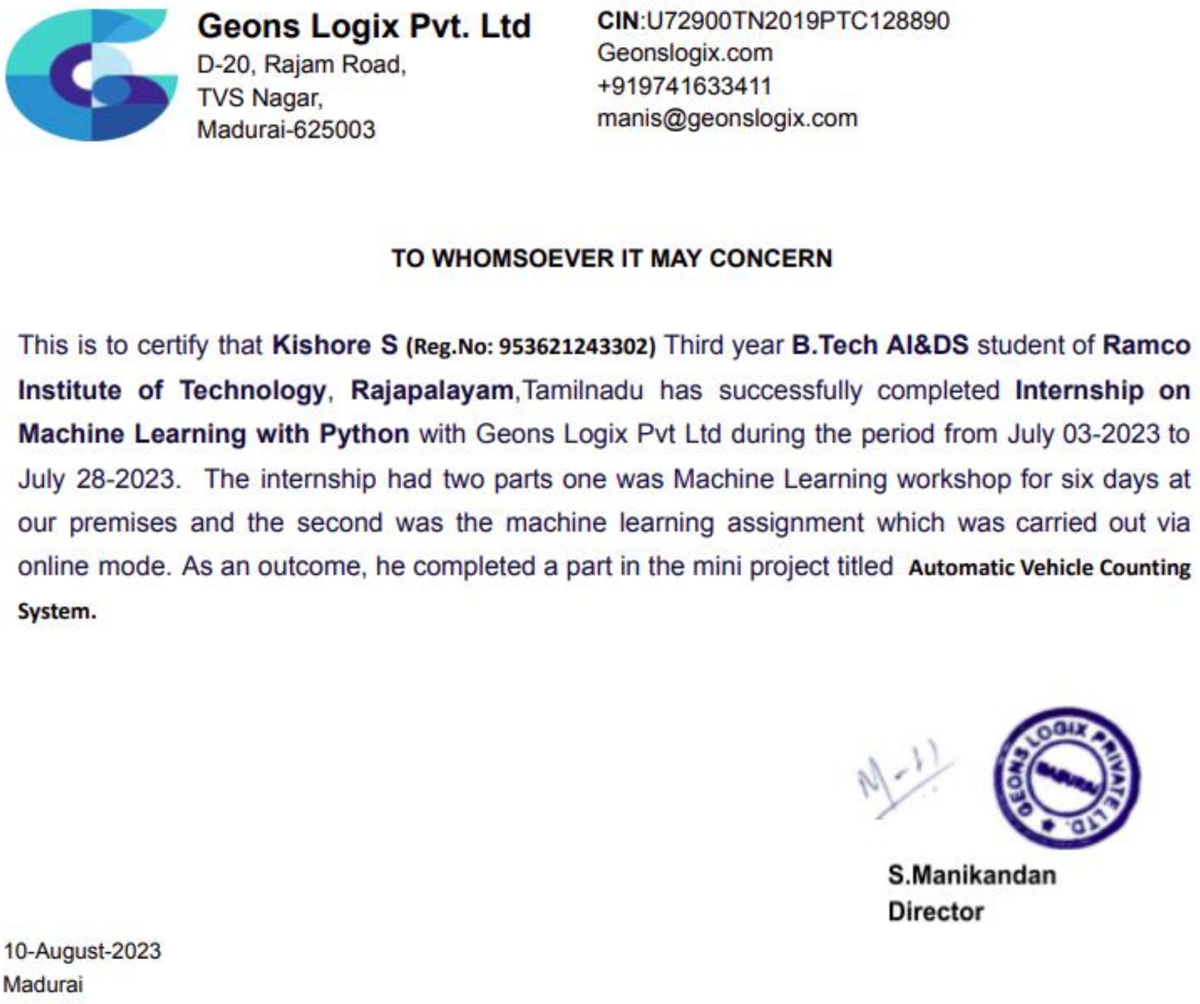
Approved by AICTE , New Delhi & Affiliated to Anna University

Accredited by NAAC & An ISO 9001:2015 Certified Instituion

North Venganallur Village , Rajapalayam – 626 117,Virudhunagar , Tamil Nadu.

2021 - 2025

Certificate



Geons Logix Pvt. Ltd
D-20, Rajam Road,
TVS Nagar,
Madurai-625003

CIN:U72900TN2019PTC128890
Geonslogix.com
+919741633411
manis@geonslogix.com

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Kishore S (Reg.No: 953621243302)** Third year **B.Tech AI&DS** student of **Ramco Institute of Technology**, **Rajapalayam**,Tamilnadu has successfully completed **Internship on Machine Learning with Python** with Geons Logix Pvt Ltd during the period from July 03-2023 to July 28-2023. The internship had two parts one was Machine Learning workshop for six days at our premises and the second was the machine learning assignment which was carried out via online mode. As an outcome, he completed a part in the mini project titled **Automatic Vehicle Counting System.**

GEONS LOGIX PRIVATE LTD.
MADURAI

S.Manikandan
Director

10-August-2023
Madurai

ACKNOWLEDGEMENT

First I would like to thank Head of V3 Analytics ,Madurai for giving me the opportunity to do an internship within the organization.

I also would like all the people that worked along with me V3 Analytics, Madurai with their patience and openness they created an enjoyable working environment.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I would like to thank my Head of the Department Dr.M.Kaliappan for his constructive criticism throughout my internship.

I would like to thank Dr. S.Vimal Associate Professor Department of AI & DS for his support and advices to get and complete internship in above said organization.

I would like to thank Ms.S.Selva Birundha and Mr. M. Ramnath Internship Coordinator for their support and advices to get and complete internship in above said organization.

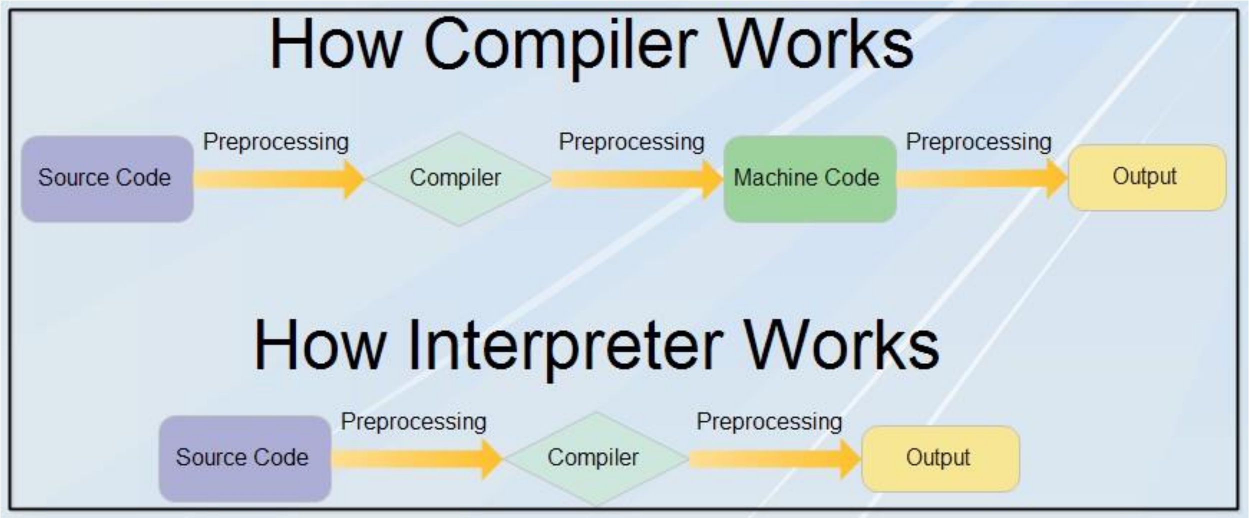
S.KISHORE
(953621243302)

Week 1
Day 1

Introduction

Compiled versus Interpreted

A compiler translates the entire source code in a single run. An interpreter translates the entire source code line by line . It consumes less time ie, it is faster than an interpreter. It consumes much more time than the compiler ie, it is slower than the compiler.

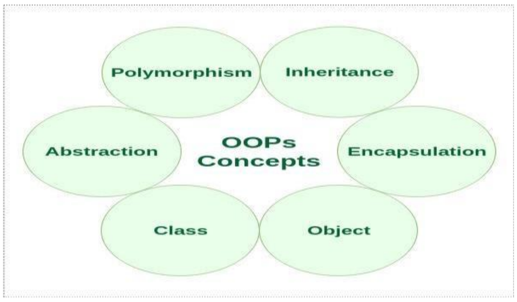


Memory

Python uses a portion of the memory for internal use and non-object memory . Another part of the memory is used for Python objects such as int, dict, list, etc.

OOPS Concept

In Python, object-oriented Programming (OOPs) is a programming paradigm that uses objects and classes in programming.



Popular Packages Used for Data Science

Numpy

NumPy is a Python library used for working with arrays . It also has functions for working in the domain of linear algebra, fourier transform, and matrices.

```
import numpy as np
```

Pandas

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive . It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python.

import pandas as pd

Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots.

import matplotlib.pyplot as plt

Scikit-learn

Scikit-learn is an open source data analysis library, and the gold standard for Machine Learning (ML) in the Python ecosystem . Key concepts and features include: Algorithmic decision-making methods, including: Classification: identifying and categorizing data based on patterns.

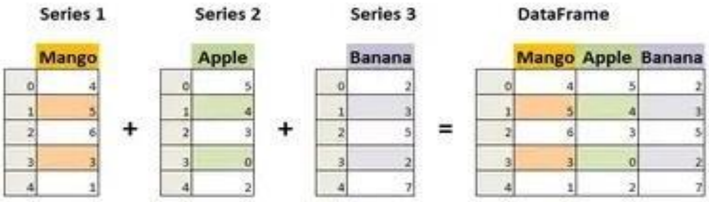
Day 2

Pandas

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive .

DataFrame and Series

A Series only has one data column, whereas a single-column DataFrame contains both a data column and an index . The labels for the data are contained in the index, which is a second column.



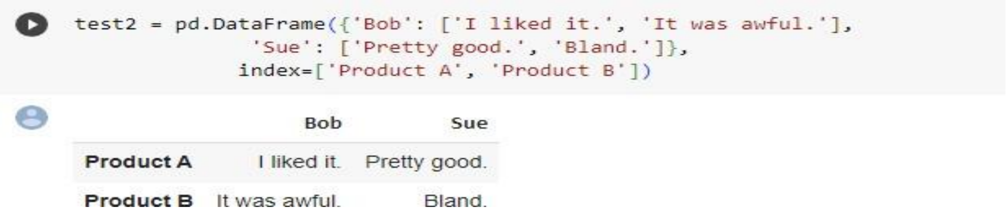
	Mango
0	4
1	5
2	6
3	3
4	1

	Apple
0	5
1	4
2	3
3	0
4	2

	Banana
0	2
1	3
2	5
3	2
4	7

	Mango	Apple	Banana
0	4	5	2
1	5	4	3
2	6	3	5
3	3	0	2
4	1	2	7

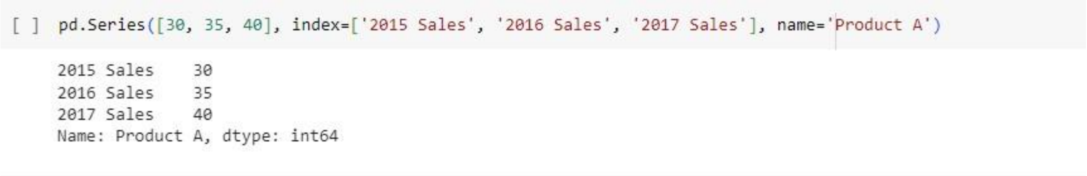
Creating dataframe:



```
test2 = pd.DataFrame({'Bob': ['I liked it.', 'It was awful.'],
                      'Sue': ['Pretty good.', 'Bland.']},
                     index=['Product A', 'Product B'])
```

	Bob	Sue
Product A	I liked it.	Pretty good.
Product B	It was awful.	Bland.

Creating series:

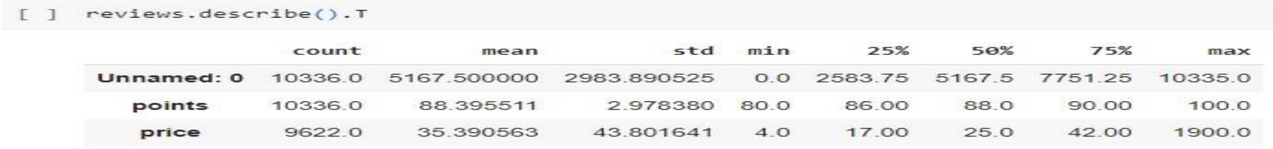


```
[ ]  pd.Series([30, 35, 40], index=['2015 Sales', '2016 Sales', '2017 Sales'], name='Product A')

     2015 Sales    30
     2016 Sales    35
     2017 Sales    40
     Name: Product A, dtype: int64
```

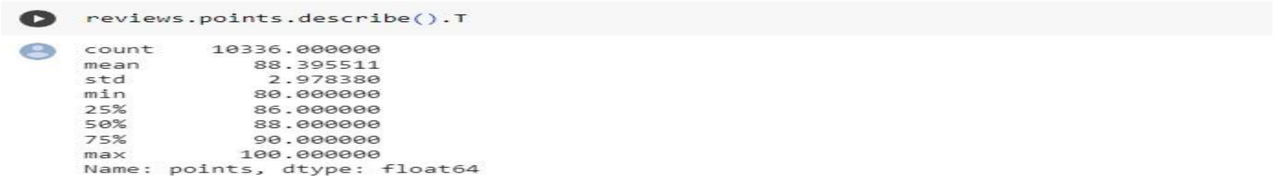
Summary Functions and Maps

Pandas provides many simple "summary functions" (not an official name) which restructure the data in some useful way.



```
[ ]  reviews.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	10336.0	5167.500000	2983.890525	0.0	2583.75	5167.5	7751.25	10335.0
points	10336.0	88.395511	2.978380	80.0	86.00	88.0	90.00	100.0
price	9622.0	35.390563	43.801641	4.0	17.00	25.0	42.00	1900.0



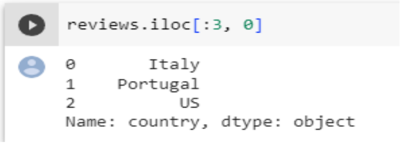
```
reviews.points.describe().T

count     10336.000000
mean         88.395511
std           2.978380
min          80.000000
25%          86.000000
50%          88.000000
75%          90.000000
max         100.000000
Name: points, dtype: float64
```

Selecting, Assigning and Indexing

The **indexing** operator and attribute selection are nice because they work just like they do in the rest of the Python ecosystem.

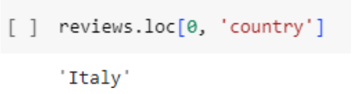
Index Based Selection:



```
reviews.iloc[:3, 0]

0       Italy
1    Portugal
2          US
Name: country, dtype: object
```

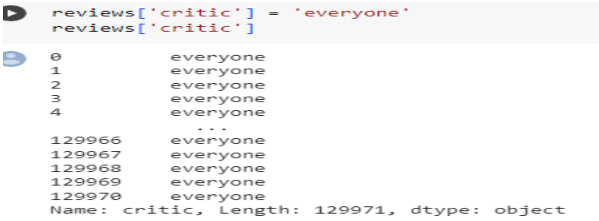
Label Based Selection:



```
[ ]  reviews.loc[0, 'country']

     'Italy'
```

Assigning Data:



```
reviews['critic'] = 'everyone'
reviews['critic']

0         everyone
1         everyone
2         everyone
3         everyone
4         everyone
            ...
129966    everyone
129967    everyone
129968    everyone
129969    everyone
129970    everyone
Name: critic, Length: 129971, dtype: object
```

Data Manipulation

Data Manipulation is the modification of information to make it easier to read or more structured .

The 9 common data manipulation techniques discussed are:

1. Filtering.

2. Sorting.
3. Grouping.
4. Pivoting.
5. Transposing.
6. Changing Data Types.
7. Adding Columns and Rows.
8. Naming Columns or Rows

Day 3

Traffic Automatic Vehicle Counting System

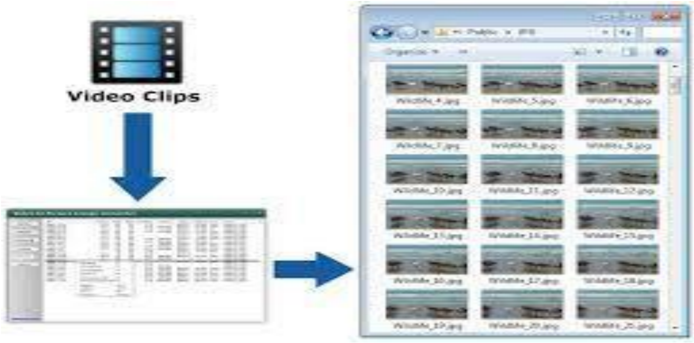
Traffic Automatic Vehicle Counting System is a complete IP Video based solution that provides high accuracy in real-time vehicle counting and classification . Snapshot capture of each vehicle that is counted and classified.

Gathering Data

Vision-based **vehicle counting** provides a highly scalable method to **count vehicles** at multiple remote locations . The Data used for the automatic vehicle counting is the real-time video.

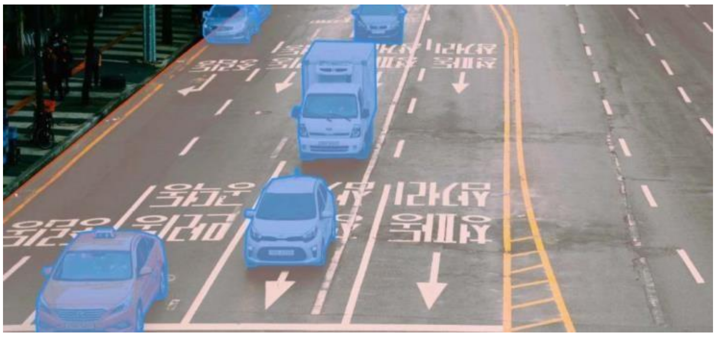
Input Frames

A frame, in a video context, is a single still image that, when played in sequence with the other frames of the video, creates motion on the playback surface .



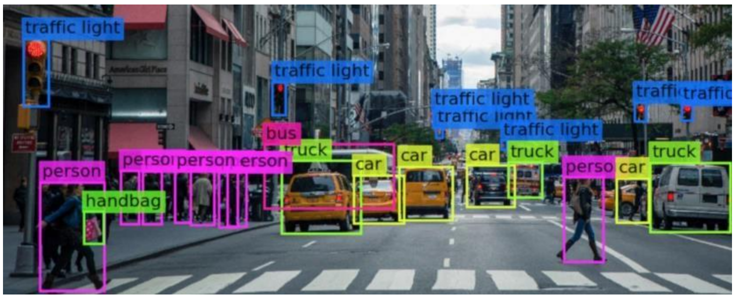
Annotation

In machine learning and deep learning, image annotation is the process of labeling or classifying an image using text, annotation tools, or both, to show the data features you want your model to recognize on its own .



Labeling and Tagging

In computer vision, the process of image labeling includes assigning particular tags to unprocessed data . The unprocessed data can be images and videos.

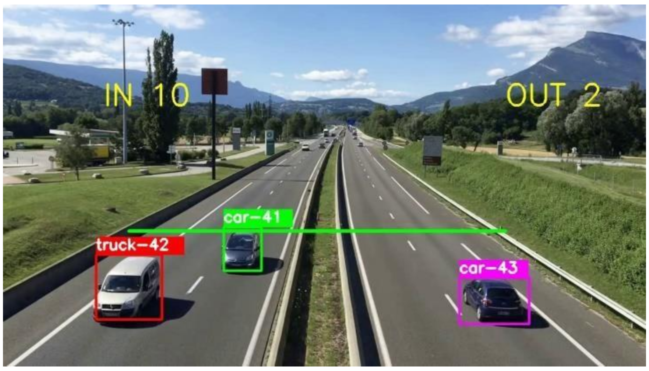


Training the model

Computer vision training data is a collection of images and labelings that are used to train a machine learning algorithm to recognize certain objects or features .

Detecting and Counting the vehicle

Vehicle detection system spots overweight vehicles moving toward overhead obstacles, such as bridges, tunnels and other structures, and it warns drivers individually.



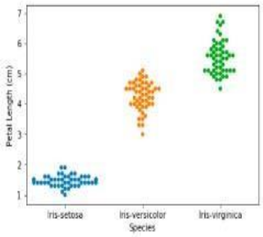
Day 4

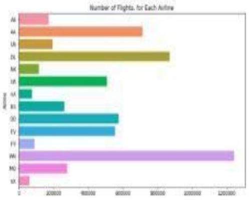
Visualization

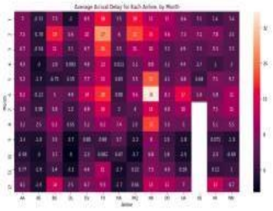
Data visualization is the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and draw insights.

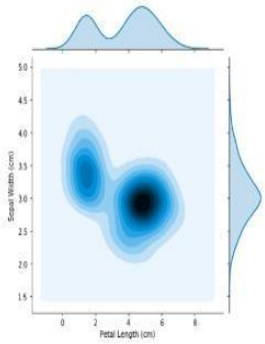
Seaborn

Seaborn is a Python data visualization library based on matplotlib . It provides a high-level interface for drawing attractive and informative statistical graphics.



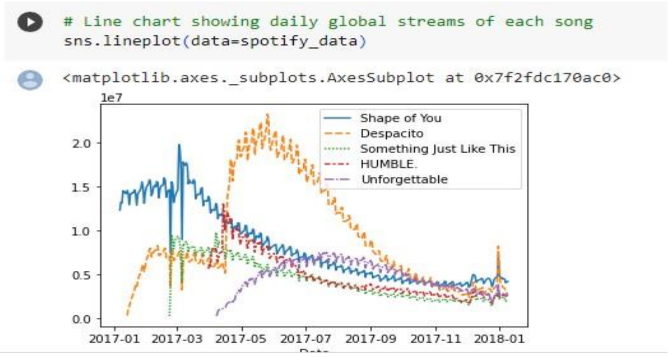






Line Plots

Line charts are used to represent the relation between two data X and Y on a different axis. Here we will see some of the examples of a line chart in Python.

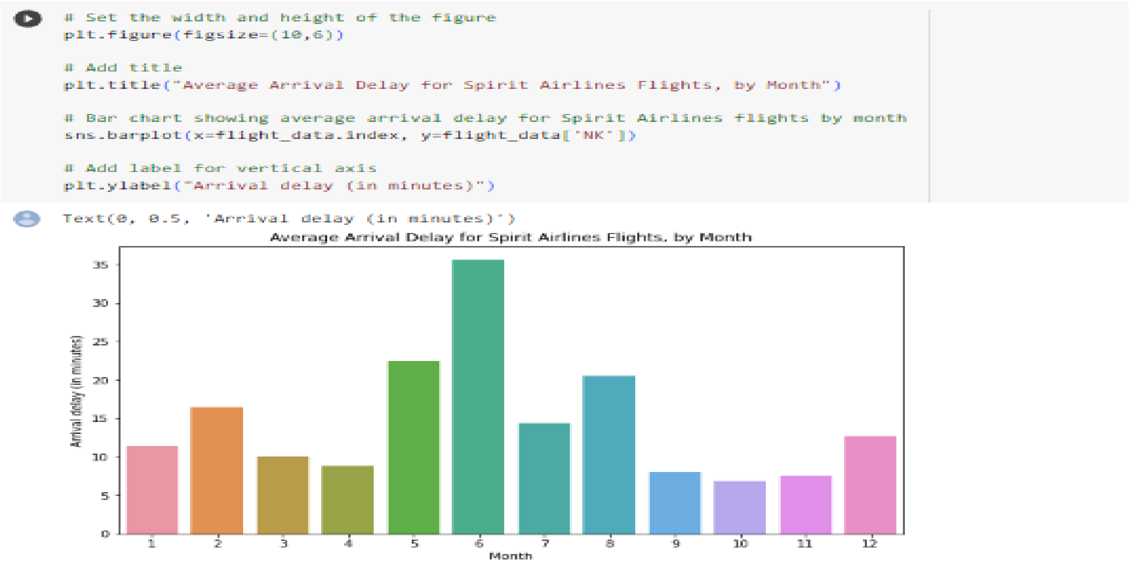


```
# Line chart showing daily global streams of each song
sns.lineplot(data=spotify_data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2fdc170ac0>
```

Bar Charts and Histogram

A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent.



```
# Set the width and height of the figure
plt.figure(figsize=(10,6))

# Add title
plt.title("Average Arrival Delay for Spirit Airlines Flights, by Month")

# Bar chart showing average arrival delay for Spirit Airlines flights by month
sns.barplot(x=flight_data.index, y=flight_data['NK'])

# Add label for vertical axis
plt.ylabel("Arrival delay (in minutes)")
```

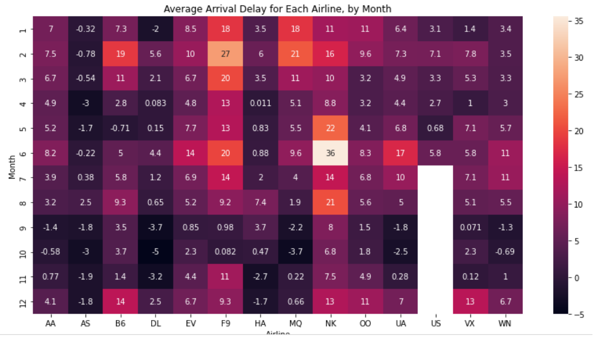
```
Text(0, 0.5, 'Arrival delay (in minutes)')
```

Average Arrival Delay for Spirit Airlines Flights, by Month
Arrival delay (in minutes)
35
30
25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
11
12
Month

Heat Maps

Heatmap is defined as a graphical representation of data using colors to visualize the value of the matrix.

```
sns.heatmap(data=flight_data, annot=True)
```

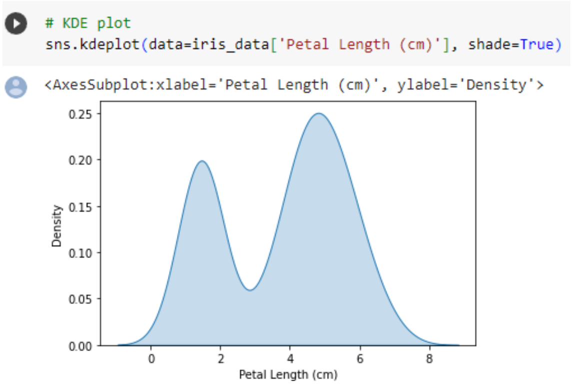


Average Arrival Delay for Each Airline, by Month

Month	AA	AS	B6	DL	EV	F9	HA	MQ	NK	OO	UA	US	VX	WN
1	7	-0.32	7.3	-2	8.5	18	3.5	18	11	11	6.4	3.1	1.4	3.4
2	7.5	-0.78	19	5.6	10	27	6	21	16	9.6	7.3	7.1	7.8	3.5
3	6.7	-0.54	11	2.1	6.7	20	3.5	11	10	3.2	4.9	3.3	5.3	3.3
4	4.9	-3	2.8	0.083	4.8	13	0.011	5.1	8.8	3.2	4.4	2.7	1	3
5	5.2	-1.7	-0.71	0.15	7.7	13	0.83	5.5	22	4.1	6.8	0.68	7.1	5.7
6	8.2	-0.22	5	4.4	14	20	0.88	9.6	36	8.3	17	5.8	5.8	11
7	3.9	0.38	5.8	1.2	6.9	14	2	4	14	6.8	10		7.1	11
8	3.2	2.5	9.3	0.65	5.2	9.2	7.4	1.9	21	5.6	5		5.1	5.5
9	-1.4	-1.8	3.5	-3.7	0.85	0.98	3.7	-2.2	8	1.5	-1.8		0.071	-1.3
10	-0.58	-3	3.7	-5	2.3	0.082	0.47	-3.7	6.8	1.8	-2.5		2.3	-0.69
11	0.77	-1.9	1.4	-3.2	4.4	11	-2.7	0.22	7.5	4.9	0.28		0.12	1
12	4.1	-1.8	14	2.5	6.7	9.3	-1.7	0.66	13	11	7		13	6.7

Distributions

Python normal distribution is a function that distributes random variables in a graph that is shaped like a symmetrical bell . It does so by arranging the probability distribution for each value.



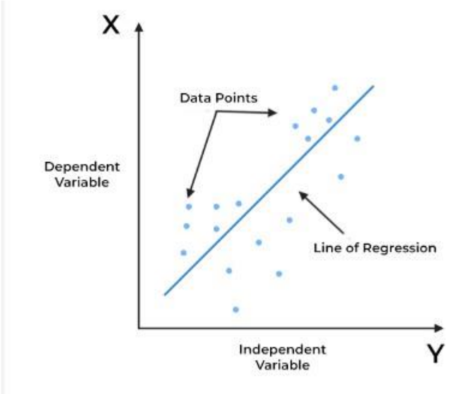
```
# KDE plot
sns.kdeplot(data=iris_data['Petal Length (cm)'], shade=True)
```

```
<AxesSubplot:xlabel='Petal Length (cm)', ylabel='Density'>
```

Day 5

Regression

Regression is a statistical method used in finance, investing, and other disciplines that attempts to determine the strength and character of the relationship between one dependent variable (usually denoted by Y) and a series of other variables (known as independent variables).



Importing packages and plotting attributes in dataframe

```
#importing libraries
from sklearn.datasets import load_boston
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
x = load_boston()
df = pd.DataFrame(x.data, columns = x.feature_names)
```

Splitting Data and Import Linear Regression

Split a dataset into train and test sets to evaluate how well our machine learning model performs. The train set is used to fit the model, and the statistics of the train set are known. The second set is called the test data set, this set is solely used for predictions.

```
 %matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.feature_selection import RFE
from sklearn.linear_model import RidgeCV, LassoCV, Ridge, Lasso
df_train, df_test = train_test_split(housing, train_size = 0.7, test_size = 0.3,
random_state = 100)
```

Metric Analysis

Evaluation metrics are a measure of how good a model performs and how well it approximates the relationship. Let us look at **MSE, MAE, R-squared, Adjusted R-squared, and RMSE.**

Model Training

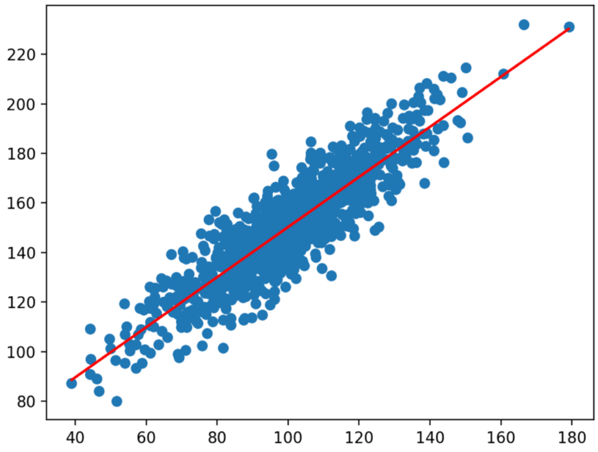
A linear regression model can be trained using the optimization algorithm gradient descent by iteratively modifying the model's parameters to reduce the mean squared error (MSE) of the model on a training dataset.

```
lm = LinearRegression()

lm.fit(X_train, y_train)
```

Test Data Prediction

- **Step 1:** Collect the data.
- **Step 2:** Fit a regression model to the data.
- **Step 3:** Verify that the model fits the data well.
- **Step 4:** Use the fitted regression equation to predict the values of new observations.



Week 2 to Week 4

VEHICLE COUNTING PROJECT

INTRODUCTION:

A vehicle counting project typically involves using computer vision techniques to automatically count the number of vehicles passing through a particular location. The projectcan be beneficial for traffic monitoring, urban planning, and data analysis.

DATA COLLECTION:

Obtain video footage or real-time camera streams from the location where you want to countvehicles. Ensure that the camera is positioned appropriately to capture the vehicles passing through the desired area.

PREPROCESSING:

Clean and enhance the video footage to improve the accuracy of the vehicle detectionalgorithms. Common pre-processing techniques include resizing and normalization.

VEHICLE DETECTION:

Implement a vehicle detection algorithm to identify vehicles in each frame of the video. YOLO a deep learning based algorithm is used for identifying the number of vehicles in eachframe.

OBJECT TRACKING:

Once vehicles are detected, use object tracking algorithms to track their movement acrossconsecutive frames. Tracking helps to maintain a continuous count of vehicles even when they temporarily disappear from the camera's view.

COUNTING AND ANALYSIS:

Keep a record of the number of vehicles detected and tracked as they pass through the REGION OF INTEREST. Additionally, you can analyze the data to obtain insights abouttraffic patterns, peak hours, or any other relevant information.

PERFORMANCE EVALUATION:

Evaluate the accuracy and performance of your vehicle counting system using appropriate metrics. Tune the system and algorithms if necessary to improve the accuracy of the countingprocess.

DEPLOYMENT:

Deploy the vehicle counting system to the desired location, either for real-time monitoring ordata collection over a period.

PACKAGES USED:

- cv2 (OpenCV)
- numpy
- pandas
- ultralytics

CODE:

COUNT:

```
import cv2
import pandas as pd
import numpy as np
from ultralytics import YOLOfrom
tracker import *
model=YOLO('best.pt')
  def RGB(event, x, y, flags, param):
      if event == cv2.EVENT_MOUSEMOVE :
          colorsBGR = [x, y]
          print(colorsBGR)
  cv2.namedWindow('RGB')
  cv2.setMouseCallback('RGB', RGB)
  cap=cv2.VideoCapture('new sample.mp4')
  my_file = open("class_list.txt", "r")data
  = my_file.read()
  class_list = data.split("\n")
  count=0
  tracker=Tracker()
  area=[(207,279),(57,339),(447,339),(467,279)]
  area_c=set()
  area1=[(555,302),(576,367),(956,367),(837,302)]
  area_c1=set()
  while True:
      ret,frame = cap.read()if
      not ret:
          break
      frame=cv2.resize(frame,(1020,500))
      results=model.predict(frame)
      a=results[0].boxes.data
      px=pd.DataFrame(a).astype("float")
      list=[]
      for index,row in px.iterrows():
          x1=int(row[0])
          y1=int(row[1])
          x2=int(row[2])
          y2=int(row[3])
          d=int(row[5])
          for d in range(len(class_list)):c =
              class_list[d]
          list.append([x1,y1,x2,y2])
      bbox_idx=tracker.update(list)for
      bbox in bbox_idx:
          x3,y3,x4,y4,id=bbox
          cx=int(x3+x4)//2
          cy=int(y3+y4)//2
          results=cv2.pointPolygonTest(np.array(area,np.int32),((x4,y4)),False)if
          results>=0:
             cv2.rectangle(frame,(x3,y3),(x4,y4),(0,255,0),2)
cv2.putText(frame,str(int(id)),(x3,y3),cv2.FONT_HERSHEY_COMPLEX,0.5,(255,0,0),1) area_c.add(id)
results1=cv2.pointPolygonTest(np.array(area1,np.int32),((x4,y4)),False)if
          results1>=0:
             cv2.rectangle(frame,(x3,y3),(x4,y4),(0,255,0),2)
  cv2.putText(frame,str(int(id)),(x3,y3),cv2.FONT_HERSHEY_COMPLEX,0.5,(255,0,0),1) area_c1.add(id)
      area1_c=(len(area_c))
      area2_c=(len(area_c1))
  cv2.putText(frame,"OUT:"+str(int(area1_c)),(66,51),cv2.FONT_HERSHEY_PLAIN,2,(0,0,0),3)
  cv2.putText(frame,"IN:"+str(int(area2_c)),(883,51),cv2.FONT_HERSHEY_PLAIN,2,(0,0,0), 3)
      a=area1_c+area2_c
print(a)
```

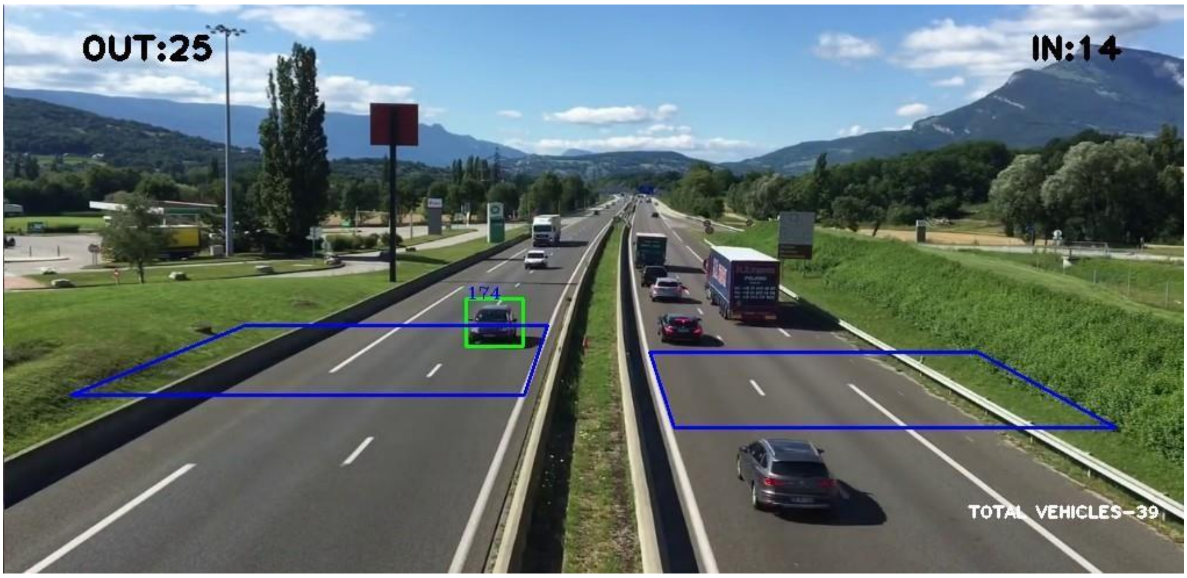
```
    cv2.putText(frame,"TOTAL
VEHICLES-"+str(int(a)),(829,445),cv2.FONT_HERSHEY_PLAIN,1,(255,255,255),2)
    cv2.polylines(frame,[np.array(area,np.int32)],True,(255,0,0),2)
   cv2.polylines(frame,[np.array(area1,np.int32)],True,(255,0,0),2)cv2.imshow("RGB", frame)
   key=cv2.waitKey(1)&0xFFif
    key==ord('z'):
        break
cap.release()
cv2.destroyAllWindows()
```

TRACKER:

```
import math
class Tracker:
def __init_(self): self.center_points = {}
        self.id_count = 0
    def update(self, objects_rect):
        objects_bbs_ids = []
        for rect in objects_rect:x,
            y, w, h = rect
            cx = (x + x + w) // 2cy
            = (y + y + h) // 2
            same_object_detected = False
            for id, pt in self.center_points.items(): dist =
                math.hypot(cx - pt[0], cy - pt[1])
                if dist < 35:
                    self.center_points[id] = (cx, cy)
                    objects_bbs_ids.append([x, y, w, h, id])
                    same_object_detected = True
                    break
            if same_object_detected is False:
                self.center_points[self.id_count] = (cx, cy)
            objects_bbs_ids.append([x, y, w, h, self.id_count])
                self.id_count += 1
        new_center_points = {}
        for obj_bb_id in objects_bbs_ids:
             _, _, _, _, object_id = obj_bb_id center =
            self.center_points[object_id]
            new_center_points[object_id] = center

        self.center_points = new_center_points.copy()return
        objects_bbs_ids
```

OUTPUT:



CONCLUSION:

The successful implementation of a vehicle counting system can provide valuable data for traffic monitoring, urban planning, and other applications. It is crucial to choose appropriatealgorithms, ensure data quality, and keep the system updated to account for changing trafficconditions and environmental factors. As technology evolves, there might be new advancements in computer vision and deep learning that can further enhance the accuracy and efficiency of vehicle counting system.