

## Zeppelin code:-

① For predicting Decupoint:-

```

① % pyspark
② import json
③ import datetime
④ % pyspark
⑤ text_file = open("filepath for decupoint.txt")
⑥ print (text_file).
⑦ % pyspark
⑧ lines = text_file.readlines()
⑨ print lines[]
⑩ % pyspark
⑪ data = json.loads(lines[])
⑫ print data.
⑬ % pyspark
⑭ for a,b in data.items():
⑮     w = a.split("T")
⑯     print w,b
⑰     x = ''.join(map(str,w))
⑱     dt = datetime.datetime.strptime
        (x, "%Y-%m-%d"
          "%H:%M:%S")
⑲ print print (dt, b).
    
```



% pyspark

```
(20) write_csv = open("filepath", 'w')
(21) for row in rdd:
(22)     for column in row:
(23)         write_csv.write('%d;' % column)
(24)     write_csv.write('\n')
(25) write_csv.close()
(26) import pandas as pd, import numpy as np.
(27) data = pd.DataFrame("filepath")
(28) date_parser = lambda date: pd.datetime.strptime(date,
(29)     plt.plot(data)  '%Y/%m/%d')
```

(11)

For predicting Humidity:-

```
(30) % pyspark.
(31) import json
(32) import date-time.
(33) % pyspark
(34) text_humidity = open("file path for humidity")
(35) % pyspark
(36) lines = text_humidity.readlines()
(37) print lines[ ]
(38) % pyspark
(39) data_humidity = json.loads (lines[ ])
(40) print data_humidity
```



```

(41) % pyspark
(42) for c, d in data_humidity.items():
(43)     row m = c.split("T")
(44)     print m, d
(45)     Y = '.'.join(map(str, m))
(46)     datetime = date.time.date.time.strptime
        (Y, "%Y-%m-%d")
        ("H:M:S")
(47)     print(datetime, d)
(48) % py spark.
(49) write_csv = open("file path for humidity", 'w')
(50) for row in 1:
(51)     for column in row:
(52)         write_csv.write('%d.%j' % (column))
(53)         write_csv.write('\n')
(54) write_csv.close()
(55) % pyspark.
(56) import pandas as pd
(57) import numpy as np
(58) import matplotlib.pyplot as plt
(59) data = pd.DataFrame("file path for csv")
(60) dateparse = lambda dates: pd.datetime.strptime(dates,
        ("%Y-%m-%d"))
(61) data = pd.read_csv("path for csv",
        parse_dates = 'Month', index_col =
        'Month', date_parser = dateparse)

```

- (62) `plt.plot(data)`
- (63) `data_log = np.log(data)`
- (64) `moveAvg = pd.rolling_mean(data_log, 12)`
- (65) `plt.plot(data_log)`
- (66) `plt.plot(moveAvg, color = "red")`



VI

For predicting Temperature:-

```
(67) % pyspark
(68) import json
(69) import datetime
(70) % pyspark
(71) text_temp = open("filepath")
(72) lines_temp = text_temp.readlines()
(73) print(lines_temp)
(74) % pyspark
(75) data_temp = json.loads(lines_temp[0])
(76) print(data_temp)
(77) % pyspark
(78) for g, h in data_temp.items():
(79)     p = g.split("T")
(80)     ("T") print(p, h)
(81)     U = datetime.datetime.strptime(
(82)         (*U, "%Y-%m-%d", "%H:%M:%S")
(83)         print(dt_temp, h)
```



```

(84) % pyspark
(85) csv_temp = open("filepath for temp", 'w')
(86) for row in i:
(87)     for column in row:
(88)         csv_temp.write('%d;' % column)
(89) csv_temp.write('\n')
(90) csv_temp.close()
(91) % pyspark
(92) import pandas as pd
(93) import numpy as np
(94) import matplotlib.pyplot as plt
(95) data = pd.DataFrame("file path for .csv")
(96) dateparse = lambda dates: pd.datetime.strptime(
    dates, '%Y-%m')
(97) data = pd.read_csv('path for .csv',
    parse_dates = 'Month', index_col =
    'Month', date_parser = dateparse)
(98) plt.plot(data)
(99) data_log = np.log(data)
(100) move-Avg = pd.rolling_mean(data_log, 12)
(101) plt.plot(data_log)
(102) plt.plot(move-Avg, color = "blue")

```



IV

For predicting wind direction

103

```
#!/usr/bin/env python
```

104

```
import json
```

105

```
import datetime
```

106

```
#!/usr/bin/env python
```

107

```
text_file = open("file_path")
```

108

```
print(text_file.read())
```

109

```
lines = text_file.readlines()
```

110

```
print lines[0]
```

111

```
#!/usr/bin/env python
```

112

```
data = json.loads(lines[0])
```

113

```
print data["wind_dir"]
```

114

```
#!/usr/bin/env python
```

115

```
for i in data["wind_dir"].items():
```

116

```
    n = re.split("T")
```

117

```
    print n, f
```

118

```
    z = ''.join(map(str, n))
```

119

```
    dt = datetime.datetime.strptime
```

```
(z, "%Y-%m-%d"
```

```
"%H:%M:%S")
```

120

```
    print(dt, data["wind_dir"], f)
```