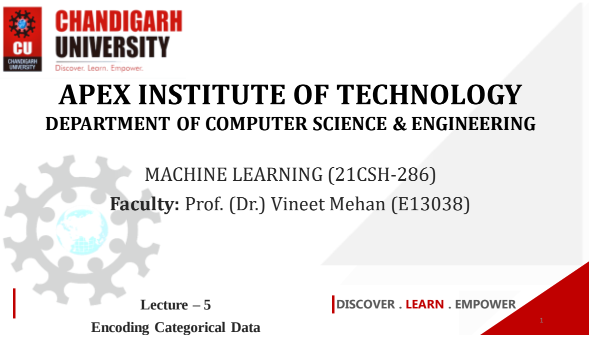


CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

APEX INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

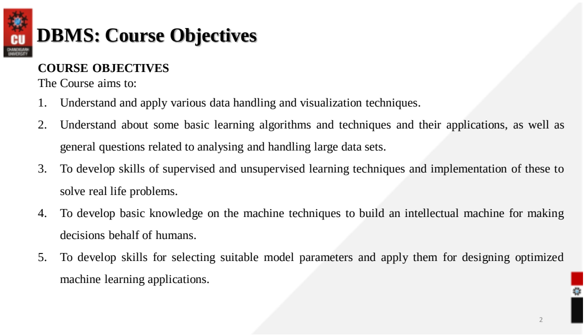
MACHINE LEARNING (21CSH-286)

Faculty: Prof. (Dr.) Vineet Mehan (E13038)

Lecture – 5

Encoding Categorical Data

DISCOVER . LEARN . EMPOWER

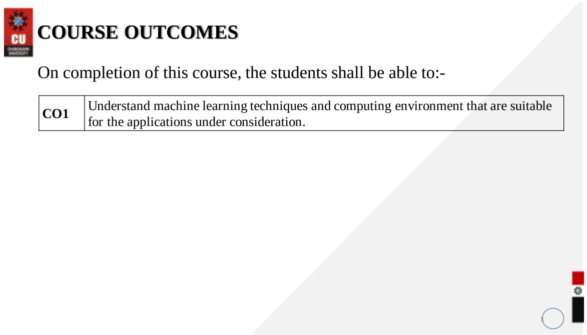


DBMS: Course Objectives

COURSE OBJECTIVES

The Course aims to:

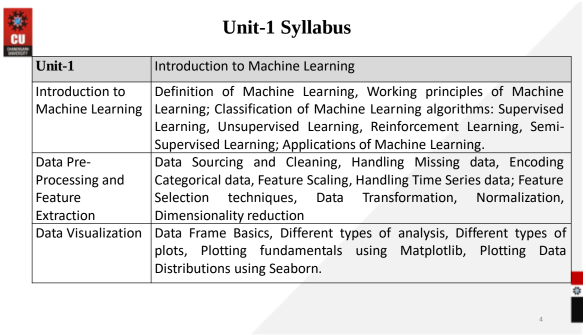
1. Understand and apply various data handling and visualization techniques.
2. Understand about some basic learning algorithms and techniques and their applications, as well as general questions related to analysing and handling large data sets.
3. To develop skills of supervised and unsupervised learning techniques and implementation of these to solve real life problems.
4. To develop basic knowledge on the machine techniques to build an intellectual machine for making decisions behalf of humans.
5. To develop skills for selecting suitable model parameters and apply them for designing optimized machine learning applications.



COURSE OUTCOMES

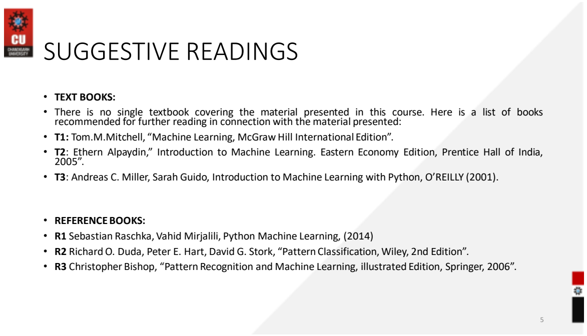
On completion of this course, the students shall be able to:-

CO1	Understand machine learning techniques and computing environment that are suitable for the applications under consideration.



Unit-1 Syllabus

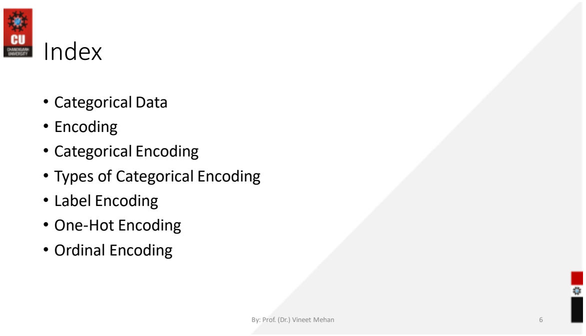
Unit-1	Introduction to Machine Learning
Introduction to Machine Learning	Definition of Machine Learning, Working principles of Machine Learning; Classification of Machine Learning algorithms: Supervised Learning, Unsupervised Learning, Reinforcement Learning, Semi-Supervised Learning; Applications of Machine Learning.
Data Pre-Processing and Feature Extraction	Data Sourcing and Cleaning, Handling Missing data, Encoding Categorical data, Feature Scaling, Handling Time Series data; Feature Selection techniques, Data Transformation, Normalization, Dimensionality reduction
Data Visualization	Data Frame Basics, Different types of analysis, Different types of plots, Plotting fundamentals using Matplotlib, Plotting Data Distributions using Seaborn.



SUGGESTIVE READINGS

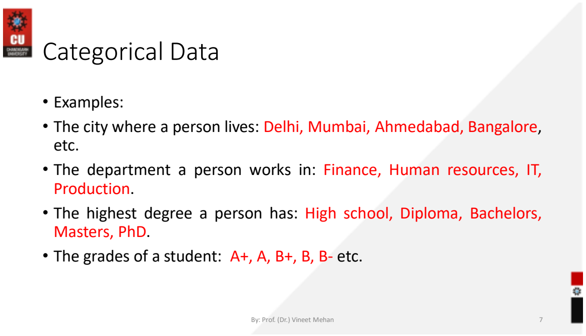
- **TEXT BOOKS:**
- There is no single textbook covering the material presented in this course. Here is a list of books recommended for further reading in connection with the material presented:
- **T1:** Tom.M.Mitchell, "Machine Learning, McGraw Hill International Edition".
- **T2**: Ethern Alpaydin," Introduction to Machine Learning. Eastern Economy Edition, Prentice Hall of India, 2005".
- **T3**: Andreas C. Miller, Sarah Guido, Introduction to Machine Learning with Python, O'REILLY (2001).

- **REFERENCE BOOKS:**
- **R1** Sebastian Raschka, Vahid Mirjalili, Python Machine Learning, (2014)
- **R2** Richard O. Duda, Peter E. Hart, David G. Stork, "Pattern Classification, Wiley, 2nd Edition".
- **R3** Christopher Bishop, "Pattern Recognition and Machine Learning, illustrated Edition, Springer, 2006".



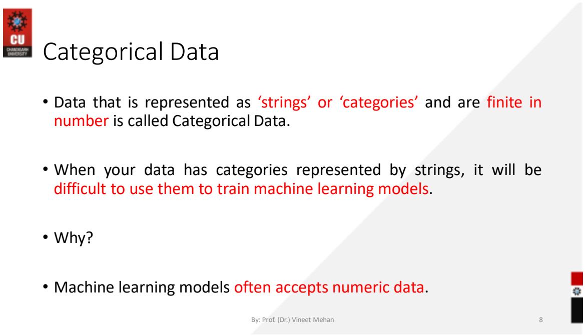
Index

- Categorical Data
- Encoding
- Categorical Encoding
- Types of Categorical Encoding
- Label Encoding
- One-Hot Encoding
- Ordinal Encoding



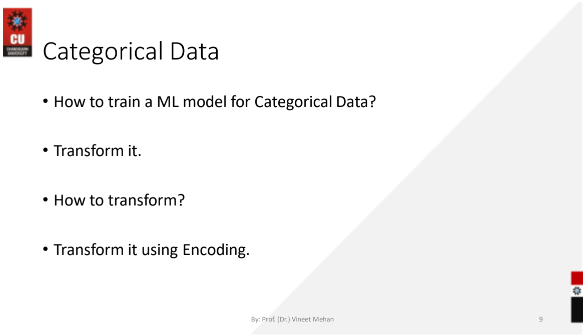
Categorical Data

- Examples:
- The city where a person lives: Delhi, Mumbai, Ahmedabad, Bangalore, etc.
- The department a person works in: Finance, Human resources, IT, Production.
- The highest degree a person has: High school, Diploma, Bachelors, Masters, PhD.
- The grades of a student: A+, A, B+, B, B- etc.



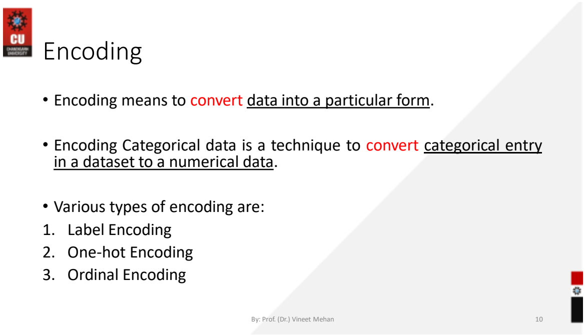
Categorical Data

- Data that is represented as 'strings' or 'categories' and are finite in number is called Categorical Data.
- When your data has categories represented by strings, it will be difficult to use them to train machine learning models.
- Why?
- Machine learning models often accepts numeric data.



Categorical Data

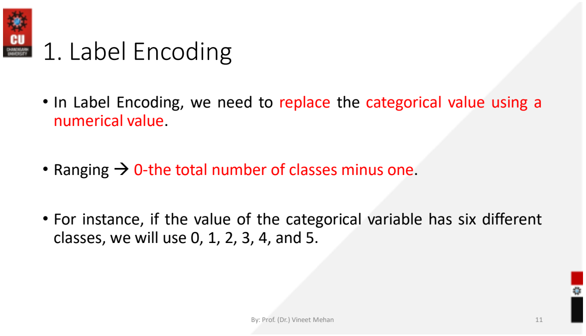
- How to train a ML model for Categorical Data?
- Transform it.
- How to transform?
- Transform it using Encoding.



Encoding

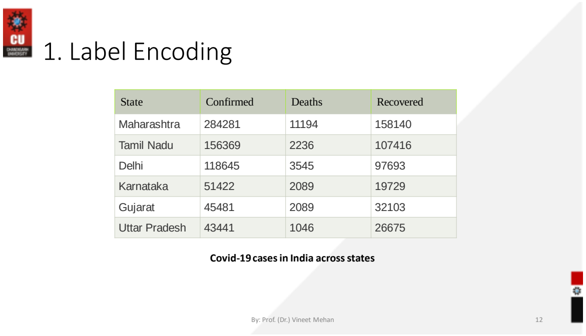
- Encoding means to convert data into a particular form.
- Encoding Categorical data is a technique to convert categorical entry in a dataset to a numerical data.
- Various types of encoding are:

1. Label Encoding
2. One-hot Encoding
3. Ordinal Encoding



1. Label Encoding

- In Label Encoding, we need to replace the categorical value using a numerical value.
- Ranging → 0-the total number of classes minus one.
- For instance, if the value of the categorical variable has six different classes, we will use 0, 1, 2, 3, 4, and 5.



1. Label Encoding

State	Confirmed	Deaths	Recovered
Maharashtra	284281	11194	158140
Tamil Nadu	156369	2236	107416
Delhi	118645	3545	97693
Karnataka	51422	2089	19729
Gujarat	45481	2089	32103
Uttar Pradesh	43441	1046	26675

Covid-19 cases in India across states

1. Label Encoding

State (Nominal Scale)	State (Label Encoding)
Maharashtra	3
Tamil Nadu	4
Delhi	0
Karnataka	2
Gujarat	1
Uttar Pradesh	5

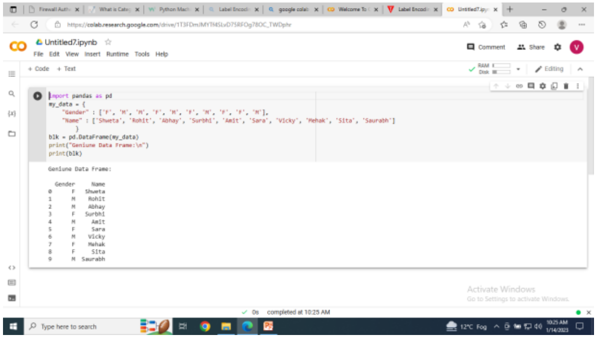
Program

We have created a dictionary 'data' and transformed it into a Data Frame with the help of the **DataFrame()** function of pandas.

- import pandas as pd
- my_data = {
-     "Gender" : ['F', 'M', 'M', 'F', 'M', 'F', 'M', 'F', 'F', 'M'],
-     "Name" : ['Shweta', 'Rohit', 'Abhay', 'Surbhi', 'Amit', 'Sara', 'Vicky', ' Mehak', 'Sita', 'Saurabh']
-         }
- blk = pd.DataFrame(my_data)
- print("Geniune Data Frame:\n")
- print(blk)

Output

```
Gender   Name
0   F  Shweta
1   M   Rohit
2   M   Abhay
3   F  Surbhi
4   M    Amit
5   F    Sara
6   M   Vicky
7   F   Mehak
8   F    Sita
9   M Saurabh
```


https://colab.research.google.com/drive/1T3FDmIMYTI4SLvD75RFOg78OC_TWDphr
Untitled7.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Comment Share
Editing
Activate Windows
Go to Settings to activate Windows.
Type here to search
12°C Fog
10:25 AM
1/14/2023

```
import pandas as pd
my_data = {
    "Gender" : ['F', 'M', 'M', 'F', 'M', 'F', 'M', 'F', 'F', 'M'],
    "Name" : ['Shweta', 'Rohit', 'Abhay', 'Surbhi', 'Amit', 'Sara', 'Vicky', 'Mehak', 'Sita', 'Saurabh']
        }
blk = pd.DataFrame(my_data)
print("Geniune Data Frame:\n")
print(blk)
```

```
Geniune Data Frame:

  Gender     Name
0      F   Shweta
1      M    Rohit
2      M    Abhay
3      F   Surbhi
4      M     Amit
5      F     Sara
6      M    Vicky
7      F    Mehak
8      F     Sita
9      M  Saurabh
```

✓ 0s completed at 10:25 AM

Program

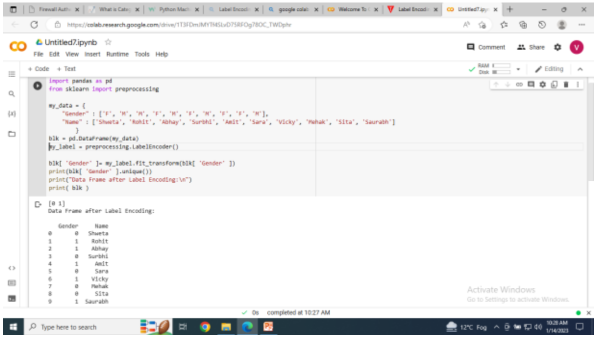
import **pandas** and **preprocessing** modules of the **scikit-learn** library.

- import pandas as pd
- from sklearn import preprocessing
- my_data = {
-     "Gender" : ['F', 'M', 'M', 'F', 'M', 'F', 'M', 'F', 'F', 'M'],
-     "Name" : ['Shweta', 'Rohit', 'Abhay', 'Surbhi', 'Amit', 'Sara', 'Vicky', 'Mehak', 'Sita', 'Saurabh']
-         }
- blk = pd.DataFrame(my_data)
- my_label = preprocessing.LabelEncoder()
-
- blk['Gender']= my_label.fit_transform(blk['Gender'])
- print(blk['Gender'].unique())
- print("Data Frame after Label Encoding:\n")
- print(blk)

fit_transform() method in order to add label encoder functionality pointed by the object to the data variable.

Output

```
 Gender   Name
0    0  Shweta
1    1   Rohit
2    1   Abhay
3    0  Surbhi
4    1    Amit
5    0    Sara
6    1   Vicky
7    0   Mehak
8    0    Sita
9    1 Saurabh
```



```
import pandas as pd
from sklearn import preprocessing

my_data = {
    "Gender" : ['F', 'M', 'M', 'F', 'M', 'F', 'M', 'F', 'F', 'M'],
    "Name" : ['Shweta', 'Rohit', 'Abhay', 'Surbhi', 'Amit', 'Sara', 'Vicky', 'Mehak', 'Sita', 'Saurabh']
        }
blk = pd.DataFrame(my_data)
my_label = preprocessing.LabelEncoder()

blk[ 'Gender' ]= my_label.fit_transform(blk[ 'Gender' ])
print(blk[ 'Gender' ].unique())
print("Data Frame after Label Encoding:\n")
print( blk )
```

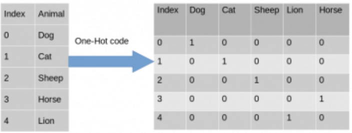
```
[0 1]
Data Frame after Label Encoding:

   Gender     Name
0       0   Shweta
1       1    Rohit
2       1    Abhay
3       0   Surbhi
4       1     Amit
5       0     Sara
6       1    Vicky
7       0    Mehak
8       0     Sita
9       1  Saurabh
```

2. One-hot Encoding

- Each category is mapped with a binary variable.

- Suppose we have a dataset with a category animal, having different animals like Dog, Cat, Sheep, Cow, Lion.



Index	Animal
0	Dog
1	Cat
2	Sheep
3	Horse
4	Lion

Index	Dog	Cat	Sheep	Lion	Horse
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	0	1
4	0	0	0	1	0

Program

- import pandas as pd

- #create DataFrame
- df = pd.DataFrame({'team': ['A', 'A', 'B', 'B', 'B', 'B', 'C', 'C'],
-                 'points': [25, 12, 15, 14, 19, 23, 25, 29]})

- #view DataFrame
- print(df)

Output

```
 team  points
0   A    25
1   A    12
2   B    15
3   B    14
4   B    19
5   B    23
6   C    25
7   C    29
```

Program

- import pandas as pd

- #create DataFrame
- df = pd.DataFrame({'team': ['A', 'A', 'B', 'B', 'B', 'B', 'C', 'C'],
-                 'points': [25, 12, 15, 14, 19, 23, 25, 29]})

- from sklearn.preprocessing import OneHotEncoder

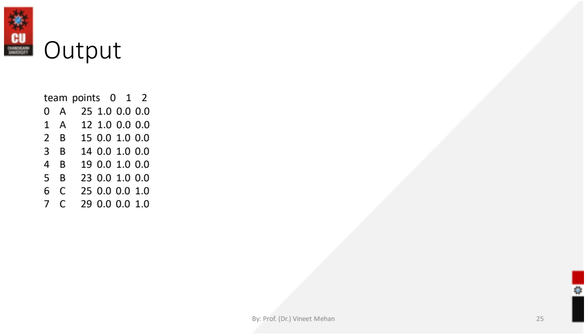
- #creating instance of one-hot-encoder
- encoder = OneHotEncoder(handle_unknown='ignore')

Program

- #perform one-hot encoding on 'team' column
- encoder_df = pd.DataFrame(encoder.fit_transform(df[['team']]).toarray())

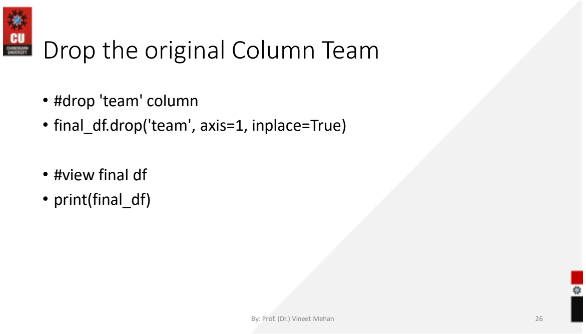
- #merge one-hot encoded columns back with original DataFrame
- final_df = df.join(encoder_df)

- #view final df
- print(final_df)



Output

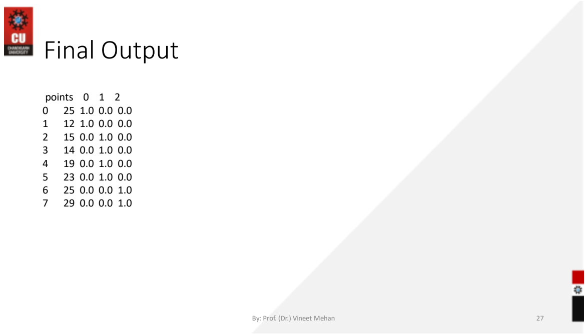
```
team points  0   1   2
0   A    25 1.0 0.0 0.0
1   A    12 1.0 0.0 0.0
2   B    15 0.0 1.0 0.0
3   B    14 0.0 1.0 0.0
4   B    19 0.0 1.0 0.0
5   B    23 0.0 1.0 0.0
6   C    25 0.0 0.0 1.0
7   C    29 0.0 0.0 1.0
```



Drop the original Column Team

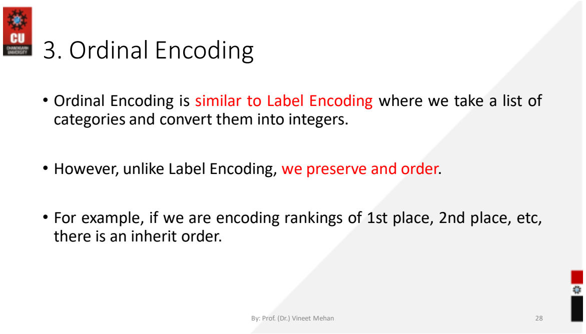
- #drop 'team' column
- final_df.drop('team', axis=1, inplace=True)

- #view final df
- print(final_df)



Final Output

```
 points  0   1   2
0    25 1.0 0.0 0.0
1    12 1.0 0.0 0.0
2    15 0.0 1.0 0.0
3    14 0.0 1.0 0.0
4    19 0.0 1.0 0.0
5    23 0.0 1.0 0.0
6    25 0.0 0.0 1.0
7    29 0.0 0.0 1.0
```

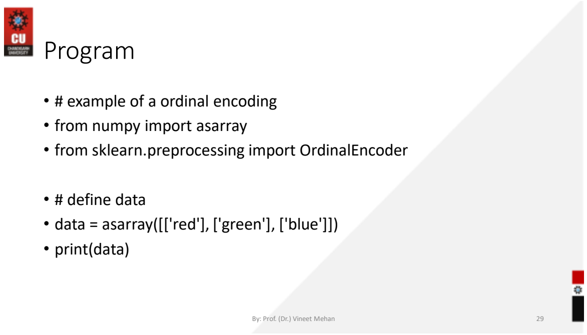


3. Ordinal Encoding

- Ordinal Encoding is similar to Label Encoding where we take a list of categories and convert them into integers.

- However, unlike Label Encoding, we preserve and order.

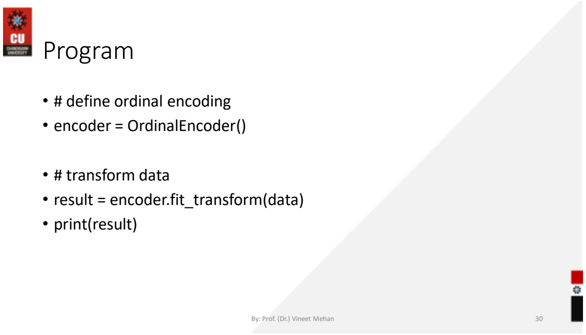
- For example, if we are encoding rankings of 1st place, 2nd place, etc, there is an inherit order.



Program

- # example of a ordinal encoding
- from numpy import asarray
- from sklearn.preprocessing import OrdinalEncoder

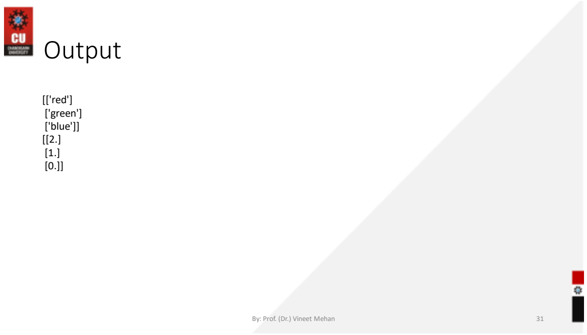
- # define data
- data = asarray([['red'], ['green'], ['blue']])
- print(data)



Program

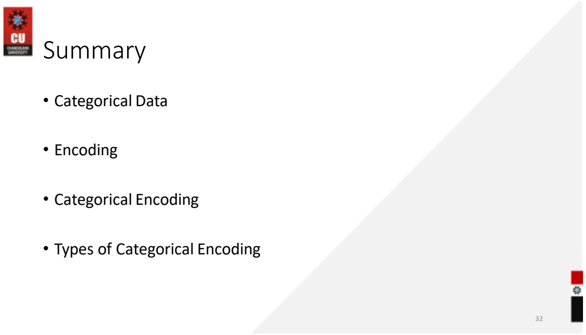
- # define ordinal encoding
- encoder = OrdinalEncoder()

- # transform data
- result = encoder.fit_transform(data)
- print(result)



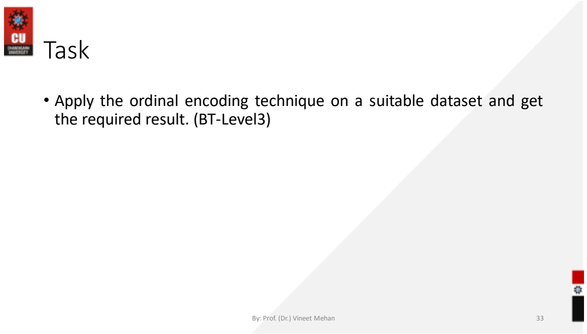
Output

```
[['red']
 ['green']
 ['blue']]
[[2.]
 [1.]
 [0.]]
```



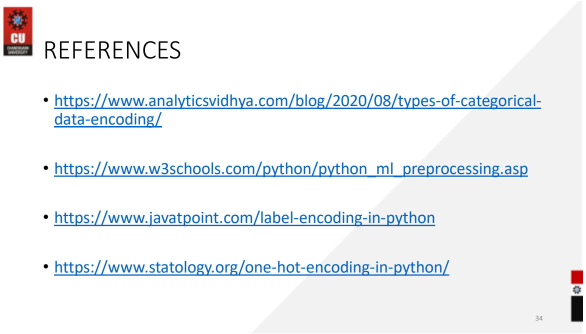
Summary

- Categorical Data
- Encoding
- Categorical Encoding
- Types of Categorical Encoding



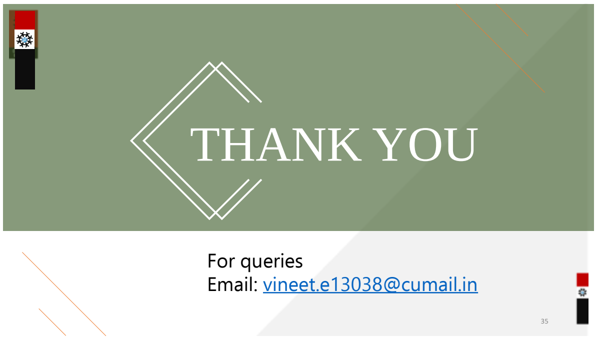
Task

- Apply the ordinal encoding technique on a suitable dataset and get the required result. (BT-Level3)



REFERENCES

- https://www.analyticsvidhya.com/blog/2020/08/types-of-categorical-data-encoding/
- https://www.w3schools.com/python/python_ml_preprocessing.asp
- https://www.javatpoint.com/label-encoding-in-python
- https://www.statology.org/one-hot-encoding-in-python/



THANK YOU

For queries
Email: vineet.e13038@cumail.in