GOVERNMENT POLYTECHNIC
NAGAMANGALA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vth Semester Diploma

Artificial Intelligence and Machine Learning
(20CS51)

Assignment: 01

NAME: Nithya
ROLL NO: 158cs22031

AIML (20CS51)
ASSIGNMENT – WEEK 02

1. Download any two datasets from the internet and perform the following operations

(a) Aggregate functions.

```
import pandas as pd

# Load dataset
df= pd.read_csv("/content/people.csv")
df.head()

# Load dataset
df= pd.read_csv("/content/Country-data.csv")
df.head()
```

OUTPUT:
Name Gender Skin Color Height(cm) Weight(kg) Date of Birth
0 Michael Male Black 175 64 1993-07-29
1 Joseph Male Black 180 132 1999-08-27
2 Matthew Male Black 162 91 1999-11-05
3 Olivia Female Brown 152 47 2003-05-14
4 Madison Female Black 160 113 1991-06-1

country child_mort exports health imports income inflation life_expec
total_fer gdpp
0 Afghanistan 90.2 10.0 7.58 44.9 1610 9.44 56.2 5.82 553

1 Albania 16.6 28.0 6.55 48.6 9930 4.49 76.3 1.65 4090
2 Algeria 27.3 38.4 4.17 31.4 12900 16.10 76.5 2.89 4460
3 Angola 119.0 62.3 2.85 42.9 5900 22.40 60.1 6.16 3530
4 Antigua and Barbuda 10.3 45.5 6.03 58.9 19100 1.44 76.8 2.13 12200

Tail():
df. tail()
df. tail()
OUTPUT:

Name Gender Skin Color Height(cm) Weight(kg) Date of Birth
15 Ashley Female White 167 50 2000-04-25
16 Isabella Female Brown 166 74 1999-09-13
17 William Male White 152 64 1993-09-16
18 Ethan Male White 160 41 1998-06-19
19 Benjamin Male Brown 170 50 1997-06

country child_mort exports health imports income inflation life_expec
total_fer gdpp
162 Vanuatu 29.2 46.6 5.25 52.7 2950 2.62 63.0 3.50 2970
163 Venezuela 17.1 28.5 4.91 17.6 16500 45.90 75.4 2.47 13500
164 Vietnam 23.3 72.0 6.84 80.2 4490 12.10 73.1 1.95 1310
165 Yemen 56.3 30.0 5.18 34.4 4480 23.60 67.5 4.67 1310
166 Zambia 83.1 37.0 5.89 30.9 3280 14.00 52.0 5.40 1460

Sum():
df. sum()
df. sum()

OUTPUT:
object
ichaelJosephMatthewOliviaMadisonEmilyAmeliaSa...
Gender MaleMaleMaleFemaleFemaleFemaleFemaleFemaleFema...
Skin Color BlackBlackBlackBrownBlackWhiteWhiteBlackBlackW...
Height(cm) 3323
Weight(kg) 1487
Date of Birth 1993-07-291999-08-271999-11-052003-05-141991-0...
dtype: objeMin

country AfghanistanAlbaniaAlgeriaAngolaAntigua and Bar...
child_mort 6391.1
exports 6865.199
health 1138.22
imports 7830.6659
income 2863163
inflation 1299.566
life_expec 11782.8
total_fer 492.31
gdpp 2165014
dtype: object

Minimum():
df. min()
df. min()
OUTPUT:
Name Alexander

Gender Female
Skin Color Black
Height(cm) 147
Weight(kg) 41
Date of Birth 1991-03-22
dtype: object

country Afghanistan
child_mort 2.6
exports 0.109
health 1.81
imports 0.0659
income 609
inflation -4.21
life_expec 32.1
total_fer 1.15
gdpp 231
dtype: object

Maximum():
df. max()
df. max()

OUTPUT:
Name William
Gender Male
Skin Color White
Height(cm) 193
Weight(kg) 132
Date of Birth 2003-05-14
dtype: object

country Zambia
child_mort 208.0
exports 200.0
health 17.9
imports 174.0
income 125000
inflation 104.0
life_expec 82.8
total_fer 7.49
gdpp 105000
dtype: object

Count():
df.count()
df. count()

OUTPUT:
Name 20
Gender 20
Skin Color 20
Height(cm) 20
Weight(kg) 20
Date of Birth 20
dtype: int64

country 167
child_mort 167
exports 167
health 167
imports 167
income 167
inflation 167
life_expec 167
total_fer 167
gdpp 167
dtype: int64

Groupby():

```
grouped=df.groupby("Name")
print(grouped.agg({"Height(cm)":"sum"}))

grouped=df.groupby("country")
print(grouped.agg({"child_mort":"sum"}))
```

OUTPUT:

Height(cm)
Name
Alexander 147
Amelia 173
Ashley 167
Benjamin 170
Daniel 160
David 193
Emily 175
Emma 157
Ethan 160
Isabella 166
James 187
Joseph 180
Madison 160
Matthew 162
Michael 175
Olivia 152
Samantha 160
Sarah 172
Sophia 155
William 152

country child_mort
Afghanistan 90.2
Albania 16.6
Algeria 27.3
Angola 119.0
Antigua and Barbuda 10.3
... ...
Vanuatu 29.2
Venezuela 17.1
Vietnam 23.3
Yemen 56.3
Zambia 83.1

[167 rows x 1 colframes

(b)Use Map, Filter, Reduce, and Lambda Functions with Pandas data frames

1.

```
import pandas as pd
data= pd.read_csv('/content/people.csv')
data['Weight(kg)'] = data['Weight(kg)'].map(lambda x: x * 2)
print(data)
filtered_df = df[df['Height(cm)'].map(lambda x: x>=60)]
print(filtered_df)
from functools import reduce
total = reduce(lambda x, y: x + y, filtered_df['Weight(kg)'])
print(total)
```

OUTPUT:

Name Gender Skin Color Height(cm) Weight(kg) Date of Birth
0 Michael Male Black 175 128 1993-07-29
1 Joseph Male Black 180 264 1999-08-27
2 Matthew Male Black 162 182 1999-11-05
3 Olivia Female Brown 152 94 2003-05-14
4 Madison Female Black 160 226 1991-06-13
5 Emily Female White 175 208 1992-02-29
6 Amelia Female White 173 192 1992-12-31
7 Sarah Female Black 172 90 1996-10-19
8 Sophia Female Black 155 138 2000-08-07
9 Emma Female White 157 120 1991-03-22
10 Alexander Male Black 147 90 2001-12-29
11 Daniel Male Brown 160 128 2001-07-10
12 Samantha Female Brown 160 82 1991-10-04
13 James Male White 187 236 1995-12-12
14 David Male Brown 193 238 1996-11-24
15 Ashley Female White 167 100 2000-04-25
16 Isabella Female Brown 166 148 1999-09-13
17 William Male White 152 128 1993-09-16
18 Ethan Male White 160 82 1998-06-19
19 Benjamin Male Brown

Name Gender Skin Color Height(cm) Weight(kg) Date of Birth
0 Michael Male Black 175 64 1993-07-29
1 Joseph Male Black 180 132 1999-08-27
2 Matthew Male Black 162 91 1999-11-05
3 Olivia Female Brown 152 47 2003-05-14
4 Madison Female Black 160 113 1991-06-13
5 Emily Female White 175 104 1992-02-29
6 Amelia Female White 173 96 1992-12-31
7 Sarah Female Black 172 45 1996-10-19
8 Sophia Female Black 155 69 2000-08-07
9 Emma Female White 157 60 1991-03-22
10 Alexander Male Black 147 45 2001-12-29
11 Daniel Male Brown 160 64 2001-07-10
12 Samantha Female Brown 160 41 1991-10-04
13 James Male White 187 118 1995-12-12
14 David Male Brown 193 119 1996-11-24
15 Ashley Female White 167 50 2000-04-25
16 Isabella Female Brown 166 74 1999-09-13

17 William Male White 152 64 1993-09-16
18 Ethan Male White 160 41 1998-06-19
19 Benjamin Male Brown 170

1487
2.

```
import pandas as pd
# read CSV file into Dataframe
d = pd.read_csv('/content/Country-data.csv')
d['child_mort '] = d['child_mort'].map(lambda x:x*1)
print(d)
filtered_d = d[d['health'].map(lambda x: x <30)]
print(filtered_d)
from functools import reduce
if not filtered_d['country'].empty: # Check if 'Country ' column is empty
  total = reduce(lambda x,y :x+y, filtered_d['inflation'])
  print(total)
```

OUTPUT:

```
country child_mort exports health imports income \
0 Afghanistan 90.2 10.0 7.58 44.9 1610
1 Albania 16.6 28.0 6.55 48.6 9930
2 Algeria 27.3 38.4 4.17 31.4 12900
3 Angola 119.0 62.3 2.85 42.9 5900
4 Antigua and Barbuda 10.3 45.5 6.03 58.9 19100
.. ... ... ... ... ... ...
162 Vanuatu 29.2 46.6 5.25 52.7 2950
163 Venezuela 17.1 28.5 4.91 17.6 16500
164 Vietnam 23.3 72.0 6.84 80.2 4490
165 Yemen 56.3 30.0 5.18 34.4 4480
166 Zambia 83.1 37.0 5.89 30.9 3280

  inflation life_expec total_fer gdpp child_mort
0 9.44 56.2 5.82 553 90.2
1 4.49 76.3 1.65 4090 16.6
2 16.10 76.5 2.89 4460 27.3
3 22.40 60.1 6.16 3530 119.0
4 1.44 76.8 2.13 12200 10.3
.. ... ... ... ... ...
162 2.62 63.0 3.50 2970 29.2
163 45.90 75.4 2.47 13500 17.1
164 12.10 73.1 1.95 1310 23.3
165 23.60 67.5 4.67 1310 56.3
166 14.00 52.0 5.40 1460 83.1

[167 rows x 11 columns]



  country child_mort exports health imports income \
0 Afghanistan 90.2 10.0 7.58 44.9 1610
1 Albania 16.6 28.0 6.55 48.6 9930
2 Algeria 27.3 38.4 4.17 31.4 12900
3 Angola 119.0 62.3 2.85 42.9 5900
4 Antigua and Barbuda 10.3 45.5 6.03 58.9 19100
.. ... ... ... ... ... ...
162 Vanuatu 29.2 46.6 5.25 52.7 2950
```

```
163 Venezuela 17.1 28.5 4.91 17.6 16500
164 Vietnam 23.3 72.0 6.84 80.2 4490
165 Yemen 56.3 30.0 5.18 34.4 4480
166 Zambia 83.1 37.0 5.89 30.9 3280

  inflation life_expec total_fer gdpp child_mort
0 9.44 56.2 5.82 553 90.2
1 4.49 76.3 1.65 4090 16.6
2 16.10 76.5 2.89 4460 27.3
3 22.40 60.1 6.16 3530 119.0
4 1.44 76.8 2.13 12200 10.3
.. ... ... ... ... ...
162 2.62 63.0 3.50 2970 29.2
163 45.90 75.4 2.47 13500 17.1
164 12.10 73.1 1.95 1310 23.3
165 23.60 67.5 4.67 1310 56.3
166 14.00 52.0 5.40 1460 83.1

[167 rows x 11 columns]

1299.566
```

(c)Visualize the data set (At least 6 different plots).

```
import pandas as pd
# load the CSV file into DataFrame
d=pd.read_csv('/content/people.csv")
import matplotlib.pyplot as plt
import seaborn as sns
# Set plot size
plt.figure(figsize=(20,15))
```

line():

```
from matplotlib import pyplot as plt
_df_54['Height(cm)'].plot(kind='line', figsize=(8, 4), title='Height(cm)')
plt.gca().spines[['top', 'right']].
```

bar():

```
from matplotlib import pyplot as plt
import seaborn as sns
_df_5.groupby('Date of Birth').size().plot(kind='bar',
color=sns.palettes.mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].
```

scatter():

```
from matplotlib import pyplot as plt
_df_6.plot(kind='scatter', x='Height(cm)', y='Weight(kg)', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].
```

histogram():

```
from matplotlib import pyplot as plt
_df_0['Height(cm)'].plot(kind='hist', bins=20, title='Height(cm)')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

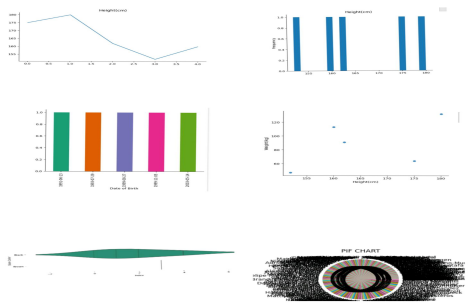
Violin():

```
from matplotlib import pyplot as plt
import seaborn as sns
figsize = (12, 1.2 * len(_df_61['Skin Color'].unique()))
plt.figure(figsize=figsize)
sns.violinplot(_df_61, x='index', y='Skin Color', inner='stick',
palette='Dark2')
sns.despine(top=True, right=True, bottom=True, left=True)
```

pie():

```
plt.subplot(2, 3, 6)
plt.pie(d['Height(cm) '], labels=d['Name'], autopct='%1.1f%%')
plt.title('PIF CHART')
```

OUTPUTS:



```
import pandas as pd
# load the CSV file into DataFrame
d=pd.read_csv('/content/Country-data.csv')
import matplotlib.pyplot as plt
import seaborn as sns
# Set plot size
plt.figure(figsize=(20,15))
```

line ():

```
from matplotlib import pyplot as plt
_df_72['child_mort'].plot(kind='line', figsize=(8, 4), title='child_mort')
plt.gca().spines[['top', 'right']].
```

bar():

```
from matplotlib import pyplot as plt
import seaborn as sns
_df_156.groupby('country').size().plot(kind='bar',
color=sns.palettes.mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].
```

histogram():

```
from matplotlib import pyplot as plt
df['child_mort'].plot(kind='hist', bins=20, title='child_mort')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

sactter():

```
from matplotlib import pyplot as plt
df.plot(kind='scatter', x='exports', y='health', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].
```

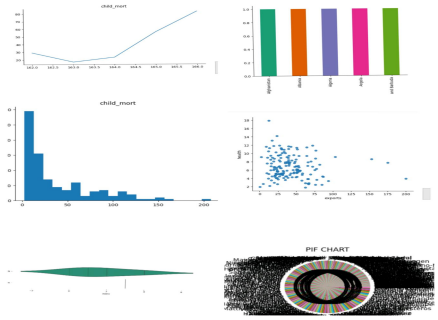
pie():

```
plt.subplot(2, 3, 6)
plt.pie(d['health'], labels=d['country'], autopct='%1.1f%%')
plt.title('PIF CHART')
```

Violin():

```
from matplotlib import pyplot as plt
import seaborn as sns
figsize = (14,0.2 * len(_df_61['country'].unique()))
plt.figure(figsize=figsize)
sns.violinplot(_df_61,x='index', y='country', inner='stick', palette='Dark2')
sns.despine(top=True, right=True, bottom=True, left=True)
```

OUTPUTS:



(d)How do you create a project plan and product backlog for an AI project? (Everyone chooses the area you want to work on or do the research work. Give a brief introduction to the work carried out and the final report to be submitted at the end of the course.)

(a)Create a Git Repository for following the Regression Project ML / deep learning.
(b)Classification Project – ML / deep learning
(c)Clustering project – ML / deep learning
(d)Natural Language Processing – ML / deep learning

How do you create a project plan and product backlog for an AI project?
(Everyone chooses the area you want to work on or do the research work. Give a brief introduction to the work carried out and the final report to be submitted at the end of the course.)
Creating a Project Plan and Product Backlog for AI Project

Project Plan and Product Backlog for AI Projects

Creating a project plan and product backlog is a critical step in the successful execution of AI projects. Below, I'll outline how to create a plan and backlog for a Natural Language Processing (NLP) project using ML/Deep Learning, with a focus on a sentiment analysis task. This framework can be adapted for regression, classification, or clustering projects.

1. **Project Plan**

A project plan defines the tasks that need to be completed, their order, and the resources required. Here's a structured plan for an NLP sentiment analysis project:

- **Week 1-2: Project Initiation and Planning**
  - **Objective Setting:** Define project goals and success metrics.
  - **Literature Review:** Gather information on best practices, models, and datasets in NLP sentiment analysis.
  - **Resource Allocation:** Identify team members and their roles, and acquire necessary tools and software.

- **Week 3-4: Data Collection and Preprocessing**
  - **Data Collection:** Gather data from social media, reviews, or other text sources.
  - **Data Cleaning:** Remove irrelevant data, correct errors, and normalize text.
  - **Data Annotation:** Label data with sentiment scores (positive, negative, neutral).

- **Week 5-6: Feature Engineering and Model Selection**
  - **Text Preprocessing:** Tokenization, stop words removal, stemming/lemmatization.
  - **Feature Extraction:** Use TF-IDF, word embeddings, or BERT embeddings.
  - **Model Selection:** Choose between ML models (e.g., SVM, Naive Bayes) and DL models (e.g., LSTM, BERT).

- **Week 7-8: Model Training and Validation**
  - **Model Training:** Train the model on the annotated dataset.
  - **Validation:** Test the model on a validation set and tune parameters.

- **Week 9-10: Model Testing and Deployment**
  - **Testing:** Evaluate the model on a test set.
  - **Deployment:** Integrate the model into a production environment.
  - **Monitoring:** Set up monitoring for model performance.

- **Week 11-12: Documentation and Presentation**
  - **Documentation:** Write detailed documentation for the model and project.
  - **Presentation:** Prepare a presentation to share findings and results.

2. **Product Backlog**

A product backlog is a prioritized list of tasks that need to be completed to achieve the project's goals. Here's a backlog for our NLP sentiment analysis project:

- **Data Collection**
- **Data Cleaning**
- **Data Annotation**
- **Text Preprocessing**
- **Feature Extraction**
- **Model Selection**
- **Model Training**
- **Validation and Tuning**
- **Testing**
- **Deployment**
- **Monitoring Setup**

- **Documentation**
- **Presentation Preparation**

(a)Create a Git Repository for following the Regression Project ML / deep learning.

Creating a Git Repository

For a regression project, for instance, to create a Git repository on GitHub, follow these steps:

1. **Create a New Repository:**
 - Log in to GitHub.
 - Click on the plus icon in the top-right corner and select "New repository."
 - Fill in the repository name and description.
 - Choose a public or private repository.
 - Initialize with a README file.

git clone https://github.com/yourusername/Regression-ML-Project.git

2. **Configure the Local Repository:**
 - Open the terminal.
 - Navigate to your local project directory.
 - Use `git init` to initialize a Git repository.
 - Use `git remote add origin <repository-url>` to link your local repository to the remote GitHub repository.

3. **Commit and Push Changes:**
 - Use `git add .` to stage all changes.
 - Use `git commit -m "Initial commit"` to commit changes.
 - Use `git push -u origin master` to push changes to GitHub.

git add .
git commit -m "Initial setup"
git push origin main

4. **Regular Workflow:**
 - Use `git add`, `git commit`, and `git push` for ongoing project development.
 - Use `git pull` to update your local repository with changes from the remote repository.

This workflow can be adapted for any AI project, including classification and clustering projects, by adjusting the tasks and repository structure as needed.

(b)Classification Project – ML / deep learning
Classification Project: ML / Deep Learning

Project Overview
A classification project using machine learning (ML) and deep learning techniques involves categorizing data into predefined classes. This could be as simple as spam detection in emails (spam vs. not spam) or as complex as identifying different species of

plants in images. For this project, we'll focus on a text classification task using a deep learning approach. Specifically, we'll classify news articles into different categories.

Project Goals
- **Data Collection:** Gather a dataset of news articles labeled with their categories.
- **Data Preprocessing:** Clean the text data, remove stop words, and convert text to a numerical format suitable for ML models.
- **Feature Engineering:** Extract text features using techniques like TF-IDF or word embeddings.
- **Model Selection:** Choose between various ML and deep learning models, such as logistic regression, support vector machines (SVM), or deep learning models like Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks.
- **Training and Evaluation:** Train the models on the preprocessed data and evaluate their performance.
- **Deployment:** Deploy the best-performing model for real-time classification.

Detailed Steps

1. **Data Collection**
    - **Source:** Use publicly available datasets like Reuters or the 20 Newsgroups dataset.
    - **Preparation:** Download the dataset and ensure it includes both the text of articles and their category labels.

2. **Data Preprocessing**
    - **Text Cleaning:** Remove special characters, numbers, and perform lowercasing.
    - **Stop Word Removal:** Eliminate common words that do not contribute much to the meaning.
    - **Tokenization:** Split text into individual words or tokens.
    - **Vectorization:** Convert text into numerical vectors.

3. **Feature Engineering**
    - **Feature Selection:** Use TF-IDF or word embeddings (Word2Vec, GloVe, or BERT) to create feature vectors.

4. **Model Selection**
    - **ML Models:** Consider logistic regression or SVM for a baseline model.
    - **Deep Learning Models:** Experiment with CNN for capturing spatial hierarchies and LSTM for understanding temporal dependencies.

5. **Training and Evaluation**
    - **Data Split:** Divide the dataset into training, validation, and test sets.
    - **Model Training:** Train the models using the training data.
    - **Evaluation:** Test the models on the test set and calculate metrics like accuracy, precision, recall, and F1-score.

6. **Deployment**
    - **Model Saving:** Save the trained model for future use.
    - **API Development:** Create a simple API for classifying new articles in real-time.
    - **Documentation:** Prepare documentation for how to use the model and its limitations.

Tools and Libraries
- **Python:** Programming language for data manipulation and model development.
- **Pandas:** Data manipulation and analysis.
- **NumPy:** Support for large, multi-dimensional arrays and matrices.
- **Scikit-learn:** Machine learning library for model selection and evaluation.

- **TensorFlow/Keras:** Frameworks for building and training deep neural networks.
- **NLTK/SpaCy:** Libraries for natural language processing.
- **Matplotlib and Seaborn:** For data visualization.

Git Repository Setup
- **Initialization:** `git init` in your project directory.
- **Committing:** Regularly commit changes with descriptive messages.
- **Branches:** Use branches for different stages of the project or for experimenting with new ideas.
- **Collaboration:** Use pull requests for integrating changes from team members.

Conclusion
This classification project using ML and deep learning will not only enhance your skills in data preprocessing, modeling, and evaluation but also provide hands-on experience in deploying a real-world AI solution. The Git repository will facilitate version control and make collaboration easier among team members.

(c)Clustering project – ML / deep learning

A clustering project in the context of Machine Learning (ML) or Deep Learning involves grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar to each other than to those in other groups. Clustering is a type of unsupervised learning used for exploratory data analysis to find inherent groupings in the data.

Project Outline

1. **Problem Definition**
 - **Objective:** Define the problem you are trying to solve. For instance, you might be clustering customer segments for marketing purposes, identifying groups of genes with similar expression patterns, or grouping news articles by topic.
 - **Data Source:** Identify the source of your data and ensure it is clean and preprocessed appropriately.

2. **Data Preparation**
 - **Data Cleaning:** Remove or impute missing values, handle outliers, and normalize the data.
 - **Feature Selection:** Choose the features that are most relevant for clustering.
 - **Dimensionality Reduction:** Use techniques like PCA (Principal Component Analysis) to reduce the dimensionality of the data if necessary.

3. **Exploratory Data Analysis**
 - **Visualize Data:** Use plots to understand the distribution and relationships between features.
 - **Statistical Analysis:** Perform statistical tests to understand the characteristics of the data.

4. **Clustering Algorithm Selection**
 - **Choose a Method:** Decide on the clustering algorithm. Common choices include K-Means, DBSCAN, hierarchical clustering, or more advanced methods such as spectral clustering or deep learning-based clustering.
 - **Parameter Tuning:** Determine the optimal parameters for your chosen algorithm. For K-Means, you might need to determine the number of clusters (K).

5. **Model Training**

- **Clustering:** Apply the chosen algorithm to cluster the data.
- **Evaluation:** Use metrics such as silhouette score, silhouette coefficient, or within-cluster sum of squares to evaluate the quality of the clustering.

6. **Result Interpretation**
- **Cluster Profiles:** Analyze the characteristics of each cluster to understand their composition and differences.
- **Visualization:** Use plots to visualize the clusters and their relationships.

7. **Post-Processing and Analysis**
- **Further Analysis:** Conduct additional analyses, such as ANOVA or t-tests, to compare the means of different features across clusters.
- **Business Impact:** Discuss the implications of the clusters for your specific business problem or research question.

8. **Reporting and Documentation**
- **Write a Report:** Document your methodology, results, and conclusions in a detailed report.
- **Presentation:** Prepare a presentation to clearly communicate your findings to stakeholders.

9. **Implementation**
- **Deployment:** If applicable, implement the clustering results in a real-world application, such as a recommendation system or customer segmentation tool.

Project Plan

1. **Week 1-2: Problem Definition and Data Preparation**
- Define the problem and collect data.
- Clean and preprocess the data.
- Perform initial exploratory data analysis.

2. **Week 3-4: Exploratory Data Analysis and Algorithm Selection**
- Complete data analysis and feature selection.
- Choose and configure the clustering algorithm.

3. **Week 5-6: Model Training and Evaluation**
- Train the model.
- Evaluate the clustering results.

4. **Week 7-8: Result Interpretation and Post-Processing**
- Interpret the cluster profiles.
- Conduct additional analyses.

5. **Week 9-10: Reporting, Documentation, and Presentation**
- Write the final report.
- Prepare and deliver a presentation.

6. **Week 11-12: Implementation (if applicable)**
- Implement the clustering results in a real application.
- Monitor and adjust as necessary.

Product Backlog

- **Data Collection and Cleaning**
- **Feature Selection**
- **Dimensionality Reduction**
- **Exploratory Data Analysis**
- **Model Selection and Tuning**
- **Model Training**
- **Clustering Evaluation**
- **Cluster Interpretation**
- **Additional Statistical Analysis**
- **Report Writing**
- **Presentation Preparation**
- **Implementation (if applicable)**
- **Monitoring (if applicable)**

By following this outline and project plan, you can effectively execute a clustering project for ML or deep learning, ensuring thorough analysis and clear communication of results.

(d)Natural Language Processing – ML / deep learning

Natural Language Processing (NLP) projects in the context of Machine Learning (ML) and Deep Learning involve teaching computers to understand, interpret, and generate human language. This field encompasses a wide range of tasks, from sentiment analysis and machine translation to question answering and text summarization. Here's a structured approach to planning and executing an NLP project using ML and deep learning techniques:

Project Outline

1. **Problem Definition**
 - **Objective:** Define the NLP task you want to accomplish. Is it sentiment analysis, text classification, named entity recognition, text generation, or something else?
 - **Scope:** Specify the domain and the language(s) the model will handle.

2. **Data Collection**
 - **Data Source:** Identify where you will get the text data. This could be from online sources, books, articles, or databases.
 - **Data Annotation:** If necessary, create or acquire annotated data for supervised learning tasks.

3. **Data Preprocessing**
 - **Text Cleaning:** Remove noise and irrelevant data.
 - **Tokenization:** Break text into words, phrases, and sentences.
 - **Normalization:** Convert text to lowercase, remove punctuation, and handle contractions.
 - **Stop Words Removal:** Remove common words that don't add meaning.
 - **Stemming or Lemmatization:** Reduce words to their root form.

4. **Feature Engineering**
 - **Word Embeddings:** Use techniques like Word2Vec, GloVe, or FastText to convert words into numerical vectors.
 - **Sequence Modeling:** Prepare data for sequence models like RNNs (Recurrent Neural Networks) and LSTMs (Long Short-Term Memory units).
 - **Transformers Preprocessing:** If using transformers like BERT or GPT, preprocess data to fit their requirements.

5. **Model Selection and Training**
 - **Choose a Model:** Select between traditional ML models (like Naive Bayes for text classification) or deep learning models (like RNNs, LSTMs, or transformers).
 - **Model Training:** Train the model on your annotated dataset.
 - **Evaluation:** Use a validation set to fine-tune model parameters and evaluate performance using metrics like accuracy, precision, recall, or F1 score.

6. **Post-Processing**
 - **Model Tuning:** Adjust model parameters based on validation results.
 - **Error Analysis:** Analyze errors to understand model weaknesses.

7. **Deployment**
 - **Integration:** Integrate the model into a larger system or application.
 - **Testing:** Conduct thorough testing in a production environment.
 - **Monitoring:** Set up monitoring to track model performance over time.

8. **Documentation and Reporting**
 - **Documentation:** Write detailed documentation for the model, including preprocessing steps and model parameters.
 - **Report:** Prepare a report that includes the methodology, results, and any limitations of the model.

Product Backlog

- **Literature Review:** Understand current research and models in your NLP task.
- **Data Collection:** Gather and clean data.
- **Data Annotation:** Annotate data for supervised learning (if necessary).
- **Text Cleaning and Tokenization:** Clean and tokenize the text data.
- **Feature Engineering:** Prepare text data for model input.
- **Model Selection:** Choose between ML and deep learning models.
- **Model Training:** Train the selected model.
- **Model Evaluation:** Evaluate model performance on a validation set.
- **Model Tuning:** Adjust model parameters as needed.
- **Integration and Testing:** Integrate the model into a larger system and conduct testing.
- **Monitoring:** Set up monitoring for model performance.
- **Documentation and Report Writing:** Document the project and write a final report.

Project Plan

1. **Weeks 1-2: Data Collection and Preprocessing**
 - Collect and preprocess data.

2. **Weeks 3-4: Feature Engineering and Model Selection**
 - Perform feature engineering.
 - Select and configure the model.

3. **Weeks 5-7: Model Training and Evaluation**
 - Train the model.
 - Evaluate the model on the validation set.

4. **Weeks 8-9: Model Tuning and Post-Processing**
 - Tune model parameters.

- Analyze model errors.

5. **Weeks 10-12: Deployment and Testing**
- Integrate the model into an application.
- Conduct thorough testing.

6. **Weeks 13-14: Documentation and Reporting**
- Write detailed documentation.
- Prepare a final report.

By following this plan and backlog, you can structure an NLP project that utilizes ML and deep learning to achieve your goals while ensuring that all critical steps are covered.