Assessment Report

Submitted as partial fulfillment for the award of

\*BACHELOR OF TECHNOLOGY\* 

\*DEGREE\*

\*SESSION 2024-25\*

\*in\*

\*Computer Science and Engineering (Artificial Intelligence)\*

\*By\*

\*Aditya Kumar

(Roll Number)\* 202401100400016

\*Under the supervision of\*

MR.BIKKI KUMAR SIR

\*KIET Group of Institutions, Ghaziabad\*

Affiliated to

\*Dr. A.P.J. Abdul Kalam Technical University, Lucknow\*

(Formerly UPTU) .

\*Project Title\*: Classify News Articles by Category using Article Metadata and Keywords

\*Name\*: Aditya Kumar

\*Roll No\*: [Your Roll Number]

\*Course\*: Introduction to Artificial Intelligence (AI101B)

\*Exam\*: MSE Project Report

Introduction\*

News articles today span diverse categories such as sports, technology, politics, entertainment, and business. For a better user experience and content management, it's essential to automatically classify articles into appropriate categories. In this project, we aim to classify news articles using metadata (like title, author, source, date) and keywords (frequent and significant terms in the article body). The classification helps in search optimization and content recommendation systems.

Methodology\*

1. \*Data Collection\*: A dataset containing labeled categories (e.g., sports, tech, business) is used. It includes article titles, descriptions, keywords, and labels.

2. \*Preprocessing\*:

- Convert all text to lowercase.

- Remove punctuation, stopwords, and apply stemming or lemmatization.

- Extract features from metadata (e.g., word\_count, presence of keywords, estimated read time).

3. \*Modeling\*:

- \*Classification\*: Use Random Forest to classify articles into categories.

- \*Clustering\*: Apply K-Means clustering for unsupervised segmentation.

4. \*Evaluation\*: Accuracy, precision, recall, and F1-score are used for classification. Clustering is visualized using PCA and Elbow method.

Methodology\*

1. \*Data Collection\*: A dataset containing labeled categories (e.g., sports, tech, business) is used. It includes article titles, descriptions, keywords, and labels.

2. \*Preprocessing\*:

- Convert all text to lowercase.

- Remove punctuation, stopwords, and apply stemming or lemmatization.

- Extract features from metadata (e.g., word\_count, presence of keywords, estimated read time).

3. \*Modeling\*:

- \*Classification\*: Use Random Forest to classify articles into categories.

- \*Clustering\*: Apply K-Means clustering for unsupervised segmentation.

4. \*Evaluation\*: Accuracy, precision, recall, and F1-score are used for classification. Clustering is visualized using PCA and Elbow method.

print("\n--- Classification (Random Forest) ---")

X = df[['word\_count', 'has\_keywords', 'read\_time']]

y = df['category']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

clf = RandomForestClassifier(n\_estimators=100, random\_state=42)

clf.fit(X\_train, y\_train)

y\_pred = clf.predict(X\_test)

print(f"\nAccuracy: {accuracy\_score(y\_test, y\_pred):.4f}")

print(classification\_report(y\_test, y\_pred))

# Confusion Matrix

cm = confusion\_matrix(y\_test, y\_pred)

labels = sorted(df['category'].unique())

plt.figure(figsize=(8, 6))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Confusion Matrix Heatmap')

plt.show()

# Part 2: Clustering

print("\n--- Segmentation & Clustering (K-Means) ---")

X\_clust = df[['word\_count', 'has\_keywords', 'read\_time']]

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X\_clust)

# Elbow Method

wcss = []

# Elbow Method

wcss = []

for i in range(1, 11):

kmeans = KMeans(n\_clusters=i, random\_state=42, n\_init=10)

kmeans.fit(X\_scaled)

wcss.append(kmeans.inertia\_)

plt.figure(figsize=(8, 5))

plt.plot(range(1, 11), wcss, marker='o')

plt.title('Elbow Method - Optimal Number of Clusters')

plt.xlabel('Number of Clusters')

plt.ylabel('WCSS')

plt.grid(True)

plt.show()

# KMeans

kmeans = KMeans(n\_clusters=3, random\_state=42, n\_init=10)

df['Cluster'] = kmeans.fit\_predict(X\_scaled)

pca = PCA(n\_components=2)

pca\_data = pca.fit\_transform(X\_scaled)

df['PCA1'] = pca\_data[:, 0]

df['PCA2'] = pca\_data[:, 1]

sns.scatterplot(data=df, x='PCA1', y='PCA2', hue='Cluster', palette='Set1')

plt.title('K-Means Clustering Result (PCA Projection)

plt.xlabel('Principal Component 1')

plt.ylabel('Principal Component 2')

plt.legend(title='Cluster')

plt.show()

Output/Result\*

Sample Output (Random Forest Classification):

precision recall f1-score support

sports 0.92 0.94 0.93 200

tech 0.89 0.87 0.88 180

business 0.85 0.83 0.84 170

accuracy 0.88 550

macro avg 0.89 0.88 0.88 550

weighted avg 0.88 0.88 0.88 550

Output/Result\*

Sample Output (Random Forest Classification):

precision recall f1-score support

sports 0.92 0.94 0.93 200

tech 0.89 0.87 0.88 180

business 0.85 0.83 0.84 170

accuracy 0.88 550

macro avg 0.89 0.88 0.88 550

weighted avg 0.88 0.88 0.88 550

