**Bahria University,**

**Karachi Campus**



## LAB EXPERIMENT NO.

6

## LIST OF TASKS

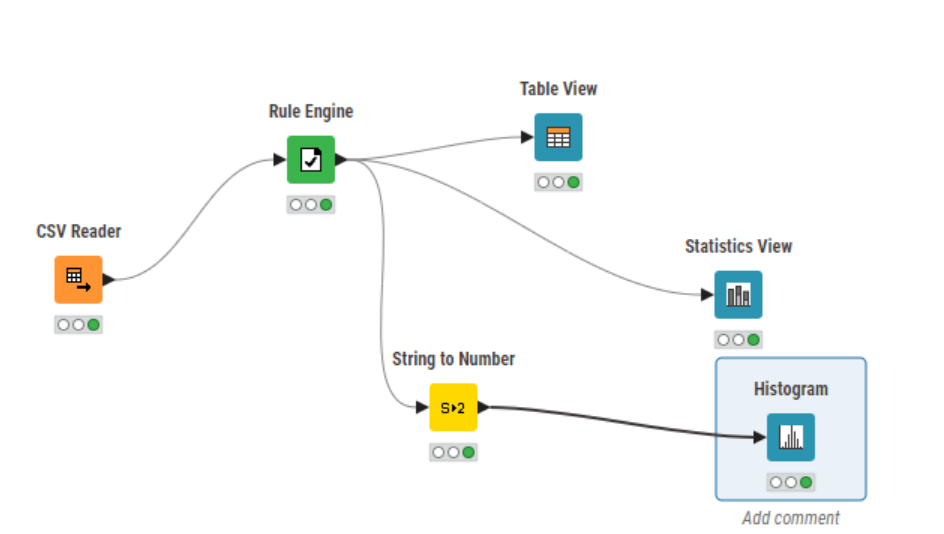
|  |  |
| --- | --- |
| **TASK NO** | **OBJECTIVE** |
| **1** | Using python implements VADER rulesbased classification algorithm to find the sentiments of different sentences. |
| **2** | :Using python implements textBlob rules-based classification algorithm to find the sentiments of different sentences and compare the results with task # 01. |

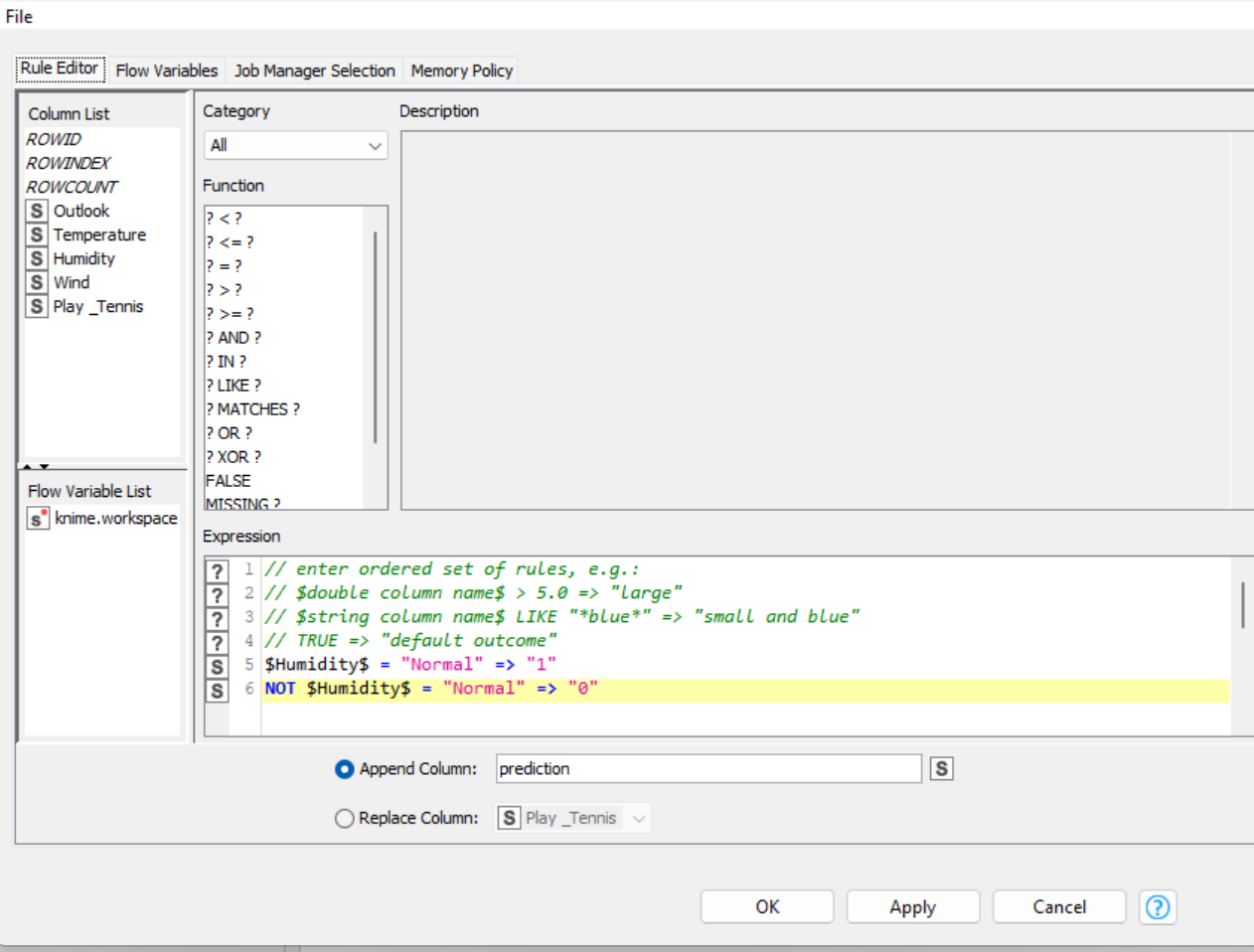
Submitted On:

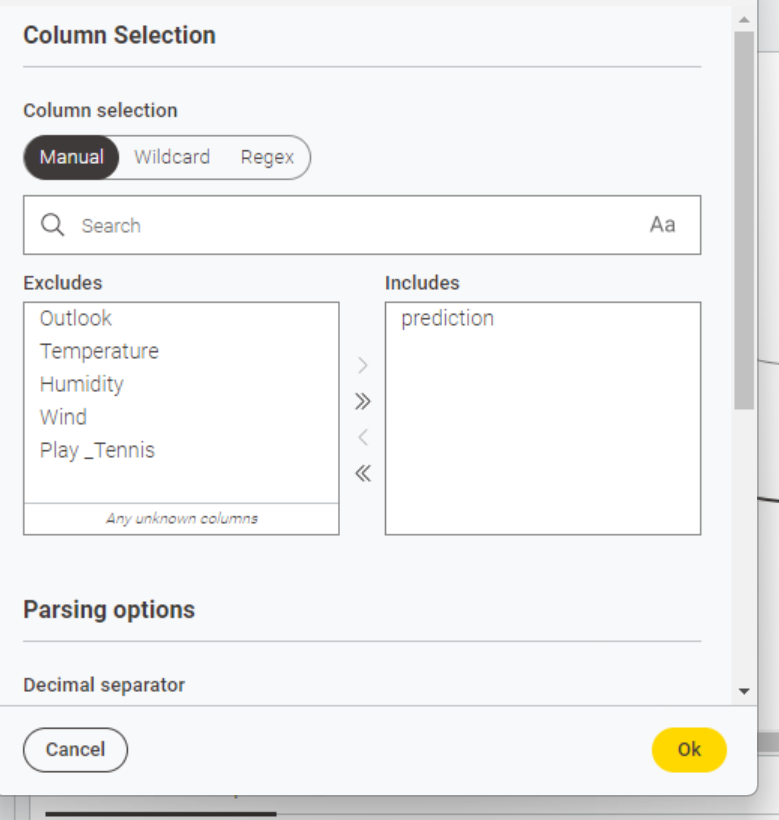
25 feb 2024

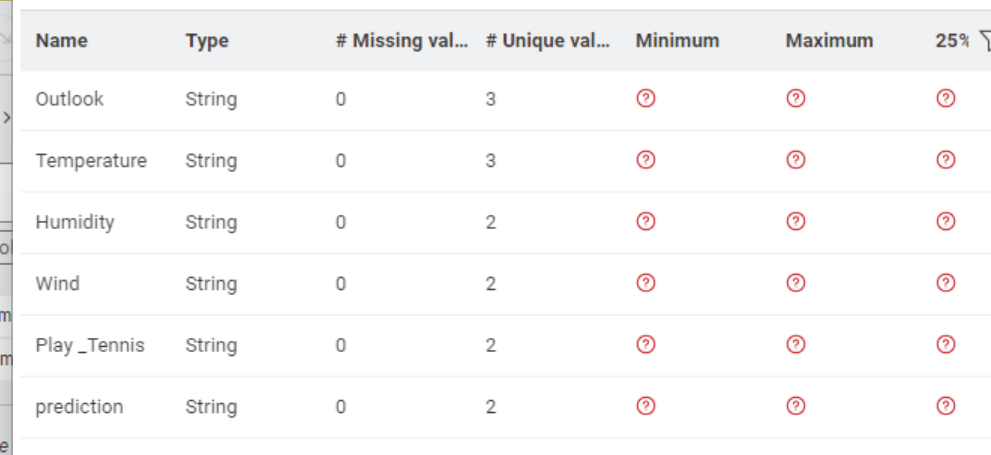
(Date: DD/MM/YY)

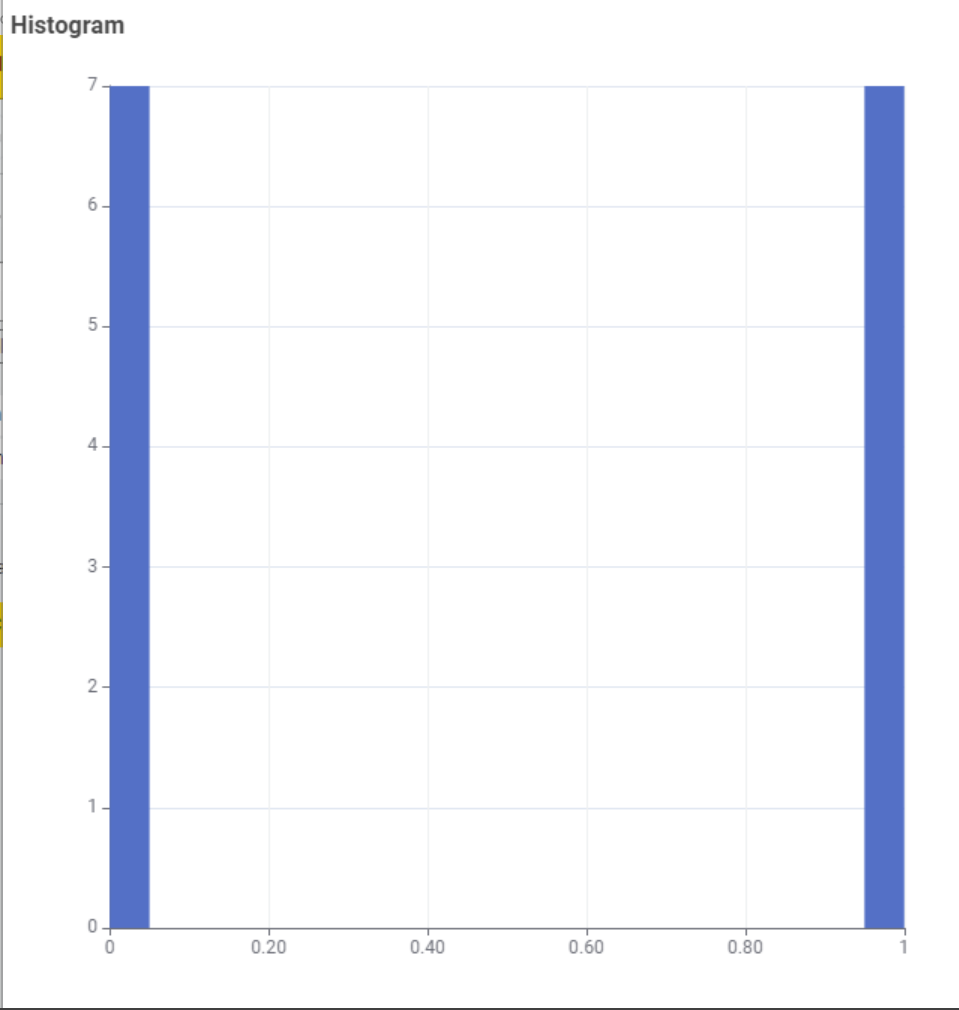
**KNIME TASK:**











**TASK 1:**

**Using python implements VADER rulesbased classification algorithm to find the sentiments of different sentences.**

**SOLUTION:**

import nltk

from nltk.sentiment import SentimentIntensityAnalyzer

import matplotlib.pyplot as plt

sid = SentimentIntensityAnalyzer()

def analyze\_sentiment\_and\_generate\_pie\_chart(sentence):

scores = sid.polarity\_scores(sentence)

compound\_score = scores['compound']

if compound\_score >= 0.05:

sentiment = "Positive"

elif compound\_score <= -0.05:

sentiment = "Negative"

else:

sentiment = "Neutral"

print(f"Sentence: {sentence}\nScores: {scores}\nSentiment: {sentiment}\n")

labels = ['Positive', 'Neutral', 'Negative']

sizes = [scores['pos'], scores['neu'], scores['neg']]

colors = ['yellowgreen', 'gold', 'lightcoral']

explode = (0.1, 0, 0) # explode the first slice if Positive is the highest

plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True, startangle=140)

plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle

plt.title('Sentiment Analysis Result')

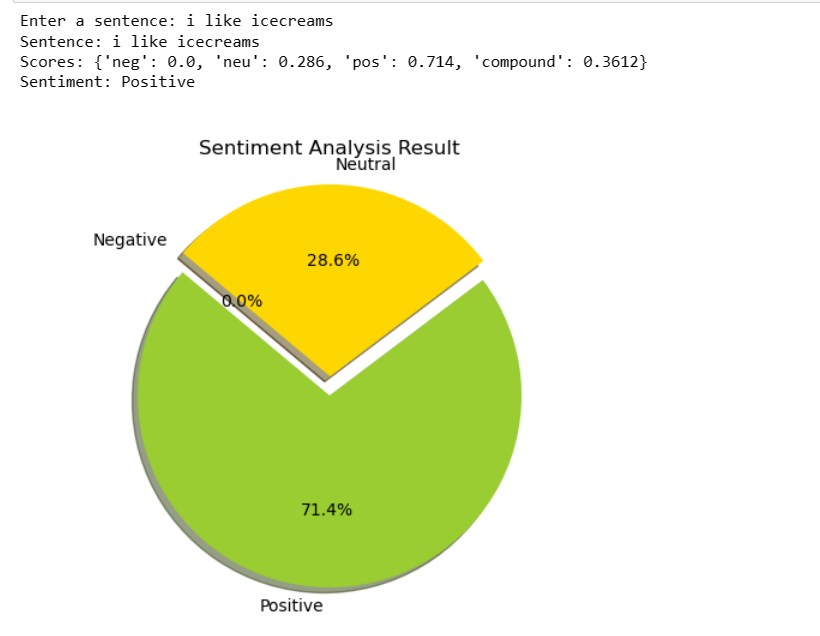
plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

sentence = input("Enter a sentence: ")

analyze\_sentiment\_and\_generate\_pie\_chart(sentence)

**OUTPUT:**



**TASK 2:**

**Using python implements textBlob rules-based classification algorithm to find the sentiments of different sentences and compare the results with task # 01**

**SOLUTION:**

import matplotlib.pyplot as plt

from textblob import TextBlob

import nltk

from nltk.sentiment import SentimentIntensityAnalyzer

# Function to get sentiment using TextBlob

def get\_textblob\_sentiment(sentence):

analysis = TextBlob(sentence)

# TextBlob sentiment.polarity ranges from -1 (negative) to 1 (positive)

return analysis.sentiment.polarity

# Function to get sentiment using VADER

def get\_vader\_sentiment(sentence):

sid = SentimentIntensityAnalyzer()

scores = sid.polarity\_scores(sentence)

# VADER compound score ranges from -1 (negative) to 1 (positive)

return scores['compound']

# Function to display a pie chart for sentiment comparison

def display\_comparison\_pie\_chart(textblob\_polarity, vader\_compound):

labels = ['TextBlob', 'VADER']

sizes = [textblob\_polarity, vader\_compound]

colors = ['lightblue', 'lightgreen']

# Determine explode distance based on sentiment polarity

explode\_distance = 0.1

explode = (explode\_distance if textblob\_polarity < 0 else 0, explode\_distance if vader\_compound < 0 else 0)

plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%',

shadow=True, startangle=90)

plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.

plt.title('Sentiment Analysis Comparison')

plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

sentence = input("Enter a sentence for sentiment analysis: ")

textblob\_polarity = get\_textblob\_sentiment(sentence)

vader\_compound = get\_vader\_sentiment(sentence)

print(f"TextBlob Sentiment Polarity: {textblob\_polarity}\nVADER Sentiment Compound: {vader\_compound}")

# Adjusting values for comparison in pie chart

# The pie chart requires non-negative values, and we're interested in showing the magnitude of sentiment

# The following adjustments allow for a comparative visualization regardless of sentiment direction

textblob\_pie\_value = (textblob\_polarity + 1) \* 50 # Adjusting scale from [-1, 1] to [0, 100]

vader\_pie\_value = (vader\_compound + 1) \* 50 # Adjusting scale from [-1, 1] to [0, 100]

display\_comparison\_pie\_chart(textblob\_pie\_value, vader\_pie\_value)

**OUTPUT:**

