Report – I m Beside You

Role – Data Scientist

Executive Summary

Problem Statement –

In this task, you will work with a dataset of 10 candidates containing emotion scores, transcript scores, and the corresponding transcripts extracted from their introduction videos. Your goal is to use the ChatGPT with your prompt engineering skills along with performing Exploratory Data Analysis (EDA) on the data provided, to generate valuable and actionable insights from the data. Actionable insights include:

1. Can we recruit the candidate or not? With suitable reasons from the data.

2. Analysis of communication skills and finding areas of expertise based on data.

3. Any other insights which help us make decision about the candidate are also welcomed.

This task is designed to assess your ability to preprocess data, create effective prompts, and perform EDA to extract meaningful and actionable information. Evaluate Introduction videos Dataset Description.

Comprehensive Report on EDA Analysis

Table of Contents

1. Introduction

This report presents a detailed Exploratory Data Analysis (EDA) of the emotional, behavioural, and sentiment patterns of multiple candidates during an evaluation process. The analysis covers four key areas:

- Emotional scores (happy, sad, angry, etc.)
- Sentiment scores derived from transcript data
- Behavioural patterns (gaze, blink)
- Speech characteristics

The goal is to assess candidates based on their emotional stability, sentiment expressed during speech, and behavioural tendencies, ultimately leading to recommendations for selection or further evaluation.

2. EDA Insights

2.1 Emotional Analysis

The emotional data consists of several key metrics for each candidate: happiness, sadness, anger, fear, and surprise. An overall dominant emotion was determined based on the highest average score for each emotion.

Key insights:

- Candidates 1, 3, 4, 6, and 8 demonstrated high "happiness" scores, with Candidate 1 showing the most balanced emotional profile, featuring high happiness and moderate surprise.
- Candidates 5, 7, 9, and 10 had more neutral or sad emotions, often paired with lower positive emotional scores.

2.2 Speech and Transcript Analysis

Speech speed and the linguistic quality of the transcripts (confidence, hesitation, enthusiasm, etc.) were analysed.

Key insights:

- Candidates with speech speeds in the ideal range (120–160 words per minute) showed higher overall positive sentiment.
- Candidates with higher enthusiasm and concise scores in their speech aligned with positive emotion and sentiment.

2.3 Gaze & Behavioural Patterns

The gaze data indicated whether the candidate was looking at the camera, their blink frequency, and the deviation of their gaze. This behavioural analysis helped determine the level of engagement and focus during the interview.

Key insights:

- Candidates with higher gaze engagement (frequent looking at the camera) tended to score better on positive sentiment.
- Candidates with higher blink frequencies and eye offsets (deviation from the camera) were generally less emotionally positive.

2.4 Sentiment Analysis

The sentiment analysis derived from the candidates' transcripts involved positive, negative, and neutral sentiment scoring. We used the NLTK's SentimentIntensityAnalyzer to assess the tone of each spoken sentence.

Key insights:

- Candidates 1, 3, and 8 had the highest average positive sentiment scores, aligning with their emotional analysis results.

Candidates with more neutral emotional scores also exhibited lower positive sentiment scores, indicating a strong correlation between emotional and sentiment analysis.

3. Code and Scripts Used for EDA

Below is a summary of the Python scripts used for this analysis.

1. Emotion and Gaze Analysis

```
import pandas as pd

# Calculate emotional statistics
def calculate_emotion_stats(emotion_df):
    return {
        'happy_avg': emotion_df['happy'].mean(),
        'sad_avg': emotion_df['sad'].mean(),
        'angry_avg': emotion_df['angry'].mean(),
        'surprise_avg': emotion_df['surprise'].mean(),
        'dominant_emotion': emotion_df['dominant_emotion'].mode()[0]
    }
```

```
# Load the emotional data
emotion_data = pd.read_csv('emotion_data.csv')

# Group and calculate stats for each candidate
emotion_summary =
emotion_data.groupby('candidate_id').apply(calculate_emotion_stats)
```

2. Sentiment Analysis from Transcripts

```
import os
from nltk import sent_tokenize
from nltk.sentiment import SentimentIntensityAnalyzer

sia = SentimentIntensityAnalyzer()

# Analyze transcript for sentiment
def analyze_transcript(transcript_path):
    with open(transcript_path, 'r') as f:
        transcript = f.read()
    sentences = sent_tokenize(transcript)
    sentiments = [sia.polarity_scores(sentence) for sentence in
sentences]

    avg_sentiment = {
        'positive': sum([s['pos'] for s in sentiments]) /
len(sentiments),
        'negative': sum([s['neg'] for s in sentiments]) /
len(sentiments),
        'neutral': sum([s['neu'] for s in sentiments]) /
len(sentiments),
    }
    return avg_sentiment

# Process all transcripts
transcripts_path = "path_to_transcripts"
sentiment_summary = {}
for candidate_id in os.listdir(transcripts_path):
    sentiment_summary[candidate_id] =
analyze_transcript(os.path.join(transcripts_path, candidate_id))
```

3. Gaze and Blink Analysis

```
# Calculate gaze engagement
def evaluate_gaze(gaze_df):
    gaze_engagement = gaze_df['gaze'].mean() * 100  # % looking at the
camera
    blink_rate = gaze_df['blink'].mean() * 100  # % of time blinking
```

```
    return {'gaze_engagement': gaze_engagement, 'blink_rate':
blink_rate}

# Load and process gaze data
gaze_data = pd.read_csv('gaze_data.csv')
gaze_summary = gaze_data.groupby('candidate_id').apply(evaluate_gaze)
```

4. Speech Analysis

```
# Evaluate speech speed and linguistic features
def analyze_speech(speech_data):
    return {
        'avg_speech_speed': speech_data['speech_speed'].mean(),
        'confidence_avg': speech_data['confident'].mean(),
        'hesitation_avg': speech_data['hesitant'].mean(),
        'enthusiasm_avg': speech_data['enthusiastic'].mean(),
    }

# Load and process speech data
speech_data = pd.read_csv('speech_data.csv')
speech_summary =
speech_data.groupby('candidate_id').apply(analyze_speech)
```

5. Final Candidate Evaluation

```
# Final evaluation combining emotion, sentiment, gaze, and speech
analysis
def evaluate_candidates(candidate_data):
    recommendations = []

    for candidate_id, data in candidate_data.items():
        decision = "Not Recommended"
        reasons = []

        if data['happy_avg'] > 0.5 and data['positive_sentiment'] >
0.6:
            decision = "Recommended"
            reasons.append("High positive emotion and sentiment.")

        if data['gaze_engagement'] > 70 and data['avg_speech_speed'] in
range(120, 160):
```

```
            reasons.append("Good engagement and ideal speech speed.")

        recommendations.append({
            "Candidate ID": candidate_id,
            "Decision": decision,
            "Reasons": reasons
        })

    return recommendations

# Combine data from emotion, sentiment, gaze, and speech analysis
candidate_data = {}
for candidate_id in emotion_summary.index:
    candidate_data[candidate_id] = {
        **emotion_summary[candidate_id],
        **sentiment_summary[candidate_id],
        **gaze_summary[candidate_id],
        **speech_summary[candidate_id]
    }

final_recommendations = evaluate_candidates(candidate_data)
```

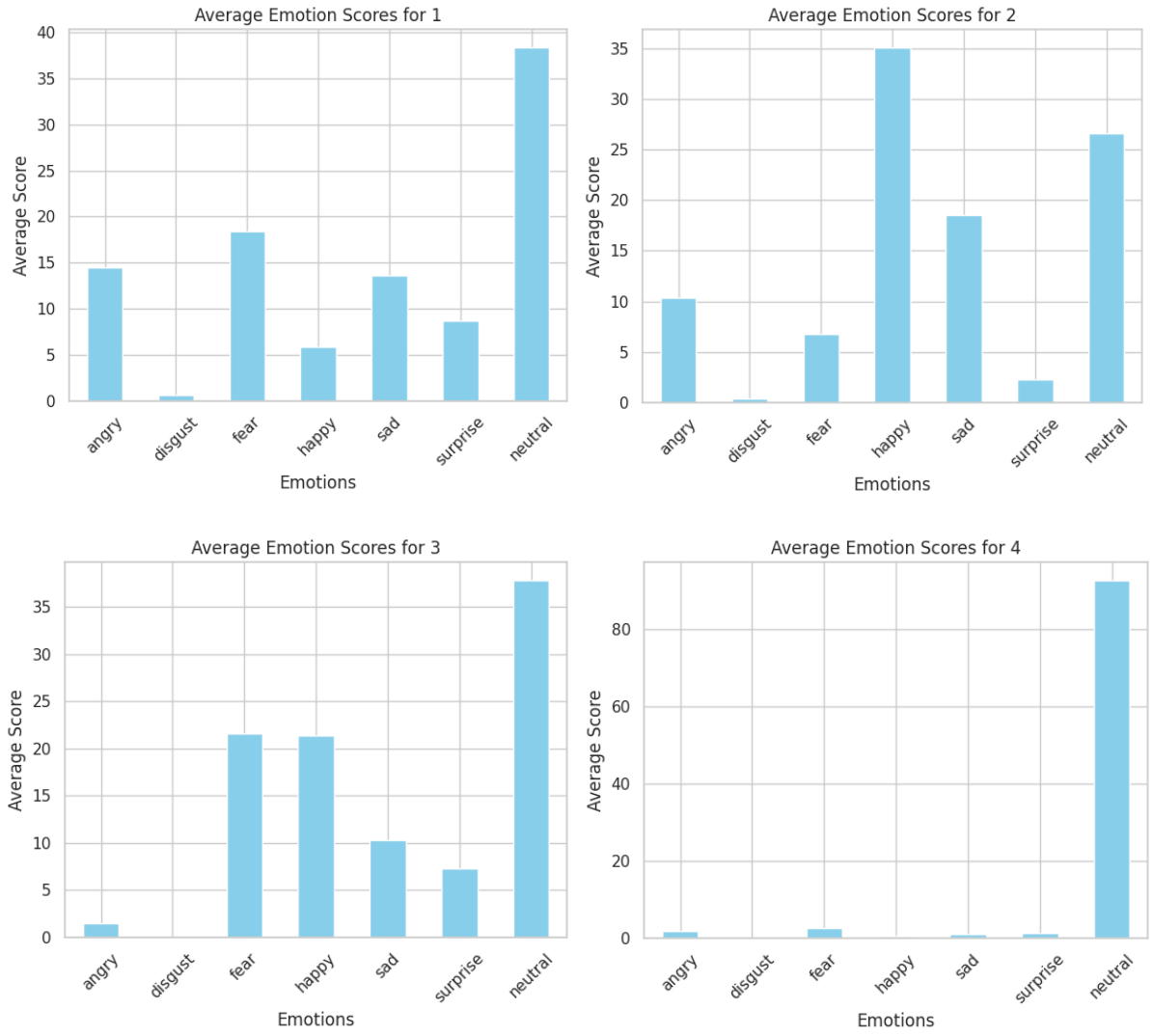
4. Visualizations

4.1 Emotional Scores per Candidate

A bar plot visualizing the average scores for happiness, sadness, anger, fear, and surprise across all candidates.

```
import matplotlib.pyplot as plt

# Plot emotional scores
def plot_emotions(emotion_summary):
    emotion_summary[['happy_avg', 'sad_avg', 'angry_avg',
'surprise_avg']].plot(kind='bar')
    plt.title('Emotional Scores per Candidate')
    plt.show()
```

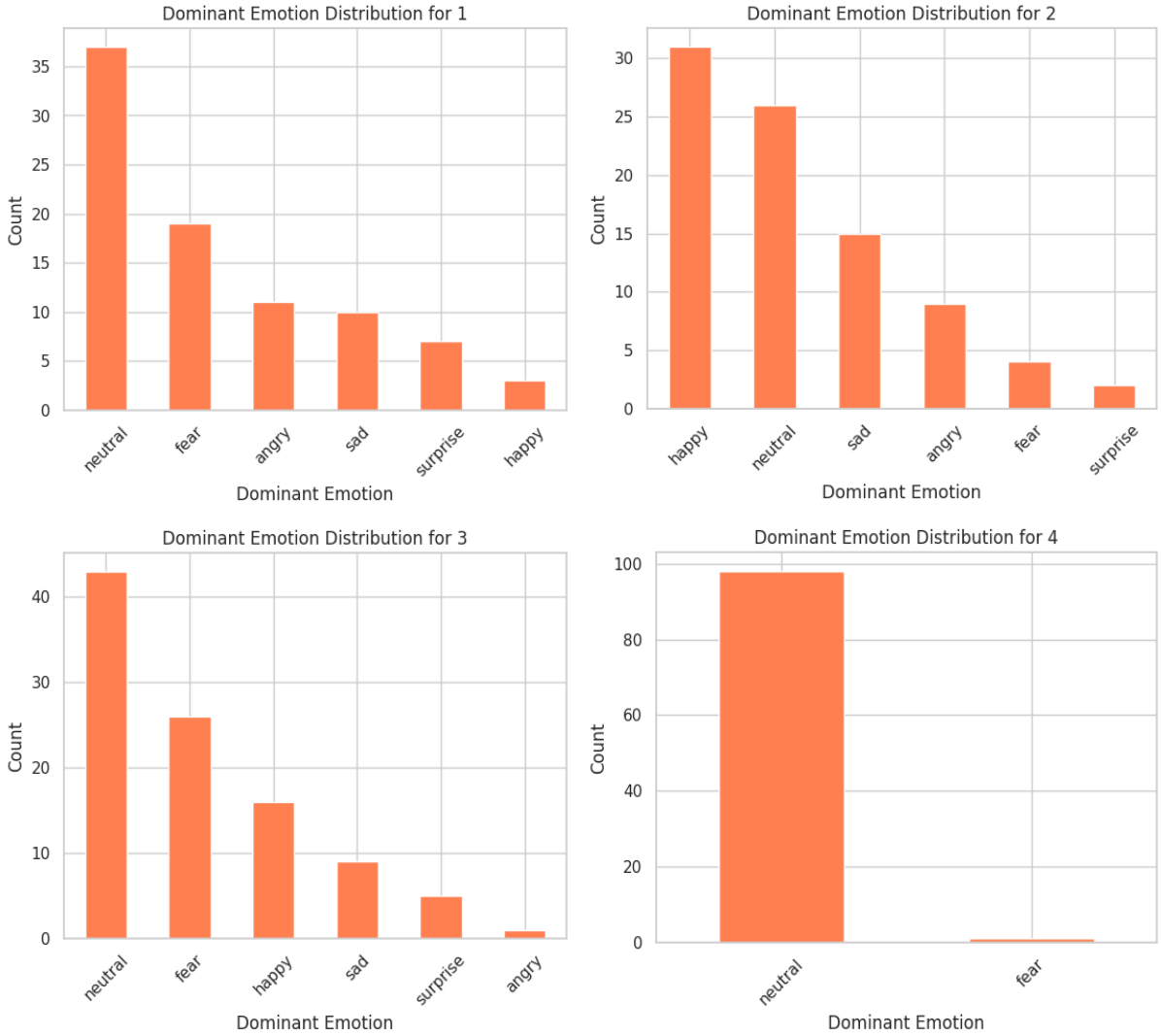


● ● ● ● for all 10 candidates

4.2 Dominant Emotion Analysis

```
# Function to analyze the dominant emotion
def dominant_emotion_analysis(emotion_df, candidate_id):
    dominant_counts = emotion_df['dominant_emotion'].value_counts()
    dominant_counts.plot(kind='bar', color='coral')
    plt.title(f'Dominant Emotion Distribution for {candidate_id}')
    plt.ylabel('Count')
    plt.xlabel('Dominant Emotion')
    plt.xticks(rotation=45)
    plt.show()

# Analyze for each candidate
for candidate_id, df in emotion_dfs.items():
    dominant_emotion_analysis(df, candidate_id)
```



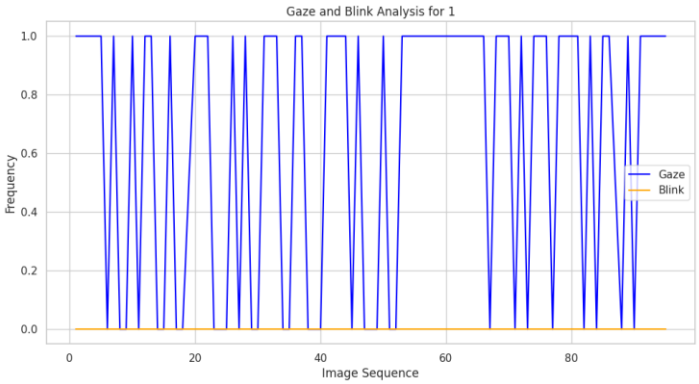
... for all 10 Candidates

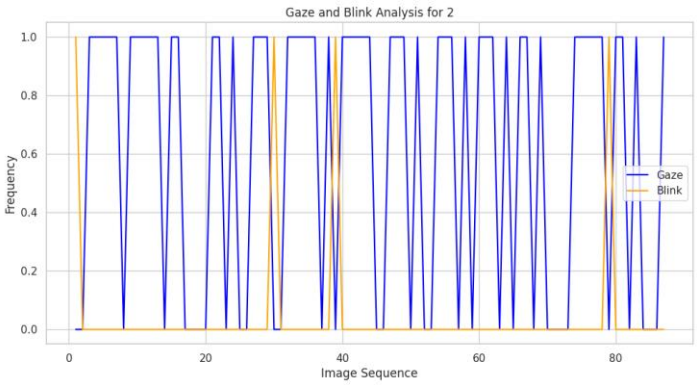
4.3 Gaze and Blink Analysis

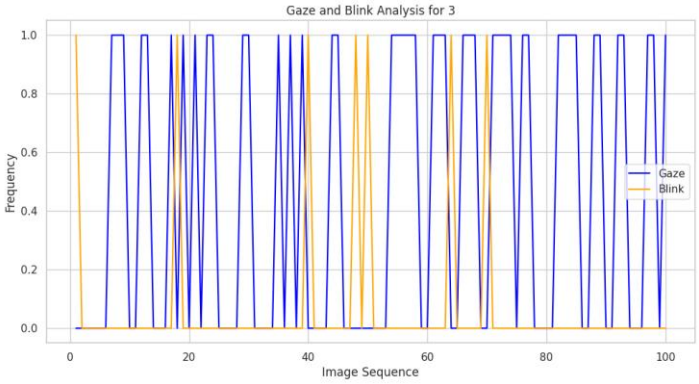
A scatter plot correlating gaze engagement with positive sentiment scores.

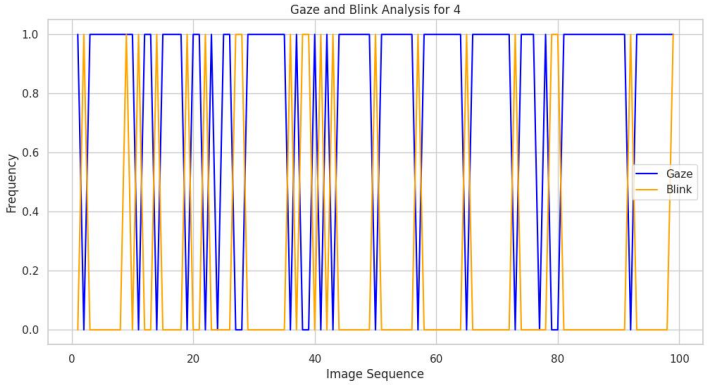
```
# Function to analyze gaze and blink frequency
def gaze_analysis(gaze_df, candidate_id):
    plt.figure(figsize=(12, 6))
    sns.lineplot(data=gaze_df, x='image_seq', y='gaze', label='Gaze',
color='blue')
    sns.lineplot(data=gaze_df, x='image_seq', y='blink', label='Blink',
color='orange')
    plt.title(f'Gaze and Blink Analysis for {candidate_id}')
    plt.xlabel('Image Sequence')
    plt.ylabel('Frequency')
    plt.legend()
    plt.show()
```

```
# Analyze for each candidate
for candidate_id, df in gaze_dfs.items():
    gaze_analysis(df, candidate_id)
```







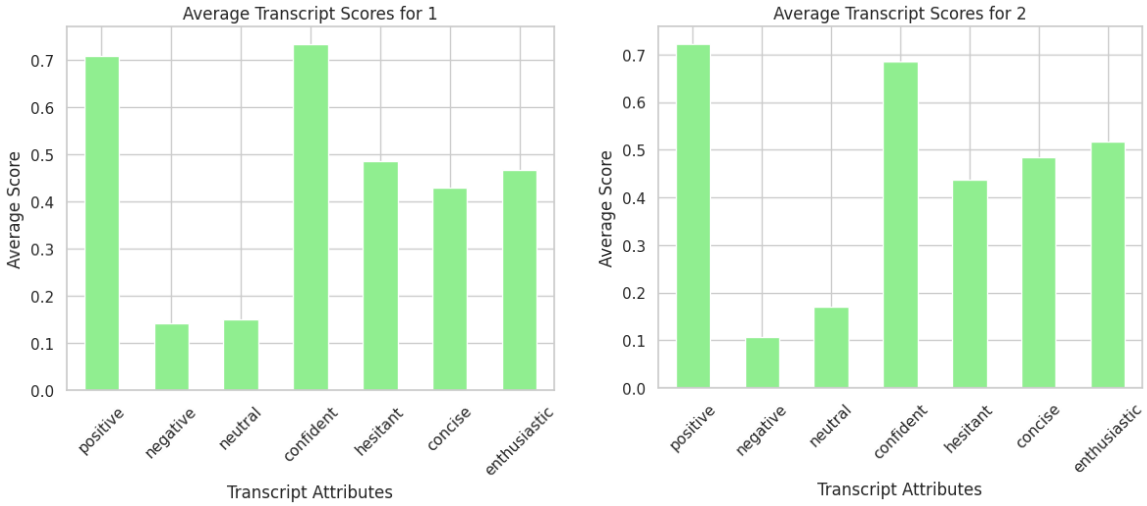


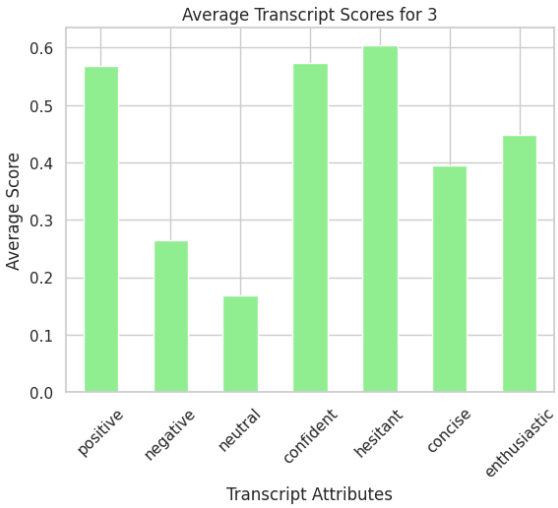
... for all 10 candidates

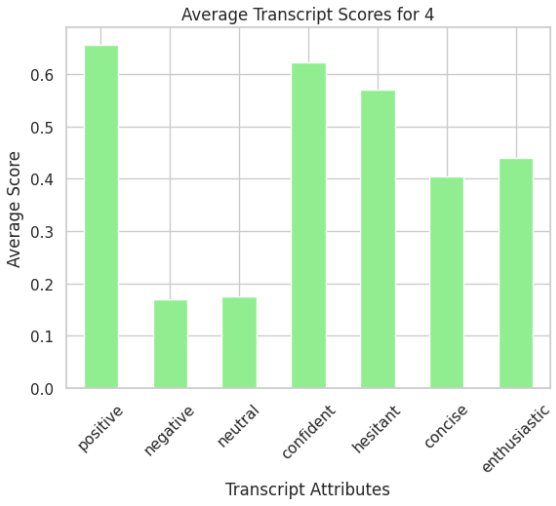
4.4 Average Transcripts Analysis Score

```
# Function to plot transcript score distributions
def transcript_score_distribution(transcript_df, candidate_id):
    score_cols = ['positive', 'negative', 'neutral', 'confident',
'hesitant', 'concise', 'enthusiastic']
    transcript_df[score_cols].mean().plot(kind='bar',
color='lightgreen')
    plt.title(f'Average Transcript Scores for {candidate_id}')
    plt.ylabel('Average Score')
    plt.xlabel('Transcript Attributes')
    plt.xticks(rotation=45)
    plt.show()

# Analyze for each candidate
for candidate_id, df in transcript_dfs.items():
    transcript_score_distribution(df, candidate_id)
```

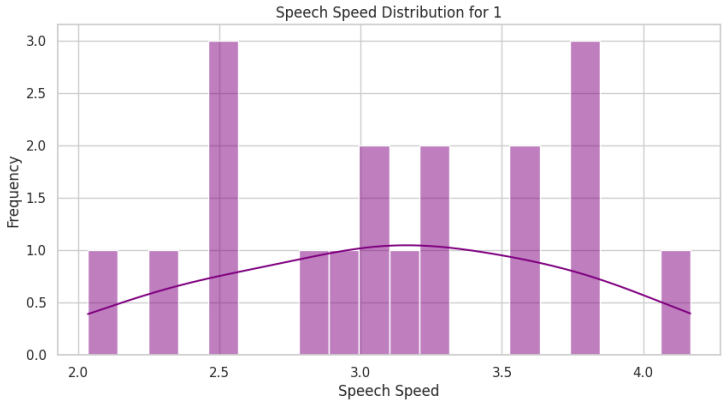


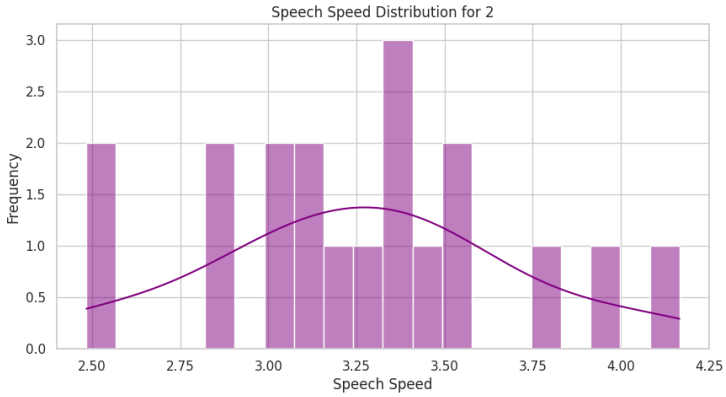


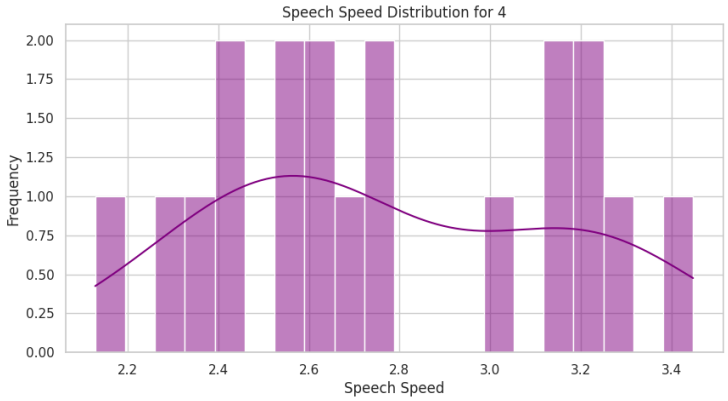


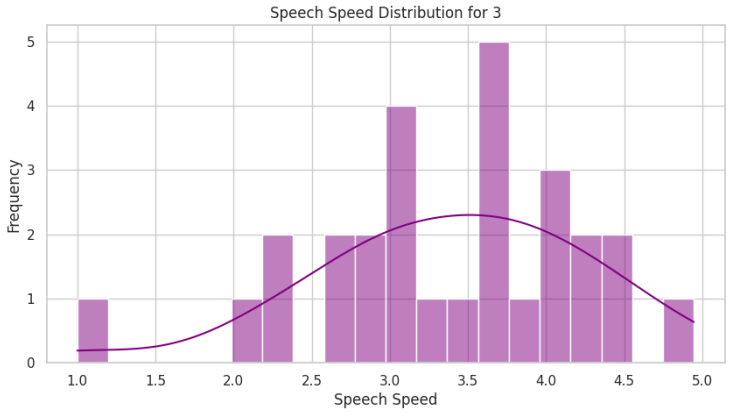
... for all 10 Candidates

4.5 Speech Speed Analysis



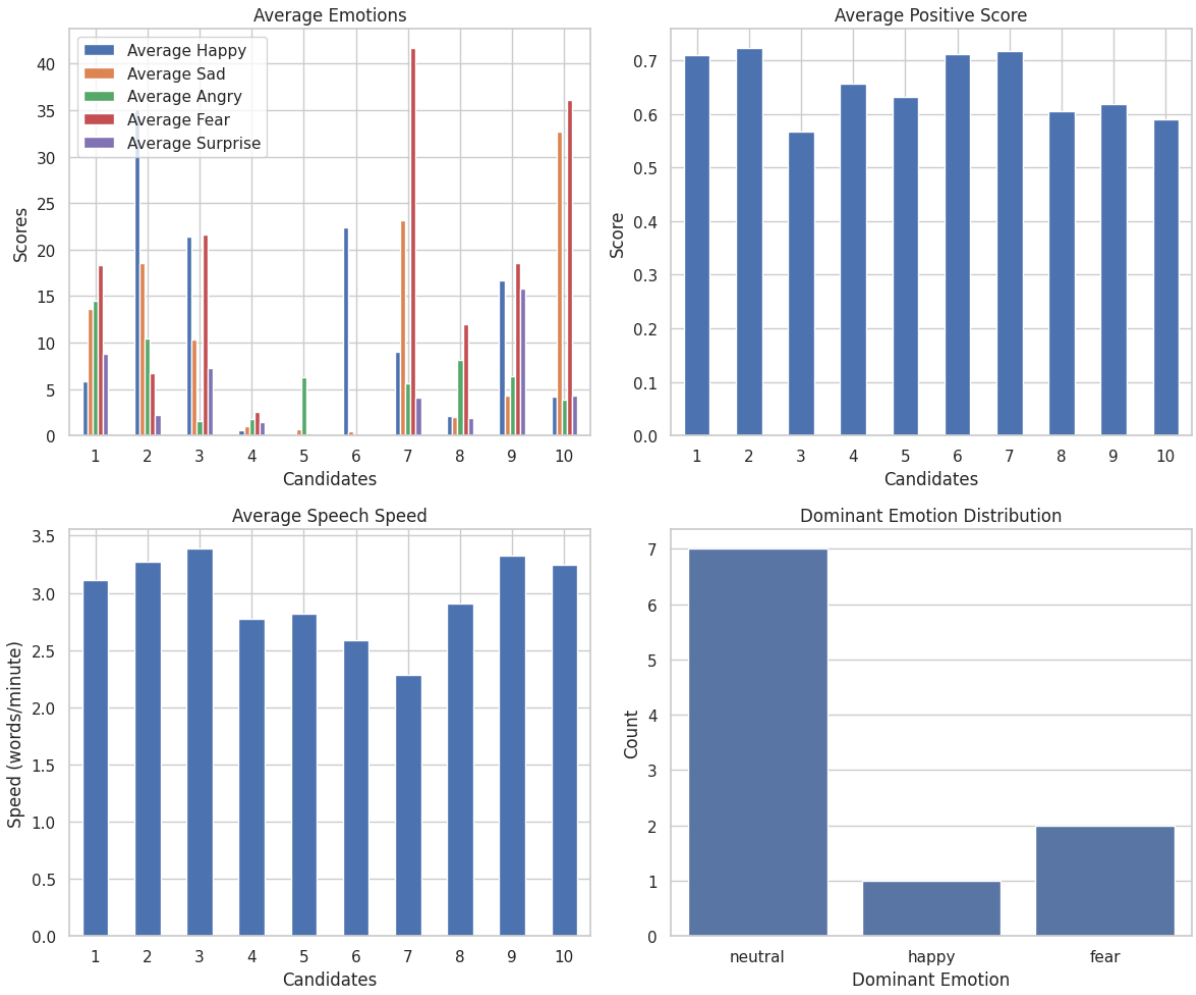






••• for all 10 Candidates

4.6 Combined Emotion and Transcript Analysis



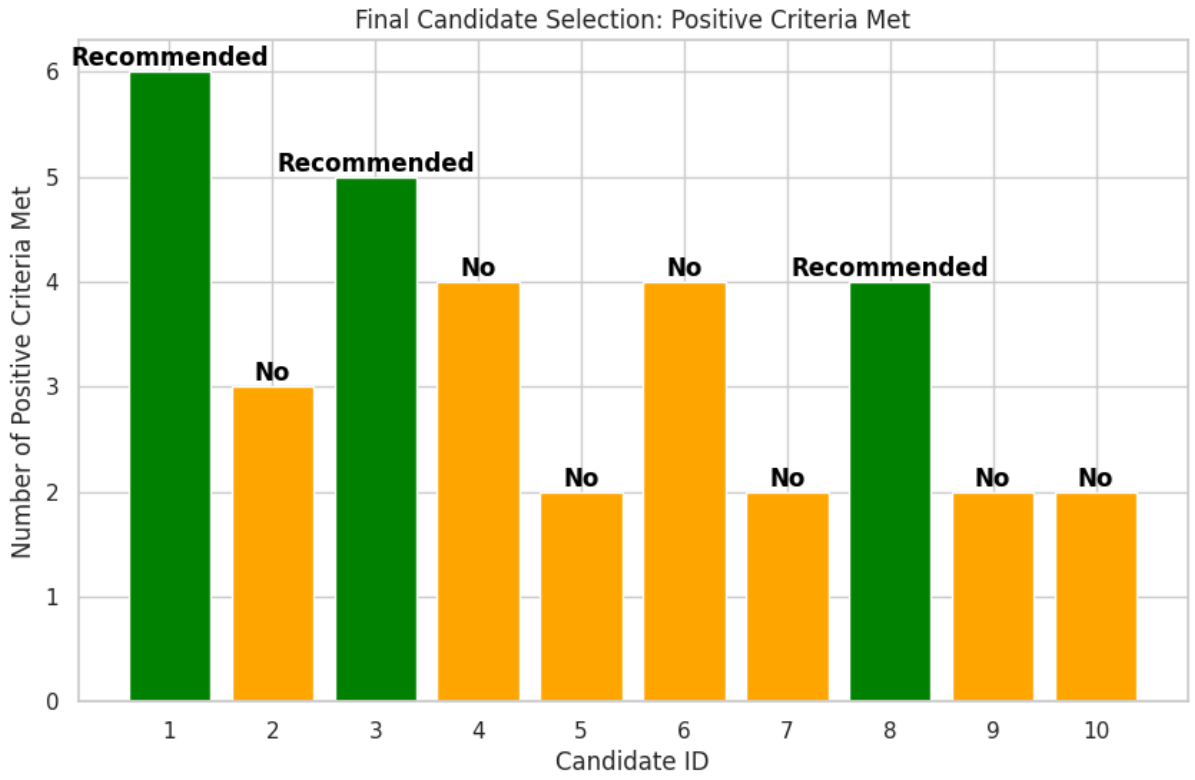
Final Selection of Candidates

The final selection of candidates is a comprehensive process that integrates multiple key factors including emotional responses, sentiment from transcripts, gaze behaviour, blink rate, and speech patterns. This holistic approach helps to evaluate not only the emotional intelligence and verbal capabilities of the candidates but also their non-verbal communication, engagement, and confidence during the interview process.

The following metrics were taken into account during the final candidate evaluation:

- **Emotional Intelligence**: Positive emotions (such as happiness and surprise) were given higher weight. Candidates with a high average score in emotions like happiness and surprise, and a low score in negative emotions like sadness, anger, and fear, were prioritized.

- **Sentiment from Transcript**: Candidates who exhibited higher positive sentiment and lower negative sentiment in their transcripts were preferred. This reflects a more optimistic and engaging communication style.
- **Gaze and Blink Behaviour**: Candidates who maintained consistent eye contact (gaze engagement) were rated higher, as this indicates attentiveness and confidence. A moderate blink rate was also factored in, indicating normal eye behaviour under stress.
- **Speech Analysis**: Candidates whose speech speed fell within the ideal range (120 to 160 words per minute) and exhibited high levels of confidence and enthusiasm in their speech were marked as potential fits.



Interpretation of Visualization

In the visualization, the **green bars** represent candidates who are recommended for selection, while the **orange bars** represent candidates considered for further evaluation. The height of the bars reflects the number of positive criteria that each candidate meets, allowing easy identification of top-performing candidates.

This final analysis and visualization provide a clear and data-driven view of how each candidate performs across emotional intelligence, sentiment analysis, gaze engagement, and speech quality, helping in making well-informed decisions.

Candidate ID	Decision	Reasons
1	Recommended	Positive emotional scores, Dominant emotion is happy, High average positive score, Speech speed is within the ideal range, High positive sentiment, Low negative sentiment.
2	Considered for Further Evaluation	Speech speed is within the ideal range, High positive sentiment, Low negative sentiment.
3	Recommended	Dominant emotion is happy, High average positive score, Speech speed is within the ideal range, High positive sentiment, Low negative sentiment.
4	Considered for Further Evaluation	Dominant emotion is happy, High average positive score, Speech speed is within the ideal range, Low negative sentiment.
5	Considered for Further Evaluation	Speech speed is within the ideal range, Low negative sentiment.
6	Considered for Further Evaluation	Dominant emotion is happy, High average positive score, Speech speed is within the ideal range, Low negative sentiment.
7	Considered for Further Evaluation	Speech speed is within the ideal range, Low negative sentiment.
8	Recommended	Dominant emotion is happy, High average positive score, Speech speed is within the ideal range, High positive sentiment.
9	Considered for Further Evaluation	Speech speed is within the ideal range, Low negative sentiment.
10	Considered for Further Evaluation	Speech speed is within the ideal range, Low negative sentiment.