Untitled17

October 20, 2024

[1]:
```
import random
import pandas as pd # type: ignore

# Define countries and their cities
countries = {
    'USA': ['New York', 'Houston', 'Chicago', 'Miami'],
    'England': ['London', 'Bristol', 'Manchester'],
    'Sri Lanka': ['Colombo', 'Kandy', 'Galle', 'Jaffna'],
    'India': ['Mumbai', 'Delhi', 'Bangalore', 'Chennai'],
    'Dubai': ['Dubai', 'Abu Dhabi', 'Sharjah', 'Al Ain'],
    'Malaysia': ['Kuching', 'Ipoh', 'Kuantan'],
    'Japan': ['Tokyo', 'Osaka', 'Kyoto']
}

# Map cities to cultures
cultures_mapping = {
    'New York': ['Western', 'Latino'],
    'Houston': ['Western', 'Black'],
    'Chicago': ['Western', 'Latino', 'Black'],
    'Miami': ['Western', 'Latino'],
    'London': ['Western', 'African/Caribbean'],
    'Bristol': ['Western', 'South Asian'],
    'Manchester': ['Western', 'African/Caribbean'],
    'Colombo': ['Traditional (Sinhala)', 'Western'],
    'Kandy': ['Traditional (Sinhala)'],
    'Galle': ['Traditional (Sinhala)'],
    'Jaffna': ['Traditional (Tamil)'],
    'Mumbai': ['Western', 'Traditional (North Indian)'],
    'Delhi': ['Traditional (North Indian)'],
    'Bangalore': ['Traditional (South Indian)'],
    'Chennai': ['Traditional (South Indian)'],
    'Dubai': ['Traditional (Middle Eastern)'],
    'Abu Dhabi': ['Traditional (Middle Eastern)'],
    'Sharjah': ['Traditional (Middle Eastern)'],
    'Al Ain': ['Traditional (Middle Eastern)'],
    'Kuching': ['Malay', 'Chinese', 'Indian'],
    'Ipoh': ['Malay', 'Chinese', 'Indian'],
```

```
    'Kuantan': ['Malay'],
    'Tokyo': ['Traditional Japanese', 'Western'],
    'Osaka': ['Traditional Japanese'],
    'Kyoto': ['Traditional Japanese']
}

# Define events and styles
events = ['Office Meeting', 'Wedding', 'Casual Party']
styles = ['Modern', 'Traditional']

# Define culturally appropriate colors and patterns
cultural_colors_patterns = {
    'Western': {
        'Office Meeting': {
            'colors': ['Black', 'Navy', 'Gray', 'Beige'],
            'patterns': ['Solid', 'Pinstripe', 'Checkered']
        },
        'Wedding': {
            'colors': ['White', 'Ivory', 'Pastel', 'Silver', 'Gold'],
            'patterns': ['Lace', 'Embroidered', 'Beaded']
        },
        'Casual Party': {
            'colors': ['Red', 'Blue', 'Green', 'Pink', 'Yellow'],
            'patterns': ['Floral', 'Polka dot', 'Striped', 'Abstract']
        }
    },
    'Latino': {
        'Office Meeting': {
            'colors': ['Beige', 'Brown', 'Cream', 'White'],
            'patterns': ['Solid', 'Subtle Embroidery']
        },
        'Wedding': {
            'colors': ['Red', 'Orange', 'Yellow', 'Gold'],
            'patterns': ['Ruffles', 'Lace', 'Embroidered']
        },
        'Casual Party': {
            'colors': ['Bright', 'Vibrant', 'Multicolor'],
            'patterns': ['Floral', 'Frills', 'Geometric']
        }
    },
    'Black': {
        'Office Meeting': {
            'colors': ['Black', 'Brown', 'Navy', 'Earth Tones'],
            'patterns': ['Solid', 'Subtle Ankara Print']
        },
        'Wedding': {
            'colors': ['Bright', 'Vibrant', 'Gold', 'Kente Colors'],
```

```
            'patterns': ['Ankara Print', 'Kente Patterns', 'Beaded']
        },
        'Casual Party': {
            'colors': ['Bright', 'Vibrant', 'Multicolor'],
            'patterns': ['Ankara Print', 'Dashiki', 'Geometric']
        }
    },
    'African/Caribbean': {
        'Office Meeting': {
            'colors': ['Navy', 'Gray', 'Brown', 'Earth Tones'],
            'patterns': ['Solid', 'Subtle Prints']
        },
        'Wedding': {
            'colors': ['Bright', 'Gold', 'Turquoise', 'Purple'],
            'patterns': ['Ankara', 'Kente', 'Beaded']
        },
        'Casual Party': {
            'colors': ['Vibrant', 'Multicolor', 'Orange', 'Yellow'],
            'patterns': ['Geometric', 'Floral', 'Tribal']
        }
    },
    'South Asian': {
        'Office Meeting': {
            'colors': ['Pastel', 'Earth Tones','Cream'],
            'patterns': ['Solid', 'Minimal Embroidery']
        },
        'Wedding': {
            'colors': ['Red', 'Maroon', 'Gold', 'Pink'],
            'patterns': ['Embroidered', 'Zari', 'Beaded', 'Sequined']
        },
        'Casual Party': {
            'colors': ['Bright', 'Vibrant', 'Pastel'],
            'patterns': ['Printed', 'Floral', 'Block Print']
        }
    },
    'Traditional (Sinhala)': {
        'Office Meeting': {
            'colors': ['Cream', 'Beige','Pastel'],
            'patterns': ['Solid', 'Minimal Embroidery']
        },
        'Wedding': {
            'colors': ['White', 'Gold', 'Cream', 'Maroon'],
            'patterns': ['Embroidered', 'Beaded', 'Lace']
        },
        'Casual Party': {
            'colors': ['Bright', 'Pastel', 'Earth Tones'],
            'patterns': ['Floral', 'Printed', 'Batik']
```

```
        }
    },
    'Traditional (Tamil)': {
        'Office Meeting': {
            'colors': ['Cream', 'Beige','Pastel'],
            'patterns': ['Solid', 'Minimal Embroidery']
        },
        'Wedding': {
            'colors': ['Red', 'Gold', 'Maroon', 'Pink'],
            'patterns': ['Embroidered', 'Zari', 'Beaded']
        },
        'Casual Party': {
            'colors': ['Bright', 'Vibrant', 'Pastel'],
            'patterns': ['Printed', 'Floral', 'Block Print']
        }
    },
    'Traditional (North Indian)': {
        'Office Meeting': {
            'colors': ['Pastel', 'Earth Tones','Cream'],
            'patterns': ['Solid', 'Minimal Embroidery']
        },
        'Wedding': {
            'colors': ['Red', 'Maroon', 'Gold', 'Pink'],
            'patterns': ['Embroidered', 'Zari', 'Beaded', 'Sequined']
        },
        'Casual Party': {
            'colors': ['Bright', 'Vibrant', 'Pastel'],
            'patterns': ['Printed', 'Floral', 'Block Print']
        }
    },
    'Traditional (South Indian)': {
        'Office Meeting': {
            'colors': ['Cream', 'Beige','Pastel'],
            'patterns': ['Solid', 'Minimal Embroidery']
        },
        'Wedding': {
            'colors': ['Red', 'Gold', 'Green', 'Maroon'],
            'patterns': ['Embroidered', 'Zari', 'Beaded']
        },
        'Casual Party': {
            'colors': ['Bright', 'Vibrant', 'Pastel'],
            'patterns': ['Printed', 'Floral', 'Batik']
        }
    },
    'Traditional (Middle Eastern)': {
        'Office Meeting': {
            'colors': ['Black', 'Navy', 'Gray', 'Beige'],
```

```
            'patterns': ['Solid', 'Minimal Embroidery']
        },
        'Wedding': {
            'colors': ['Gold', 'Silver','Emerald'],
            'patterns': ['Embroidered', 'Beaded', 'Sequined']
        },
        'Casual Party': {
            'colors': ['Bright', 'Pastel', 'Earth Tones'],
            'patterns': ['Printed', 'Floral', 'Geometric']
        }
    },
    'Malay': {
        'Office Meeting': {
            'colors': ['Pastel', 'Beige', 'Cream', 'Light Blue'],
            'patterns': ['Solid', 'Minimal Embroidery']
        },
        'Wedding': {
            'colors': ['Gold', 'Silver','Maroon'],
            'patterns': ['Embroidered', 'Beaded', 'Lace']
        },
        'Casual Party': {
            'colors': ['Bright', 'Pastel', 'Multicolor'],
            'patterns': ['Batik', 'Floral', 'Printed']
        }
    },
    'Chinese': {
        'Office Meeting': {
            'colors': ['Black', 'Navy', 'Gray', 'White'],
            'patterns': ['Solid', 'Minimal Embroidery']
        },
        'Wedding': {
            'colors': ['Red', 'Gold', 'Silver'],
            'patterns': ['Embroidered', 'Beaded', 'Traditional Motifs']
        },
        'Casual Party': {
            'colors': ['Bright', 'Pastel', 'Multicolor'],
            'patterns': ['Floral', 'Printed', 'Geometric']
        }
    },
    'Indian': {
        'Office Meeting': {
            'colors': ['Pastel', 'Earth Tones','Cream'],
            'patterns': ['Solid', 'Minimal Embroidery']
        },
        'Wedding': {
            'colors': ['Red', 'Maroon', 'Gold', 'Pink'],
            'patterns': ['Embroidered', 'Zari', 'Beaded', 'Sequined']
```

```
        },
        'Casual Party': {
            'colors': ['Bright', 'Vibrant', 'Pastel'],
            'patterns': ['Printed', 'Floral', 'Block Print']
        }
    },
    'Traditional Japanese': {
        'Office Meeting': {
            'colors': ['Muted', 'Pastel', 'Earth Tones'],
            'patterns': ['Solid', 'Subtle Patterns']
        },
        'Wedding': {
            'colors': ['White', 'Red', 'Gold', 'Silver'],
            'patterns': ['Embroidered', 'Floral', 'Traditional Motifs']
        },
        'Casual Party': {
            'colors': ['Bright', 'Pastel'],
            'patterns': ['Floral', 'Geometric', 'Traditional Prints']
        }
    }
}

# Expanded outfit options (As per previous instruction)
# Note: For brevity, only the 'Western' culture is fully included here.
# You should include all cultures similarly in your actual code.

outfit_options = {
    'Western': {
        'Office Meeting': {
            'Modern': [
                'business suit', 'pencil skirt with blouse',
                'tailored trousers with blazer', 'blazer dress', 'culottes with␣
 ↪shirt',
                'shift dress', 'wrap dress', 'pleated skirt with sweater',
                'ankle-length trousers with blouse', 'sleeveless blouse with␣
 ↪cardigan'
            ],
            'Traditional': [
                'knee-length skirt with blouse', 'conservative dress',
                'turtleneck sweater with skirt', 'button-up shirt with slacks',
                'midi skirt with jacket', 'dress with pantyhose',
                'sweater dress', 'cardigan over dress', 'blouse with long␣
 ↪skirt',
                'twinset with skirt'
            ]
        },
        'Wedding': {
```

```
            'Modern': [
                'ivory lace evening gown',
                'off-shoulder sequin dress',
                'halter-neck satin gown',
                'one-shoulder chiffon dress',
                'asymmetrical tulle dress with floral appliqués',
                'silver beaded sheath gown',
                'embroidered illusion gown with lace details',
                'gold strapless gown with crystal embellishments',
                'modern lace trumpet gown with deep illusion back',
                'silk crepe slip dress with cowl neckline'
            ],

            'Traditional': [
              'knee-length lace dress with cap sleeves',
              'floral A-line dress with a matching shawl',
              'pastel midi dress with a modest neckline',
              'sleeveless pleated dress with a belted waist',
              'tea-length silk dress with a jacket',
              'classic wrap dress with a subtle floral pattern',
              'long-sleeve chiffon dress with an empire waist',
              'vintage-inspired fit-and-flare dress with lace overlay',
              'conservative high-neck sheath dress with a pearl collar',
              'pleated midi skirt with a blouse and matching bolero'
          ]
        },
        'Casual Party': {
            'Modern': [
                 'sundress with floral prints and sandals',
                  'jeans with a stylish off-shoulder top and wedges',
                  'jumpsuit with bold patterns and ankle boots',
                  'romper with lace details and strappy heels',
                  'maxi skirt with a fitted tank top and flats',
                  'off-shoulder blouse with high-waisted jeans and heels',
                  'denim skirt with a graphic tee and sneakers',
                  'crop top with high-waisted pants and block heels',
                  'leggings with an oversized tunic and ballet flats',
                  'shorts with a flowy blouse and espadrilles'
            ],
            'Traditional': [
              'floral A-line dress with a cardigan and ballet flats',
              'knee-length skirt with a blouse and pearl necklace',
              'turtleneck sweater with a pleated skirt and loafers',
              'button-up shirt with tailored trousers and loafers',
              'midi skirt with a sweater and kitten heels',
              'conservative shift dress with a belt and pumps',
              'sweater dress with tights and ankle boots',
```

```
            'cardigan over a dress with Mary Jane shoes',
            'blouse with a long skirt and low heels',
            'twinset with a pencil skirt and classic pumps'
        ]

    }
  },
  'Latino': {
      'Office Meeting': {
          'Modern': [
              'business suit', 'pencil skirt with ruffled blouse',
              'tailored pants with peplum top', 'wrap dress with bold␣
↪patterns',
              'culottes with vibrant shirt', 'blazer with embroidered␣
↪details',
              'sleeveless blouse with cardigan', 'ankle-length trousers with␣
↪frill blouse',
              'floral print dress', 'shift dress with belt'
          ],
          'Traditional': [
              'embroidered blouse with skirt', 'frilled dress',
              'ruffled blouse with trousers', 'peasant top with skirt',
              'bolero jacket with dress', 'tiered skirt with blouse',
              'off-shoulder dress', 'lace blouse with pants',
              'traditional poncho over dress', 'embroidered tunic with␣
↪leggings'
          ]
      },
      'Wedding': {
          'Modern': [
                'evening gown with ruffles',
                'off-shoulder dress with floral embroidery',
                'mermaid gown with lace detailing',
                'halter-neck dress with intricate embroidery',
                'maxi dress with a high slit and lace accents',
                'one-shoulder gown with bold ruffles',
                'asymmetrical hem dress with lace trim',
                'floral applique gown with a fitted bodice',
                'backless dress with intricate beading',
                'lace trumpet gown with a sweetheart neckline'
            ],
          'Traditional': [
                'flamenco-style dress with tiered ruffles',
                'traditional embroidered gown with lace detailing',
                'mantilla veil paired with an embroidered gown',
                'lace bolero over a classic gown',
                'red silk dress with intricate embroidery',
```

```
                'tiered skirt gown with lace accents',
                'embroidered hem dress with a fitted bodice',
                'ruffled train dress with floral appliqué',
                'fringed shawl over a traditional gown',
                'off-shoulder lace dress with detailed embroidery'
            ]

        },
        'Casual Party': {
            'Modern': [
                'off-shoulder top with skirt', 'sundress',
                'romper with floral print', 'crop top with high-waisted shorts',
                'maxi dress with slit', 'peasant blouse with jeans',
                'wrap dress with bold patterns', 'jumpsuit with ruffles',
                'embroidered tunic with leggings', 'frilled top with capris'
            ],
            'Traditional': [
                'ruffled dress', 'peasant blouse with skirt',
                'tiered skirt with embroidered top', 'off-shoulder dress with␣
↪belt',
                'traditional poncho over outfit', 'lace top with flowing skirt',
                'embroidered dress', 'fringed shawl over dress',
                'blouse with flared sleeves', 'floral print skirt with blouse'
            ]
        }
    },
    'Black': {
        'Office Meeting': {
            'Modern': [
                'business suit with African print scarf',
                'blouse with African print skirt',
                'tailored pants with Ankara top', 'blazer with Kente lining',
                'dress with tribal print accents', 'Ankara blazer with␣
↪trousers',
                'peplum top with slacks', 'African print tie with suit',
                'shirt dress with belt', 'pencil skirt with dashiki blouse'
            ],
            'Traditional': [
                'dashiki top with trousers', 'Ankara print dress',
                'Kitenge skirt with blouse', 'Kente wrap dress',
                'mudcloth tunic with pants', 'Ankara peplum top with skirt',
                'boubou dress', 'gele headwrap with dress',
                'African print maxi skirt with top', 'Kaftan with embroidery'
            ]
        },
        'Wedding': {
            'Modern': [
```

```
                'evening gown with Ankara accents',
                'mermaid gown with Kente cloth details',
                'one-shoulder gown with bold geometric patterns',
                'lace dress with tribal print accents',
                'off-shoulder dress with African-inspired embroidery',
                'halter-neck dress with subtle cultural motifs',
                'sequin gown with Ankara print detailing',
                'ball gown with tribal print overlay',
                'asymmetrical gown with beaded accents',
                'fringed evening dress with traditional patterns'
            ]
            ,
            'Traditional': [
                'Kente cloth dress', 'Ankara mermaid dress',
                'gele headwrap with lace dress', 'beaded Maasai dress',
                'Yoruba iro and buba', 'Zulu traditional attire',
                'Igbo wrapper and blouse', 'Ashanti Kente gown',
                'Ethiopian habesha kemis', 'Swahili kanga dress'
            ]
        },
        'Casual Party': {
            'Modern': [
                'African print jumpsuit with bold patterns',
                'dashiki tunic with geometric accents',
                'off-shoulder top with Ankara print detailing',
                'maxi skirt with tribal pattern overlay',
                'crop top with African-inspired prints',
                'high-waisted pants with Kente cloth details',
                'romper with Ankara pattern accents',
                'sleeveless blouse with subtle tribal motifs',
                'shorts with traditional African print designs',
                'jumpsuit with beaded accents and cultural motifs'
            ],
            'Traditional': [
                'dashiki dress', 'Ankara skirt with blouse',
                'kitenge wrap skirt with top', 'boubou dress with headwrap',
                'kaftan with embroidery', 'gele with maxi dress',
                'mudcloth tunic dress', 'kanga wrap dress',
                'batik print dress', 'beaded necklace with dress'
            ]
        }
    },
    'African/Caribbean': {
        'Office Meeting': {
            'Modern': [
                'business suit with cultural accessories',
                'plain dress with headwrap', 'blazer with bright lining',
```

```
                'tailored pants with printed blouse', 'shift dress with scarf',
                'skirt suit with cultural motifs', 'peplum top with trousers',
                'button-down shirt with statement necklace', 'midi dress with␣
↪belt',
                'culottes with blouse'
            ],
            'Traditional': [
                'Kente skirt with blouse', 'dashiki top with trousers',
                'Ankara dress', 'kaftan with pants', 'wrap skirt with top',
                'gele headwrap with dress', 'batik print suit',
                'boubou with embroidery', 'kanga skirt with blouse',
                'traditional tunic with leggings'
            ]
        },
        'Wedding': {
            'Modern': [
                'colorful gown with cultural patterns', 'evening dress with␣
↪headwrap',
                'mermaid dress with lace', 'ball gown with bold prints',
                'off-shoulder gown with embroidery', 'halter-neck dress with␣
↪beadwork',
                'sequined gown with cultural motifs', 'asymmetrical hem dress',
                'one-shoulder gown with ruffles', 'lace dress'
            ],
            'Traditional': [
                'traditional Kente dress', 'African lace dress',
                'Ankara mermaid gown', 'gele with embroidered dress',
                'Caribbean madras dress', 'beaded Maasai gown',
                'Yoruba iro and buba', 'Kente ball gown',
                'Swahili kanga dress', 'Zulu traditional attire'
            ]
        },
        'Casual Party': {
            'Modern': [
                'maxi dress with prints', 'jumpsuit with patterns',
                'Ankara skirt with tank top', 'off-shoulder top with jeans',
                'crop top with printed shorts', 'romper with cultural motifs',
                'peplum blouse with leggings', 'wrap dress with bold patterns',
                'fringed top with capris', 'kaftan dress'
            ],
            'Traditional': [
                'dashiki dress', 'wrap skirt with blouse',
                'Ankara tunic with leggings', 'kitenge dress',
                'boubou with headwrap', 'kanga wrap dress',
                'batik print skirt with top', 'mudcloth dress',
                'gele with maxi dress', 'beaded accessories with outfit'
            ]
```

```
        }
    },
    'South Asian': {
        'Office Meeting': {
            'Modern': [
                'salwar kameez in muted tones', 'saree with simple patterns',
                'kurti with trousers', 'churidar with long top',
                'straight-cut suit', 'palazzo pants with kurti',
                'cotton saree', 'long tunic with leggings',
                'formal Anarkali suit', 'front-slit kurti with pants'
            ],
            'Traditional': [
                'cotton saree', 'formal kurti with trousers',
                'block print salwar kameez', 'handloom saree',
                'embroidered kurti with leggings', 'silk blend saree',
                'high-neck kurti with palazzos', 'chikan work suit',
                'Banarasi saree', 'linen saree'
            ]
        },
        'Wedding': {
            'Modern': [
                'lehenga with contemporary design', 'designer saree',
                'gown with ethnic motifs', 'fusion wear dress',
                'Anarkali gown', 'asymmetrical hem lehenga',
                'saree gown', 'cape style lehenga',
                'crop top with skirt', 'jacket style lehenga'
            ],
            'Traditional': [
                'bridal saree', 'heavy embroidered lehenga',
                'Banarasi silk saree', 'Kanjeevaram saree',
                'Zardosi work lehenga', 'Bandhani saree',
                'Paithani saree', 'Gota patti work lehenga',
                'Sharara suit', 'Silk saree with temple border'
            ]
        },
        'Casual Party': {
            'Modern': [
                'Anarkali suit', 'kurti with leggings',
                'long skirt with crop top', 'palazzo pants with tunic',
                'dhoti pants with kurti', 'high-low hem dress',
                'cold shoulder kurti', 'printed jumpsuit',
                'front-slit kurti with jeans', 'kaftan style kurti'
            ],
            'Traditional': [
                'printed saree', 'simple salwar kameez',
                'cotton Anarkali', 'chikankari kurti with leggings',
                'block print suit', 'phulkari dupatta with suit',
```

```
                'handloom saree', 'embroidered kurti with palazzos',
                'mirror work top with skirt', 'bandhani print suit'
            ]
        }
    },
    'Traditional (Sinhala)': {
        'Office Meeting': {
            'Modern': [
                'Western business attire', 'simple saree',
                'kandyan saree with minimal design', 'blazer with pencil skirt␣
↪and subtle embroidery',
                'tailored trousers with formal blouse and minimal detailing',
                'shift dress with structured silhouette and simple accents',
                'blazer dress with sleek lines and light embellishments',
                'wrap dress with formal detailing and modest fit',
                'formal trousers with long-sleeve blouse and subtle patterns',
                'culottes with fitted shirt and understated accents',
                'sleeveless blouse with cardigan and traditional motif',
                'A-line midi skirt with formal blouse and modern cuts',
                'blazer with ankle-length trousers and modest patterns'
            ],
            'Traditional': [
                 'Kandyan saree with lace detailing and embroidered borders',
                'handloom saree with intricate motifs and traditional draping',
                'osariya with beaded embellishments and classic pleats',
                'silk saree with traditional patterns and woven designs',
                'cotton saree with minimalistic embroidery and temple border',
                'saree with traditional Kandyan draping and floral lace␣
↪accents',
                'saree with lacework and intricate beadwork along the edges',
                'Kandyan saree with a veil and delicate hand-embroidered␣
↪details',
                'saree with half drape and subtle geometric patterns',
                'saree with traditional motifs and lightly embroidered borders'
            ]
        },
        'Wedding': {
            'Modern': [
                'designer saree', 'evening gown',
                'saree gown', 'lehenga', 'fusion wear dress',
                'Anarkali gown', 'lace dress', 'off-shoulder gown',
                'mermaid dress', 'ball gown'
            ],
            'Traditional': [
                'Kandyan saree', 'traditional wedding saree',
                'silk saree with heavy embroidery', 'saree with pearl work',
                'golden saree', 'saree with silver work',
```

```
                'saree with intricate beadwork', 'veil with saree',
                'saree with long train', 'Kandyan saree with jewellery'
            ]
        },
        'Casual Party': {
            'Modern': [
                'dress with local patterns', 'stylish kurti',
                'maxi dress', 'jumpsuit', 'skirt with embroidered top',
                'off-shoulder dress', 'crop top with long skirt',
                'palazzo pants with tunic', 'wrap dress', 'printed romper'
            ],
            'Traditional': [
                'casual Kandyan saree', 'cotton saree',
                'handloom dress', 'kurti with leggings',
                'batik print saree', 'sarong with blouse',
                'ethnic skirt with top', 'simple saree with contrast blouse',
                'embroidery work saree', 'linen saree'
            ]
        }
    },
    'Traditional (Tamil)': {
        'Office Meeting': {
            'Modern': [
                'Western business attire', 'plain saree',
                'kurti with trousers', 'formal salwar kameez',
                'cotton saree', 'palazzo pants with kurti',
                'long skirt with blouse', 'straight-cut suit',
                'midi dress', 'shift dress with scarf'
            ],
            'Traditional': [
                'saree with traditional draping', 'silk saree',
                'cotton saree', 'handloom saree', 'simple saree with border',
                'saree with temple design', 'saree with contrast blouse',
                'kerala kasavu saree', 'half saree', 'saree with zari border'
            ]
        },
        'Wedding': {
            'Modern': [
                'designer saree', 'lehenga', 'saree gown',
                'Anarkali dress', 'fusion wear outfit',
                'gown with ethnic motifs', 'off-shoulder dress',
                'mermaid saree', 'crop top with skirt', 'jacket style saree'
            ],
            'Traditional': [
                'bridal saree with gold embroidery', 'half-saree',
                'Kanjeevaram silk saree', 'silk saree with heavy zari work',
                'temple border saree', 'saree with intricate motifs',
```

```
                'traditional red saree', 'saree with paisley design',
                'saree with kundan work', 'bridal saree with veil'
            ]
        },
        'Casual Party': {
            'Modern': [
                'Anarkali suit', 'kurti with leggings',
                'long skirt with top', 'palazzo pants with kurti',
                'dhoti pants with tunic', 'maxi dress',
                'printed jumpsuit', 'cold shoulder kurti',
                'front-slit kurti with jeans', 'off-shoulder top with skirt'
            ],
            'Traditional': [
                'cotton saree', 'simple salwar kameez',
                'handloom saree', 'block print suit',
                'kerala saree', 'embroidered kurti with leggings',
                'mirror work blouse with saree', 'bandhani saree',
                'kantha work saree', 'saree with tassel border'
            ]
        }
    },
    'Traditional (North Indian)': {
        'Office Meeting': {
            'Modern': [
                'salwar kameez', 'Western business attire',
                'kurti with trousers', 'palazzo pants with kurti',
                'cotton saree', 'churidar with long top',
                'straight-cut suit', 'formal Anarkali',
                'saree with simple border', 'front-slit kurti with pants'
            ],
            'Traditional': [
                'saree', 'formal kurti',
                'block print salwar kameez', 'handloom saree',
                'embroidered kurti with leggings', 'Banarasi saree',
                'chikankari suit', 'phulkari dupatta with suit',
                'linen saree', 'silk blend saree'
            ]
        },
        'Wedding': {
            'Modern': [
                'designer lehenga', 'fancy saree',
                'gown with ethnic embroidery', 'Anarkali gown',
                'fusion wear dress', 'crop top with skirt',
                'jacket style lehenga', 'cape style saree',
                'saree gown', 'lehenga with fringe details'
            ],
            'Traditional': [
```

```
                'heavy embroidered saree', 'bridal lehenga',
                'Banarasi silk saree', 'Zardosi work lehenga',
                'Bandhani saree', 'Sharara suit',
                'Gota patti work lehenga', 'Kundan work saree',
                'Phulkari embroidered lehenga', 'Silk saree with temple border'
            ]
        },
        'Casual Party': {
            'Modern': [
                'Anarkali', 'churidar',
                'long skirt with crop top', 'dhoti pants with kurti',
                'palazzo pants with tunic', 'printed jumpsuit',
                'high-low hem dress', 'cold shoulder kurti',
                'front-slit kurti with jeans', 'kaftan style kurti'
            ],
            'Traditional': [
                'printed saree', 'casual salwar kameez',
                'block print suit', 'phulkari dupatta with suit',
                'chikankari kurti with leggings', 'handloom saree',
                'mirror work top with skirt', 'bandhani print suit',
                'kantha work saree', 'batik print kurti with palazzos'
            ]
        }
    },
    'Traditional (South Indian)': {
        'Office Meeting': {
            'Modern': [
                'Western business attire', 'simple saree',
                'kurti with trousers', 'formal salwar kameez',
                'cotton saree', 'palazzo pants with kurti',
                'straight-cut suit', 'long skirt with blouse',
                'shift dress', 'midi dress with scarf'
            ],
            'Traditional': [
                'silk saree', 'cotton saree',
                'handloom saree', 'simple saree with border',
                'saree with temple design', 'Kanjeevaram saree',
                'saree with contrast blouse', 'kerala kasavu saree',
                'half saree', 'saree with zari border'
            ]
        },
        'Wedding': {
            'Modern': [
                'designer saree', 'lehenga',
                'Anarkali gown', 'fusion wear outfit',
                'saree gown', 'gown with ethnic motifs',
                'off-shoulder dress', 'mermaid saree',
```

```
                'crop top with skirt', 'jacket style saree'
            ],
            'Traditional': [
                'Kanjeevaram saree', 'traditional silk saree',
                'bridal saree with heavy zari work', 'silk saree with temple␣
↪border',
                'saree with intricate motifs', 'traditional red saree',
                'saree with gold embroidery', 'saree with paisley design',
                'saree with kundan work', 'bridal saree with veil'
            ]
        },
        'Casual Party': {
            'Modern': [
                'kurti with leggings', 'maxi dress',
                'long skirt with top', 'palazzo pants with kurti',
                'dhoti pants with tunic', 'printed jumpsuit',
                'cold shoulder kurti', 'front-slit kurti with jeans',
                'off-shoulder top with skirt', 'kaftan style kurti'
            ],
            'Traditional': [
                'cotton saree', 'simple salwar',
                'handloom saree', 'block print suit',
                'kerala saree', 'mirror work blouse with saree',
                'bandhani saree', 'kantha work saree',
                'saree with tassel border', 'batik print kurti with palazzos'
            ]
        }
    },
    'Traditional (Middle Eastern)': {
        'Office Meeting': {
            'Modern': [
                'long sleeve blouse with trousers', 'business suit',
                'midi dress with cardigan', 'blazer with modest dress',
                'wide-leg pants with tunic', 'high-neck blouse with skirt',
                'pantsuit with hijab', 'peplum top with trousers',
                'maxi skirt with blouse', 'shirt dress with belt'
            ],
            'Traditional': [
                'abaya', 'hijab with modest dress',
                'kaftan with trousers', 'long tunic with pants',
                'embroidered abaya', 'cloak with dress',
                'traditional jalabiya', 'thobe with hijab',
                'modest gown', 'kimono-style abaya'
            ]
        },
        'Wedding': {
            'Modern': [
```

```
                'evening gown', 'kaftan with modern design',
                'ball gown with sleeves', 'mermaid dress with lace',
                'high-neck gown', 'off-shoulder dress with sleeves',
                'gown with cape', 'sequin dress', 'embellished A-line gown',
                'gown with train'
            ],
            'Traditional': [
                'embroidered abaya', 'traditional kaftan',
                'velvet gown with gold embroidery', 'bejewelled jalabiya',
                'thobe with intricate designs', 'embroidered cloak',
                'gown with veil and hijab', 'gold-threaded dress',
                'traditional bridal hijab', 'silk kaftan with belt'
            ]
        },
        'Casual Party': {
            'Modern': [
                'maxi dress', 'jeans with tunic',
                'jumpsuit with sleeves', 'long skirt with blouse',
                'palazzo pants with top', 'dress with kimono jacket',
                'wide-leg trousers with tunic', 'crop jacket over dress',
                'printed maxi dress', 'shirt dress with leggings'
            ],
            'Traditional': [
                'casual abaya', 'kaftan',
                'jalabiya with minimal embroidery', 'modest dress with hijab',
                'cloak over outfit', 'long tunic with pants',
                'thobe with belt', 'embroidered shawl with dress',
                'kimono-style abaya', 'hijab with maxi dress'
            ]
        }
    },
    'Malay': {
        'Office Meeting': {
            'Modern': [
                'baju kurung with simple patterns', 'Western business attire',
                'baju kebaya with trousers', 'long blouse with skirt',
                'peplum top with batik skirt', 'blazer over baju kurung',
                'modern kebaya', 'formal dress with hijab',
                'tunic with pants', 'midi dress with cardigan'
            ],
            'Traditional': [
                'baju kurung', 'baju kebaya',
                'batik print baju kurung', 'songket skirt with blouse',
                'baju kurung moden', 'kebaya with sarong',
                'traditional batik dress', 'embroidered baju kurung',
                'baju kurung teluk belanga', 'baju kurung cekak musang'
            ]
```

```
        },
        'Wedding': {
            'Modern': [
                'modern baju kurung', 'evening gown',
                'lace dress with hijab', 'mermaid dress',
                'ball gown with sleeves', 'peplum dress',
                'gown with cape', 'sequin dress', 'embroidered gown',
                'off-shoulder dress with sleeves'
            ],
            'Traditional': [
                'bridal baju kurung', 'traditional wedding dress',
                'songket baju kurung', 'kebaya pengantin',
                'embroidered baju kebaya', 'brocade baju kurung',
                'veil with traditional dress', 'gold-threaded songket',
                'traditional attire with tengkolok', 'batik sarong with kebaya'
            ]
        },
        'Casual Party': {
            'Modern': [
                'floral dress', 'stylish baju kurung',
                'jumpsuit with sleeves', 'long skirt with blouse',
                'palazzo pants with tunic', 'maxi dress',
                'peplum top with skirt', 'tunic with leggings',
                'printed dress with hijab', 'shirt dress'
            ],
            'Traditional': [
                'casual baju kebaya', 'batik dress',
                'baju kurung with modern twist', 'sarong with blouse',
                'embroidered baju kurung', 'kebaya with songket skirt',
                'batik print skirt with top', 'baju kurung kedah',
                'kebarung', 'lace kebaya with skirt'
            ]
        }
    },
    'Chinese': {
        'Office Meeting': {
            'Modern': [
                'Western business attire', 'cheongsam-inspired dress',
                'blouse with mandarin collar', 'pantsuit',
                'skirt suit with Chinese motifs', 'shift dress with embroidery',
                'long tunic with leggings', 'peplum top with trousers',
                'midi dress with cardigan', 'button-up shirt with skirt'
            ],
            'Traditional': [
                 'qipao dress',
                'two-piece cheongsam', 'silk blouse with trousers',
                'embroidered tunic', 'long cheongsam with side slits',
```

```
                'cheongsam top with skirt', 'mandarin collar dress',
                'brocade jacket with pants', 'silk suit'
            ]
        },
        'Wedding': {
            'Modern': [
                'modern cheongsam', 'evening gown',
                'mermaid dress with Chinese motifs', 'off-shoulder gown',
                'gown with mandarin collar', 'lace cheongsam',
                'gown with embroidery', 'asymmetrical dress',
                'sequined cheongsam', 'ball gown with sleeves'
            ],
            'Traditional': [
                'red bridal cheongsam', 'traditional qipao',
                'phoenix embroidered dress', 'dragon and phoenix gown',
                'kwa (traditional wedding dress)', 'embroidered mandarin␣
↪jacket',
                'gold-threaded qipao', 'longfeng gua', 'silk brocade dress',
                'veil with traditional dress'
            ]
        },
        'Casual Party': {
            'Modern': [
                'cheongsam-style top with trousers', 'stylish dress',
                'jumpsuit with mandarin collar', 'skirt with embroidered␣
↪blouse',
                'maxi dress with Chinese motifs', 'crop top with high-waisted␣
↪pants',
                'tunic with leggings', 'peplum top with skirt',
                'printed dress', 'off-shoulder top with jeans'
            ],
            'Traditional': [
                'casual cheongsam', 'silk blouse with skirt',
                'embroidered tunic', 'cheongsam dress with modern print',
                'mandarin collar top with jeans', 'brocade skirt with top',
                'short cheongsam', 'traditional blouse with capris',
                'silk wrap dress', 'embroidered jacket with pants'
            ]
        }
    },
    'Indian': {
        'Office Meeting': {
            'Modern': [
                'salwar kameez', 'Western business attire',
                'kurti with trousers', 'palazzo pants with kurti',
                'cotton saree', 'churidar with long top',
                'straight-cut suit', 'formal Anarkali',
```

```
                'saree with simple border', 'front-slit kurti with pants'
            ],
            'Traditional': [
                'saree', 'formal kurti',
                'block print salwar kameez', 'handloom saree',
                'embroidered kurti with leggings', 'Banarasi saree',
                'chikankari suit', 'phulkari dupatta with suit',
                'linen saree', 'silk blend saree'
            ]
        },
        'Wedding': {
            'Modern': [
                'designer lehenga', 'fancy saree',
                'gown with ethnic embroidery', 'Anarkali gown',
                'fusion wear dress', 'crop top with skirt',
                'jacket style lehenga', 'cape style saree',
                'saree gown', 'lehenga with fringe details'
            ],
            'Traditional': [
                'heavy embroidered saree', 'bridal lehenga',
                'Banarasi silk saree', 'Zardosi work lehenga',
                'Bandhani saree', 'Sharara suit',
                'Gota patti work lehenga', 'Kundan work saree',
                'Phulkari embroidered lehenga', 'Silk saree with temple border'
            ]
        },
        'Casual Party': {
            'Modern': [
                'Anarkali', 'churidar',
                'long skirt with crop top', 'dhoti pants with kurti',
                'palazzo pants with tunic', 'printed jumpsuit',
                'high-low hem dress', 'cold shoulder kurti',
                'front-slit kurti with jeans', 'kaftan style kurti'
            ],
            'Traditional': [
                'printed saree', 'casual salwar kameez',
                'block print suit', 'phulkari dupatta with suit',
                'chikankari kurti with leggings', 'handloom saree',
                'mirror work top with skirt', 'bandhani print suit',
                'kantha work saree', 'batik print kurti with palazzos'
            ]
        }
    },
    'Traditional Japanese': {
        'Office Meeting': {
            'Modern': [
                'Western business attire', 'simple kimono',
```

```
                'blouse with kimono sleeves', 'pantsuit',
                'midi dress with cardigan', 'skirt suit with Japanese motifs',
                'tunic with leggings', 'wrap dress',
                'peplum top with trousers', 'shift dress with obi belt'
            ],
            'Traditional': [
                'komon kimono', 'tsukesage kimono',
                'iromuji kimono', 'kimono with simple patterns',
                'kimono with obi sash', 'silk kimono',
                'kimono with haori jacket', 'monochrome kimono',
                'kimono with subtle designs',
            ]
        },
        'Wedding': {
            'Modern': [
                'evening gown', 'Western-style wedding dress',
                'kimono-style gown', 'lace dress with obi belt',
                'mermaid gown', 'ball gown with sleeves',
                'off-shoulder dress', 'gown with Japanese embroidery',
                'gown with cape', 'asymmetrical dress'
            ],
            'Traditional': [
                'shiromuku (white wedding kimono)', 'uchikake (embroidered␣
↪kimono)',
                'hikifurisode', 'kakeshita kimono', 'red uchikake',
                'furisode kimono', 'wedding kimono with cranes',
                'gold embroidered kimono', 'bridal kimono with obi',
                'tsunokakushi (bridal headdress)'
            ]
        },
        'Casual Party': {
            'Modern': [
                'summer dress', 'casual wear with Japanese motifs',
                'wrap dress', 'tunic with leggings',
                'kimono-style top with jeans', 'maxi dress',
                'printed jumpsuit', 'off-shoulder top with skirt',
                'crop top with high-waisted pants', 'shirt dress with belt'
            ],
            'Traditional': [
                'yukata', 'casual kimono',
                'kimono with floral patterns', 'komon kimono with obi',
                'haori over outfit', 'kimono with sash',
                'short kimono with leggings', 'traditional print dress',
                'kimono-style wrap', 'obi belt over dress'
            ]
        }
    }
```

```
}


# The code continues as before, with the rest of the cultures included in the␣
 ↪outfit_options dictionary.

data = []

# Generate data
for country, cities in countries.items():
    for city in cities:
        cultures = cultures_mapping.get(city, [])
        for culture in cultures:
            for event in events:
                for style in styles:
                    # Get outfit choices for the culture, event, and style
                    outfit_choices = outfit_options.get(culture, {}).get(event,␣
 ↪{}).get(style, [])
                    if not outfit_choices:
                        continue
                    # Get culturally appropriate colors and patterns
                    color_pattern_options = cultural_colors_patterns.
 ↪get(culture, {}).get(event, {})
                    if not color_pattern_options:
                        continue
                    colors = color_pattern_options['colors']
                    patterns = color_pattern_options['patterns']
                    # Randomly select a subset of colors and patterns from the␣
 ↪appropriate options
                    selected_colors = random.sample(colors, k=min(5,␣
 ↪len(colors)))
                    selected_patterns = random.sample(patterns, k=min(3,␣
 ↪len(patterns)))
                    for color in selected_colors:
                        for pattern in selected_patterns:
                            for outfit in outfit_choices:
                                # Create outfit description
                                outfit_description = f"{pattern} {outfit}"
                                data.append({
                                    'Country': country,
                                    'City': city,
                                    'Culture': culture,
                                    'Event': event,
                                    'Preferred Color': color,
                                    'Preferred Style': style,
                                    'Outfit': outfit_description
```

```
                                })

print(f"Total rows generated: {len(data)}")

# Shuffle and sample data to get at least 4,000 rows
random.shuffle(data)
data_sampled = data[:4000]

# Create DataFrame and save to CSV
df = pd.DataFrame(data_sampled)
df.to_csv('fashion_dataset.csv', index=False)
```

```
Total rows generated: 24660
```

[2]:
```
# prompt: read dataset

# Assuming your dataset is in a CSV file named 'dataset.csv'
df = pd.read_csv('fashion_dataset.csv')

# Display the first few rows of the dataset
print(df.head())
```

```
   Country      City                      Culture           Event  \
0    Japan     Tokyo        Traditional Japanese    Casual Party
1      USA     Miami                     Western    Casual Party
2  England   Bristol                     Western  Office Meeting
3      USA  New York                      Latino         Wedding
4    India     Delhi  Traditional (North Indian)    Casual Party

  Preferred Color Preferred Style  \
0          Pastel     Traditional
1            Pink     Traditional
2            Navy          Modern
3          Orange          Modern
4         Vibrant          Modern

                                          Outfit
0               Traditional Prints kimono with sash
1  Striped turtleneck sweater with a pleated skir…
2          Checkered sleeveless blouse with cardigan
3           Lace one-shoulder gown with bold ruffles
4                             Block Print Anarkali
```

[3]:
```
# prompt: check number of rows

print(len(df))
```

```
4000
```

[4]:
```
# prompt: check for duplicates

import random
import pandas as pd

# Assuming your dataset is in a CSV file named 'dataset.csv'
df = pd.read_csv('fashion_dataset.csv')

# Display the first few rows of the dataset
print(df.head())

print(len(df))

# Check for duplicate rows based on all columns
duplicate_rows = df[df.duplicated()]

# Print the number of duplicate rows
print("Number of duplicate rows:", len(duplicate_rows))

# Print the duplicate rows (optional)
print("Duplicate rows:")
print(duplicate_rows)
```

```
   Country      City                      Culture           Event  \
0    Japan     Tokyo        Traditional Japanese    Casual Party
1      USA     Miami                     Western    Casual Party
2  England   Bristol                     Western  Office Meeting
3      USA  New York                      Latino         Wedding
4    India     Delhi  Traditional (North Indian)    Casual Party

  Preferred Color Preferred Style  \
0          Pastel     Traditional
1            Pink     Traditional
2            Navy          Modern
3          Orange          Modern
4         Vibrant          Modern

                                          Outfit
0               Traditional Prints kimono with sash
1  Striped turtleneck sweater with a pleated skir…
2          Checkered sleeveless blouse with cardigan
3           Lace one-shoulder gown with bold ruffles
4                             Block Print Anarkali
4000
Number of duplicate rows: 0
Duplicate rows:
Empty DataFrame
Columns: [Country, City, Culture, Event, Preferred Color, Preferred Style,
```

```
Outfit]
Index: []
```

[5]:
```python
# prompt: check missing values

# Check for missing values in the DataFrame
missing_values = df.isnull().sum()

# Print the number of missing values for each column
print("Missing values per column:")
print(missing_values)
```

```
Missing values per column:
Country            0
City               0
Culture            0
Event              0
Preferred Color    0
Preferred Style    0
Outfit             0
dtype: int64
```

[6]:
```python
# Function to categorize outfit
def categorize_outfit(outfit_description):
    if 'business suit' in outfit_description or 'blazer' in outfit_description␣
 ↪or 'pantsuit' in outfit_description:
        return 'Business Attire'
    elif 'evening gown' in outfit_description or 'ball gown' in␣
 ↪outfit_description or 'mermaid dress' in outfit_description:
        return 'Evening Wear'
    elif 'sundress' in outfit_description or 'maxi dress' in outfit_description␣
 ↪or 'casual dress' in outfit_description:
        return 'Casual Wear'
    elif 'saree' in outfit_description or 'lehenga' in outfit_description or␣
 ↪'salwar kameez' in outfit_description or 'kurti' in outfit_description:
        return 'Traditional Wear'
    elif 'kimono' in outfit_description or 'cheongsam' in outfit_description or␣
 ↪'kaftan' in outfit_description or 'abaya' in outfit_description:
        return 'Cultural Wear'
    else:
        return 'Other'

# Apply the function to create a new column
df['Outfit Category'] = df['Outfit'].apply(categorize_outfit)

# Check the unique outfit categories
print("Unique Outfit Categories:")
print(df['Outfit Category'].unique())
```

```
print(f"Number of unique outfit categories: {df['Outfit Category'].nunique()}")
```

```
Unique Outfit Categories:
['Cultural Wear' 'Other' 'Casual Wear' 'Business Attire'
 'Traditional Wear' 'Evening Wear']
Number of unique outfit categories: 6
```

[7]:
```
# List of categorical columns
categorical_columns = ['Country', 'City', 'Culture', 'Event', 'Preferred␣
 ↪Color', 'Preferred Style']

# Use One-Hot Encoding for categorical variables
df_encoded = pd.get_dummies(df, columns=categorical_columns)

# Encode 'Outfit Category' using LabelEncoder
from sklearn.preprocessing import LabelEncoder

le_outfit_category = LabelEncoder()
df_encoded['Outfit Category Encoded'] = le_outfit_category.
 ↪fit_transform(df['Outfit Category'])

# Drop unnecessary columns
df_encoded = df_encoded.drop(['Outfit', 'Outfit Category'], axis=1)

# Display encoded DataFrame info
print(df_encoded.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4000 entries, 0 to 3999
Data columns (total 80 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Country_Dubai                         4000 non-null   bool
 1   Country_England                       4000 non-null   bool
 2   Country_India                         4000 non-null   bool
 3   Country_Japan                         4000 non-null   bool
 4   Country_Malaysia                      4000 non-null   bool
 5   Country_Sri Lanka                     4000 non-null   bool
 6   Country_USA                           4000 non-null   bool
 7   City_Abu Dhabi                        4000 non-null   bool
 8   City_Al Ain                           4000 non-null   bool
 9   City_Bangalore                        4000 non-null   bool
 10  City_Bristol                          4000 non-null   bool
 11  City_Chennai                          4000 non-null   bool
 12  City_Chicago                          4000 non-null   bool
 13  City_Colombo                          4000 non-null   bool
 14  City_Delhi                            4000 non-null   bool
 15  City_Dubai                            4000 non-null   bool
```

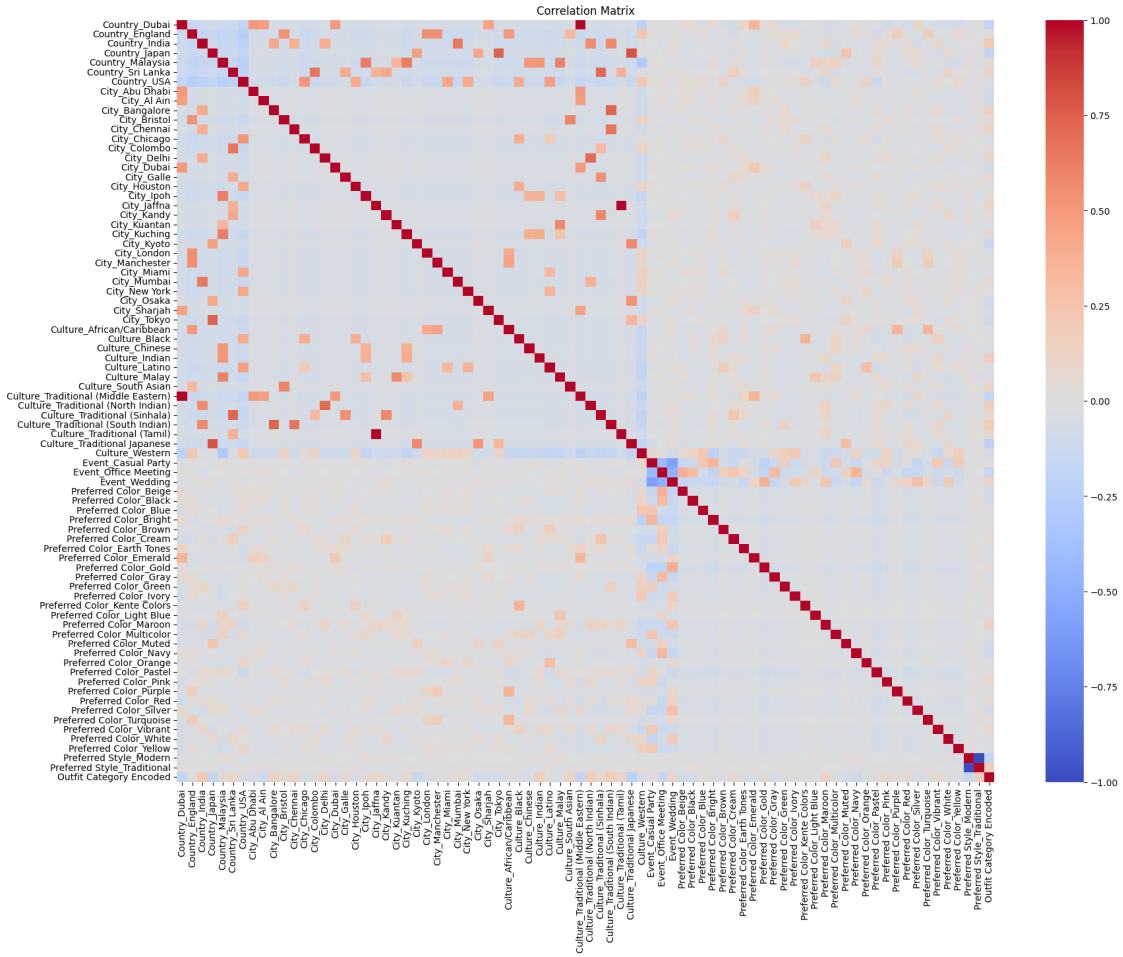
```
16  City_Galle                          4000 non-null   bool
17  City_Houston                        4000 non-null   bool
18  City_Ipoh                           4000 non-null   bool
19  City_Jaffna                         4000 non-null   bool
20  City_Kandy                          4000 non-null   bool
21  City_Kuantan                        4000 non-null   bool
22  City_Kuching                        4000 non-null   bool
23  City_Kyoto                          4000 non-null   bool
24  City_London                         4000 non-null   bool
25  City_Manchester                     4000 non-null   bool
26  City_Miami                          4000 non-null   bool
27  City_Mumbai                         4000 non-null   bool
28  City_New York                       4000 non-null   bool
29  City_Osaka                          4000 non-null   bool
30  City_Sharjah                        4000 non-null   bool
31  City_Tokyo                          4000 non-null   bool
32  Culture_African/Caribbean           4000 non-null   bool
33  Culture_Black                       4000 non-null   bool
34  Culture_Chinese                     4000 non-null   bool
35  Culture_Indian                      4000 non-null   bool
36  Culture_Latino                      4000 non-null   bool
37  Culture_Malay                       4000 non-null   bool
38  Culture_South Asian                 4000 non-null   bool
39  Culture_Traditional (Middle Eastern) 4000 non-null   bool
40  Culture_Traditional (North Indian)  4000 non-null   bool
41  Culture_Traditional (Sinhala)       4000 non-null   bool
42  Culture_Traditional (South Indian)  4000 non-null   bool
43  Culture_Traditional (Tamil)         4000 non-null   bool
44  Culture_Traditional Japanese        4000 non-null   bool
45  Culture_Western                     4000 non-null   bool
46  Event_Casual Party                  4000 non-null   bool
47  Event_Office Meeting                4000 non-null   bool
48  Event_Wedding                       4000 non-null   bool
49  Preferred Color_Beige               4000 non-null   bool
50  Preferred Color_Black               4000 non-null   bool
51  Preferred Color_Blue                4000 non-null   bool
52  Preferred Color_Bright              4000 non-null   bool
53  Preferred Color_Brown               4000 non-null   bool
54  Preferred Color_Cream               4000 non-null   bool
55  Preferred Color_Earth Tones         4000 non-null   bool
56  Preferred Color_Emerald             4000 non-null   bool
57  Preferred Color_Gold                4000 non-null   bool
58  Preferred Color_Gray                4000 non-null   bool
59  Preferred Color_Green               4000 non-null   bool
60  Preferred Color_Ivory               4000 non-null   bool
61  Preferred Color_Kente Colors        4000 non-null   bool
62  Preferred Color_Light Blue          4000 non-null   bool
63  Preferred Color_Maroon              4000 non-null   bool
```

```
 64  Preferred Color_Multicolor            4000 non-null   bool
 65  Preferred Color_Muted                 4000 non-null   bool
 66  Preferred Color_Navy                  4000 non-null   bool
 67  Preferred Color_Orange                4000 non-null   bool
 68  Preferred Color_Pastel                4000 non-null   bool
 69  Preferred Color_Pink                  4000 non-null   bool
 70  Preferred Color_Purple                4000 non-null   bool
 71  Preferred Color_Red                   4000 non-null   bool
 72  Preferred Color_Silver                4000 non-null   bool
 73  Preferred Color_Turquoise             4000 non-null   bool
 74  Preferred Color_Vibrant               4000 non-null   bool
 75  Preferred Color_White                 4000 non-null   bool
 76  Preferred Color_Yellow                4000 non-null   bool
 77  Preferred Style_Modern                4000 non-null   bool
 78  Preferred Style_Traditional           4000 non-null   bool
 79  Outfit Category Encoded               4000 non-null   int64
dtypes: bool(79), int64(1)
memory usage: 340.0 KB
None
```

[8]:
```python
# Define X and y
X = df_encoded.drop(['Outfit Category Encoded'], axis=1)
y = df_encoded['Outfit Category Encoded']

print(f"Feature matrix shape: {X.shape}")
print(f"Target variable shape: {y.shape}")
```

```
Feature matrix shape: (4000, 79)
Target variable shape: (4000,)
```

[9]:
```python
import matplotlib.pyplot as plt
import seaborn as sns

# Compute the correlation matrix
corr_matrix = df_encoded.corr()

# Plot the correlation matrix
plt.figure(figsize=(20, 15))
sns.heatmap(corr_matrix, annot=False, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```



[10]:
```
from sklearn.model_selection import train_test_split

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

print(f"Training set size: {X_train.shape}")
print(f"Testing set size: {X_test.shape}")
```

```
Training set size: (3200, 79)
Testing set size: (800, 79)
```

[11]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
```

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report
```

[12]:
```
# Initialize a dictionary to store accuracies and classification reports
model_accuracies = {}

def train_and_evaluate_model(model, model_name):
    print(f"Training {model_name}...")
    # Train the model
    model.fit(X_train, y_train)

    # Predict on training data
    y_train_pred = model.predict(X_train)
    train_accuracy = accuracy_score(y_train, y_train_pred)

    # Predict on testing data
    y_test_pred = model.predict(X_test)
    test_accuracy = accuracy_score(y_test, y_test_pred)

    # Compute classification report
    report = classification_report(
        y_test, y_test_pred,
        target_names=le_outfit_category.classes_,
        zero_division=0
    )

    print(f"{model_name} Training Accuracy: {train_accuracy:.2f}")
    print(f"{model_name} Testing Accuracy: {test_accuracy:.2f}")
    print(f"{model_name} Classification Report:")
    print(report)
    print("-" * 50)

    # Store accuracies and classification report
    model_accuracies[model_name] = {
        'Train Accuracy': train_accuracy,
        'Test Accuracy': test_accuracy,
        'Classification Report': report
    }

    # Return the trained model and predictions
    return model, y_test_pred
```

[13]:
```
# Model 1: Logistic Regression
logistic_model = LogisticRegression(max_iter=1000, random_state=42)
logistic_model, logistic_y_pred = train_and_evaluate_model(logistic_model,␣
 ↪"Logistic Regression")
```

```
# Model 2: Decision Tree Classifier
decision_tree_model = DecisionTreeClassifier(random_state=42)
decision_tree_model, decision_tree_y_pred =␣
 ↪train_and_evaluate_model(decision_tree_model, "Decision Tree Classifier")

# Model 3: Random Forest Classifier
random_forest_model = RandomForestClassifier(n_estimators=100, random_state=42,␣
 ↪class_weight='balanced')
random_forest_model, random_forest_y_pred =␣
 ↪train_and_evaluate_model(random_forest_model, "Random Forest Classifier")

# Model 4: Support Vector Machine (SVM)
svm_model = SVC(kernel='linear', random_state=42, class_weight='balanced')
svm_model, svm_y_pred = train_and_evaluate_model(svm_model, "Support Vector␣
 ↪Machine")

# Model 5: K-Nearest Neighbors (K-NN)
knn_model = KNeighborsClassifier(n_neighbors=5)
knn_model, knn_y_pred = train_and_evaluate_model(knn_model, "K-Nearest␣
 ↪Neighbors")

# Model 6: Multinomial Naive Bayes
# MultinomialNB requires non-negative features, ensure that your features are␣
 ↪appropriate
naive_bayes_model = MultinomialNB()
naive_bayes_model, naive_bayes_y_pred =␣
 ↪train_and_evaluate_model(naive_bayes_model, "Multinomial Naive Bayes")

# Model 7: Gradient Boosting Classifier
gradient_boosting_model = GradientBoostingClassifier(n_estimators=100,␣
 ↪random_state=42)
gradient_boosting_model, gradient_boosting_y_pred =␣
 ↪train_and_evaluate_model(gradient_boosting_model, "Gradient Boosting␣
 ↪Classifier")
```

```
Training Logistic Regression…
Logistic Regression Training Accuracy: 0.82
Logistic Regression Testing Accuracy: 0.82
Logistic Regression Classification Report:
                  precision    recall  f1-score   support

 Business Attire       0.57      0.17      0.27        23
     Casual Wear       0.00      0.00      0.00        16
   Cultural Wear       0.73      0.60      0.66        40
    Evening Wear       0.00      0.00      0.00        24
           Other       0.85      0.92      0.88       575
Traditional Wear       0.74      0.85      0.79       122
```

```
        accuracy                           0.82       800
       macro avg       0.48      0.42      0.43       800
    weighted avg       0.78      0.82      0.80       800

--------------------------------------------------
Training Decision Tree Classifier…
Decision Tree Classifier Training Accuracy: 0.85
Decision Tree Classifier Testing Accuracy: 0.79
Decision Tree Classifier Classification Report:
                  precision    recall  f1-score   support

 Business Attire       0.31      0.35      0.33        23
     Casual Wear       0.00      0.00      0.00        16
   Cultural Wear       0.52      0.62      0.57        40
    Evening Wear       0.39      0.29      0.33        24
           Other       0.85      0.86      0.85       575
Traditional Wear       0.79      0.79      0.79       122

        accuracy                           0.79       800
       macro avg       0.48      0.48      0.48       800
    weighted avg       0.78      0.79      0.78       800

--------------------------------------------------
Training Random Forest Classifier…
Random Forest Classifier Training Accuracy: 0.74
Random Forest Classifier Testing Accuracy: 0.68
Random Forest Classifier Classification Report:
                  precision    recall  f1-score   support

 Business Attire       0.25      0.48      0.33        23
     Casual Wear       0.04      0.12      0.06        16
   Cultural Wear       0.40      0.75      0.52        40
    Evening Wear       0.21      0.54      0.31        24
           Other       0.88      0.65      0.75       575
Traditional Wear       0.74      0.92      0.82       122

        accuracy                           0.68       800
       macro avg       0.42      0.58      0.46       800
    weighted avg       0.78      0.68      0.71       800

--------------------------------------------------
Training Support Vector Machine…
Support Vector Machine Training Accuracy: 0.55
Support Vector Machine Testing Accuracy: 0.54
Support Vector Machine Classification Report:
                  precision    recall  f1-score   support
```

```
 Business Attire       0.29      0.96      0.44        23
     Casual Wear       0.08      0.62      0.15        16
   Cultural Wear       0.39      0.97      0.56        40
    Evening Wear       0.18      0.83      0.30        24
           Other       0.96      0.39      0.55       575
Traditional Wear       0.71      0.94      0.81       122

        accuracy                           0.54       800
       macro avg       0.44      0.79      0.47       800
    weighted avg       0.83      0.54      0.57       800

--------------------------------------------------
Training K-Nearest Neighbors…
K-Nearest Neighbors Training Accuracy: 0.84
K-Nearest Neighbors Testing Accuracy: 0.81
K-Nearest Neighbors Classification Report:
                  precision    recall  f1-score   support

 Business Attire       0.64      0.30      0.41        23
     Casual Wear       0.11      0.06      0.08        16
   Cultural Wear       0.56      0.47      0.51        40
    Evening Wear       0.50      0.21      0.29        24
           Other       0.85      0.89      0.87       575
Traditional Wear       0.76      0.82      0.79       122

        accuracy                           0.81       800
       macro avg       0.57      0.46      0.49       800
    weighted avg       0.79      0.81      0.79       800

--------------------------------------------------
Training Multinomial Naive Bayes…
Multinomial Naive Bayes Training Accuracy: 0.75
Multinomial Naive Bayes Testing Accuracy: 0.73
Multinomial Naive Bayes Classification Report:
                  precision    recall  f1-score   support

 Business Attire       0.27      0.26      0.27        23
     Casual Wear       0.00      0.00      0.00        16
   Cultural Wear       0.35      0.90      0.50        40
    Evening Wear       0.26      0.25      0.26        24
           Other       0.91      0.72      0.81       575
Traditional Wear       0.64      1.00      0.78       122

        accuracy                           0.73       800
       macro avg       0.41      0.52      0.43       800
    weighted avg       0.79      0.73      0.74       800

--------------------------------------------------
```

```
Training Gradient Boosting Classifier…
Gradient Boosting Classifier Training Accuracy: 0.84
Gradient Boosting Classifier Testing Accuracy: 0.83
Gradient Boosting Classifier Classification Report:
                   precision    recall  f1-score   support

 Business Attire       1.00      0.17      0.30        23
     Casual Wear       0.00      0.00      0.00        16
   Cultural Wear       0.64      0.57      0.61        40
    Evening Wear       0.20      0.04      0.07        24
           Other       0.86      0.92      0.89       575
Traditional Wear       0.76      0.90      0.83       122

        accuracy                           0.83       800
       macro avg       0.58      0.43      0.45       800
    weighted avg       0.80      0.83      0.81       800

--------------------------------------------------
```

[14]:
```
!pip install tensorflow
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-
packages (2.17.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=3.10.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (3.11.0)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.3.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.4.1)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (24.1)
Requirement already satisfied:
protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3
in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.20.3)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.32.3)
```

```
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (71.0.4)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.5.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (1.16.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.64.1)
Requirement already satisfied: tensorboard<2.18,>=2.17 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.17.0)
Requirement already satisfied: keras>=3.2.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (3.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.37.1)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.26.4)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow)
(0.44.0)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow) (13.9.2)
Requirement already satisfied: namex in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow) (0.0.8)
Requirement already satisfied: optree in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow) (0.13.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow)
(3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests<3,>=2.21.0->tensorflow) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow)
(2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow)
(2024.8.30)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.10/dist-packages (from
tensorboard<2.18,>=2.17->tensorflow) (3.7)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from
tensorboard<2.18,>=2.17->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from
```

```
tensorboard<2.18,>=2.17->tensorflow) (3.0.4)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.10/dist-packages (from
werkzeug>=1.0.1->tensorboard<2.18,>=2.17->tensorflow) (3.0.1)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras>=3.2.0->tensorflow)
(3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras>=3.2.0->tensorflow)
(2.18.0)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-
packages (from markdown-it-py>=2.2.0->rich->keras>=3.2.0->tensorflow) (0.1.2)
```

[15]:
```
!pip install keras #This line installs the keras module
```

```
Requirement already satisfied: keras in /usr/local/lib/python3.10/dist-packages
(3.4.1)
Requirement already satisfied: absl-py in /usr/local/lib/python3.10/dist-
packages (from keras) (1.4.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(from keras) (1.26.4)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages
(from keras) (13.9.2)
Requirement already satisfied: namex in /usr/local/lib/python3.10/dist-packages
(from keras) (0.0.8)
Requirement already satisfied: h5py in /usr/local/lib/python3.10/dist-packages
(from keras) (3.11.0)
Requirement already satisfied: optree in /usr/local/lib/python3.10/dist-packages
(from keras) (0.13.0)
Requirement already satisfied: ml-dtypes in /usr/local/lib/python3.10/dist-
packages (from keras) (0.4.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from keras) (24.1)
Requirement already satisfied: typing-extensions>=4.5.0 in
/usr/local/lib/python3.10/dist-packages (from optree->keras) (4.12.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras) (2.18.0)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-
packages (from markdown-it-py>=2.2.0->rich->keras) (0.1.2)
```

[16]:
```
# Install tensorflow if needed
!pip install tensorflow

# Import the necessary modules from TensorFlow instead of Keras
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

```
#from keras.preprocessing.text import Tokenizer  # Remove this import
#from keras.preprocessing.sequence import pad_sequences # Remove this import


from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, SimpleRNN, Dense, Dropout
from tensorflow.keras.utils import to_categorical
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-
packages (2.17.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=3.10.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (3.11.0)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.3.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.4.1)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (24.1)
Requirement already satisfied:
protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3
in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.20.3)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (71.0.4)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.5.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (4.12.2)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (1.16.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.64.1)
```

```
Requirement already satisfied: tensorboard<2.18,>=2.17 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.17.0)
Requirement already satisfied: keras>=3.2.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (3.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.37.1)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.26.4)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow)
(0.44.0)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow) (13.9.2)
Requirement already satisfied: namex in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow) (0.0.8)
Requirement already satisfied: optree in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow) (0.13.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow)
(3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests<3,>=2.21.0->tensorflow) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow)
(2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow)
(2024.8.30)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.10/dist-packages (from
tensorboard<2.18,>=2.17->tensorflow) (3.7)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from
tensorboard<2.18,>=2.17->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from
tensorboard<2.18,>=2.17->tensorflow) (3.0.4)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.10/dist-packages (from
werkzeug>=1.0.1->tensorboard<2.18,>=2.17->tensorflow) (3.0.1)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras>=3.2.0->tensorflow)
(3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras>=3.2.0->tensorflow)
(2.18.0)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-
packages (from markdown-it-py>=2.2.0->rich->keras>=3.2.0->tensorflow) (0.1.2)
```

[17]:
```
# Prepare the text input
texts = df['Outfit'].values

# Tokenization
max_words = 5000
tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(texts)

sequences = tokenizer.texts_to_sequences(texts)
word_index = tokenizer.word_index
print(f"Found {len(word_index)} unique tokens.")

# Padding
max_sequence_length = 20  # Adjust as necessary
data_seq = pad_sequences(sequences, maxlen=max_sequence_length)
print(f"Data shape after padding: {data_seq.shape}")

# Prepare labels
labels_seq = to_categorical(y)
print(f"Labels shape: {labels_seq.shape}")

# Split the data
X_train_seq, X_test_seq, y_train_seq, y_test_seq = train_test_split(
    data_seq, labels_seq, test_size=0.2, random_state=42, stratify=y
)
```

```
Found 370 unique tokens.
Data shape after padding: (4000, 20)
Labels shape: (4000, 6)
```

[18]:
```
# Build the LSTM model
lstm_model = Sequential()
lstm_model.add(Embedding(input_dim=max_words, output_dim=128,␣
 ↪input_length=max_sequence_length))
lstm_model.add(LSTM(64, dropout=0.2, recurrent_dropout=0.2))
lstm_model.add(Dense(labels_seq.shape[1], activation='softmax'))

lstm_model.compile(loss='categorical_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])
lstm_model.summary()

# Train the LSTM model
epochs = 5  # Adjust as needed
batch_size = 32
lstm_history = lstm_model.fit(
    X_train_seq, y_train_seq,
    epochs=epochs,
```

```
    batch_size=batch_size,
    validation_split=0.1
)
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	?	0 (unbuilt)
lstm (LSTM)	?	0 (unbuilt)
dense (Dense)	?	0 (unbuilt)

```
Total params: 0 (0.00 B)

Trainable params: 0 (0.00 B)

Non-trainable params: 0 (0.00 B)
```

```
Epoch 1/5
90/90             4s 22ms/step -
accuracy: 0.6949 - loss: 1.1350 - val_accuracy: 0.8188 - val_loss: 0.6300
Epoch 2/5
90/90             2s 18ms/step -
accuracy: 0.8673 - loss: 0.4504 - val_accuracy: 0.9062 - val_loss: 0.2812
Epoch 3/5
90/90             3s 31ms/step -
accuracy: 0.9483 - loss: 0.1819 - val_accuracy: 0.9688 - val_loss: 0.1160
Epoch 4/5
90/90             2s 21ms/step -
accuracy: 0.9824 - loss: 0.0724 - val_accuracy: 0.9969 - val_loss: 0.0370
Epoch 5/5
90/90             2s 19ms/step -
accuracy: 0.9923 - loss: 0.0347 - val_accuracy: 1.0000 - val_loss: 0.0179
```

[19]:
```
import numpy as np # Import the numpy library with the alias np

# Evaluate the LSTM model
lstm_scores = lstm_model.evaluate(X_test_seq, y_test_seq, verbose=0)
print(f"LSTM Test Accuracy: {lstm_scores[1]:.2f}")

# Predict classes
lstm_y_pred_probs = lstm_model.predict(X_test_seq)
lstm_y_pred = np.argmax(lstm_y_pred_probs, axis=1) # Now np is defined and can␣
 ↪be used
y_test_classes = np.argmax(y_test_seq, axis=1)

print("LSTM Classification Report:")
print(classification_report(y_test_classes, lstm_y_pred,␣
 ↪target_names=le_outfit_category.classes_, zero_division=0))
```

```
LSTM Test Accuracy: 1.00
25/25             0s 5ms/step
LSTM Classification Report:
                  precision    recall  f1-score   support

 Business Attire       1.00      1.00      1.00        23
     Casual Wear       1.00      1.00      1.00        16
   Cultural Wear       1.00      1.00      1.00        40
    Evening Wear       0.96      1.00      0.98        24
           Other       1.00      1.00      1.00       575
Traditional Wear       1.00      1.00      1.00       122

        accuracy                           1.00       800
       macro avg       0.99      1.00      1.00       800
    weighted avg       1.00      1.00      1.00       800
```

[20]:
```
# Build the Simple RNN model
rnn_model = Sequential()
rnn_model.add(Embedding(input_dim=max_words, output_dim=128,␣
 ↪input_length=max_sequence_length))
rnn_model.add(SimpleRNN(64, dropout=0.2, recurrent_dropout=0.2))
rnn_model.add(Dense(labels_seq.shape[1], activation='softmax'))

rnn_model.compile(loss='categorical_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])
rnn_model.summary()

# Train the RNN model
rnn_history = rnn_model.fit(
    X_train_seq, y_train_seq,
```

```
    epochs=epochs,
    batch_size=batch_size,
    validation_split=0.1
)
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90:
UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
```

```
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	?	0 (unbuilt)
simple_rnn (SimpleRNN)	?	0 (unbuilt)
dense_1 (Dense)	?	0 (unbuilt)

```
 Total params: 0 (0.00 B)

 Trainable params: 0 (0.00 B)

 Non-trainable params: 0 (0.00 B)
```

```
Epoch 1/5
90/90             4s 19ms/step -
accuracy: 0.6054 - loss: 1.1928 - val_accuracy: 0.8062 - val_loss: 0.6791
Epoch 2/5
90/90             1s 12ms/step -
accuracy: 0.8575 - loss: 0.4881 - val_accuracy: 0.9219 - val_loss: 0.2877
Epoch 3/5
90/90             1s 13ms/step -
accuracy: 0.9413 - loss: 0.2047 - val_accuracy: 0.9625 - val_loss: 0.1392
Epoch 4/5
90/90             2s 17ms/step -
accuracy: 0.9777 - loss: 0.1124 - val_accuracy: 0.9812 - val_loss: 0.0779
Epoch 5/5
90/90             3s 18ms/step -
```

```
accuracy: 0.9855 - loss: 0.0700 - val_accuracy: 0.9875 - val_loss: 0.0494
```

[22]:
```python
# Evaluate the RNN model
rnn_scores = rnn_model.evaluate(X_test_seq, y_test_seq, verbose=0)
print(f"Simple RNN Test Accuracy: {rnn_scores[1]:.2f}")

# Predict classes
rnn_y_pred_probs = rnn_model.predict(X_test_seq)
rnn_y_pred = np.argmax(rnn_y_pred_probs, axis=1)

print("Simple RNN Classification Report:")
print(classification_report(y_test_classes, rnn_y_pred,␣
  ↪target_names=le_outfit_category.classes_, zero_division=0))
```

```
Simple RNN Test Accuracy: 0.99
25/25             0s 5ms/step
Simple RNN Classification Report:
                  precision    recall  f1-score   support

 Business Attire       1.00      0.91      0.95        23
     Casual Wear       1.00      0.94      0.97        16
   Cultural Wear       0.97      0.97      0.97        40
    Evening Wear       0.88      0.92      0.90        24
           Other       0.99      1.00      0.99       575
Traditional Wear       1.00      1.00      1.00       122

        accuracy                           0.99       800
       macro avg       0.97      0.96      0.97       800
    weighted avg       0.99      0.99      0.99       800
```

[24]:
```python
# prompt: i need to hyper parameter tunning and cross validation for gradient␣
  ↪booster

from sklearn.model_selection import GridSearchCV

# Define the parameter grid for Gradient Boosting
param_grid = {
    'n_estimators': [50, 100, 150],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 4, 5],
    'subsample': [0.8, 1.0],
}

# Create a Gradient Boosting Classifier
gradient_boosting_model = GradientBoostingClassifier(random_state=42)

# Create a GridSearchCV object
```

```python
grid_search = GridSearchCV(
    estimator=gradient_boosting_model,
    param_grid=param_grid,
    scoring='accuracy',
    cv=5,  # Use 5-fold cross-validation
    n_jobs=-1  # Use all available CPU cores
)

# Fit the GridSearchCV object to the training data
grid_search.fit(X_train, y_train)

# Print the best parameters found
print("Best Parameters:", grid_search.best_params_)

# Print the best score (accuracy) found
print("Best Score:", grid_search.best_score_)

# Get the best estimator (model)
best_gradient_boosting_model = grid_search.best_estimator_

# Evaluate the best model on the test set
y_pred = best_gradient_boosting_model.predict(X_test)
test_accuracy = accuracy_score(y_test, y_pred)
print("Test Accuracy:", test_accuracy)

# Print the classification report
report = classification_report(
    y_test, y_pred,
    target_names=le_outfit_category.classes_,
    zero_division=0
)
print("Classification Report:")
print(report)
```

```
Best Parameters: {'learning_rate': 0.01, 'max_depth': 4, 'n_estimators': 150,
'subsample': 1.0}
Best Score: 0.8153124999999999
Test Accuracy: 0.83375
Classification Report:
                  precision    recall  f1-score   support

 Business Attire       0.00      0.00      0.00        23
     Casual Wear       0.00      0.00      0.00        16
   Cultural Wear       0.74      0.57      0.65        40
    Evening Wear       0.00      0.00      0.00        24
           Other       0.85      0.94      0.89       575
Traditional Wear       0.79      0.87      0.82       122
```

```
          accuracy                           0.83       800
         macro avg       0.40      0.40      0.39       800
      weighted avg       0.77      0.83      0.80       800
```

[25]:
```python
# prompt: now we need to save this model as a pike file

import pickle

# Assuming your best model is 'best_logistic_model'
# Replace with your actual best model

# Save the model to a pickle file
with open('outfit_classification_model.pkl', 'wb') as file:
    pickle.dump(best_gradient_boosting_model, file)

print("Model saved as outfit_classification_model.pkl")
```

```
Model saved as outfit_classification_model.pkl
```

[61]:
```python
def predict_outfit(model, input_data):
    """
    Predict the outfit category and suggest an outfit description.

    Parameters:
    - model: Trained Random Forest Classifier.
    - input_data: A DataFrame containing the input features.

    Returns:
    - Suggested outfit description.
    """
    # Ensure input_data has the same features as X_train
    input_data_encoded = pd.get_dummies(input_data)
    input_data_encoded = input_data_encoded.reindex(columns=X_train.columns,␣
 ↪fill_value=0)

    # Predict the outfit category
    prediction_encoded = model.predict(input_data_encoded)
    predicted_category = le_outfit_category.
 ↪inverse_transform(prediction_encoded)

    # Filter the original data to find the outfit descriptions in that category
    matching_outfits = df[df['Outfit Category'] == predicted_category[0]]

    if not matching_outfits.empty:
        # Choose a random outfit description from the category
        outfit_description = matching_outfits['Outfit'].sample(1).values[0]
    else:
```

```
        outfit_description = "No matching outfit found."

    return outfit_description
```

[62]:
```
# prompt: load the pkl file

import pickle

# Load the saved model from the pickle file
with open('outfit_classification_model.pkl', 'rb') as file:
    loaded_model = pickle.load(file)

print("Model loaded successfully!")

# Now you can use 'loaded_model' to make predictions
# ... your code to use the loaded model ...
```

```
Model loaded successfully!
```

[67]:
```
# Example input data
new_data = pd.DataFrame({
    'Country': ['Japan'],
    'City': ['Osaka'],
    'Culture': ['Traditional Japanese'],
    'Event': ['Wedding'],
    'Preferred Color': ['Green'],
    'Preferred Style': ['Traditional']
})

# Get the outfit prediction
suggested_outfit = predict_outfit(loaded_model, new_data)
print(f"Suggested Outfit: {suggested_outfit}")
```

```
Suggested Outfit: Solid tsukesage kimono
```