دانشکده فنی و مهندسی
گروه مهندسی کامپیوتر

پروژه داده کاوی

نام دانشجو
صادق رجائی
شماره دانشجویی
40090992513

استاد درس
دکتر فاطمه باقری

فهرست مطالب

نیمسال اول سال تحصیلی 1403-1404

1- مجموعه داده

مجموعه داده‌ها به موضوع تاثیر دورکاری بر سلامت روان می‌پردازد. این داده‌ها مربوط به پاسخ‌های افرادی است که در بازه‌های زمانی مختلف و در مکان‌های گوناگون به سوالاتی پیرامون سلامت روان و دورکاری پاسخ داده‌اند. این مجموعه شامل 5000 رکورد و 20 ویژگی است. ویژگی‌های این داده شامل عواملی مانند سن، جنسیت، وضعیت کاری، تأثیرات دورکاری بر سلامت روان، و میزان رضایت شغلی هستند. ویژگی‌ها دارای مقادیر null نیستند. در این مجموعه داده نقاط پرت نیز مشاهده می‌شود که ممکن است نیاز به توجه ویژه در فرآیند تحلیل داده‌ها داشته باشد.

1-1- مشخصات مجموعه داده

جدول توصیف مجموعه داده

ردیف	ویژگی	توصیف
1	Employee ID	شناسه منحصر به فرد هر کارمند
2	Age	سن هر کارمند
3	Gender	جنسیت هر کارمند
4	Job Role	موقعیت کاری هر کارمند
5	Industry	حوزه شغلی هر کارمند
6	Years of Experience	سابقه کار هر کارمند
7	Work Location	محل کار هر کارمند
8	Hours Worked Per Week	ساعات کاری هفتگی هر کارمند
9	Number of Virtual Meetings	تعداد ملاقات مجازی برقرار شده کارمند
10	Work Life Balance Rating	وضعیت تعادل بین زندگی شغلی و اجتماعی کارمند
11	Stress Level	میزان استرس هر کارمند
12	Mental Health Condition	وضعیت سلامت روان کارمند
13	Access to Mental Health Resources	آیا کارمند به امکانات بهبود سلامت روان دسترسی دارد؟
14	Productivity Change	تغیر تغییر کارآمدی هر کارمند
15	Social Isolation Rating	شدت انزوای محیط هر کارمند
16	Satisfaction with Remote Work	میزان رضایت از دور کاری برای هر کارمند

17	Company Support for Remote Work	میزان همکاری شرکت برای فراهم شدن دورکاری
18	Physical Activity	میزان فعالیت بدنی هر کارمند
19	Sleep Quality	وضعیت خواب هر کارمند
20	Region	منطقه جغرافیایی

1-2- نمونه هایی از مجموعه داده

Employee ID	Age	Gender	Job Role	Industry	Years of Experience
EMP0002	40	Female	Data Scientist	IT	3
EMP0135	60	Male	Marketing	Finance	2
EMP0137	44	Male	Data Scientist	Healthcare	14
EMP4891	50	Male	HR	Education	13
EMP4887	47	Female	Designer	Finance	10
EMP4910	34	Non-binary	Project Manager	Consulting	7

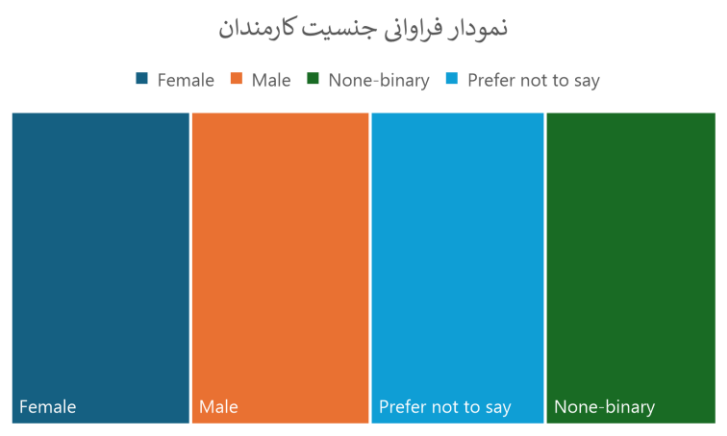
1-3- مشخصات اولیه ویژگی های مجموعه داده

ویژگی	تعداد	ماکزیمم مقدار	مینیمم مقدار	مقدار میانه	تعداد داده های ناموجود
Age	5000	60	22	41	0
Years of Experience	5000	35	1	18	0
Hours Worked Per Week	5000	60	20	40	0
Hours Worked Per Week	5000	15	0	8	0
Social Isolation Rating	5000	5	1	3	0
Company Support for Remote Work	5000	5	1	3	0
Gender	5000	Female → 1274 مورد Male → 1270 مورد Non-binary → 1214 مورد Prefer not to say → 1242 مورد			0
Job Role	5000	Data Scientist → 696 مورد Designer → 723 مورد HR → 716 مورد			0

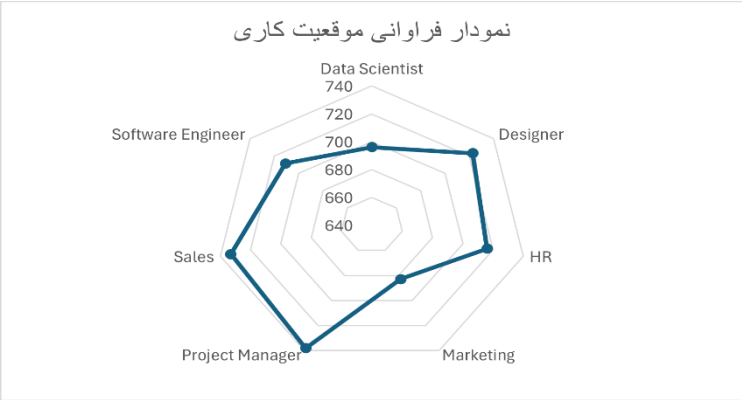
	Marketing → مورد 683 Project Manager → مورد 738 Sales → مورد 733 Software Engineer → مورد 711		
0	Consulting → مورد 680 Education → مورد 690 Finance → مورد 747 Healthcare → مورد 728 IT → مورد 746 Manufacturing → مورد 683 Retail → مورد 726	5000	Industry
0	Hybrid → مورد 1649 Onsite → مورد 1637 Remote → مورد 1714	5000	Work Location
0	High → مورد 1686 Low → مورد 1645 Medium → مورد 1669	5000	Stress Level
0	Anxiety → مورد 1278 Burnout → مورد 1280 Depression → مورد 1246 None → مورد 1196	5000	Mental Health Condition
0	Neutral → مورد 1648 Satisfied → مورد 1677 Unsatisfied → مورد 1675	5000	Satisfaction with Remote Work
0	Daily → مورد 1616 None → مورد 1629 Weekly → مورد 1755	5000	Physical Activity
0	Average → مورد 1628 Good → مورد 1687 Poor → مورد 1685	5000	Sleep Quality
0	Africa → مورد 860 Asia → مورد 829 Europe → مورد 840 North America → مورد 777 Oceania → مورد 867 South America → مورد 827	5000	Region

4-1- نمودارهای توصیف داده

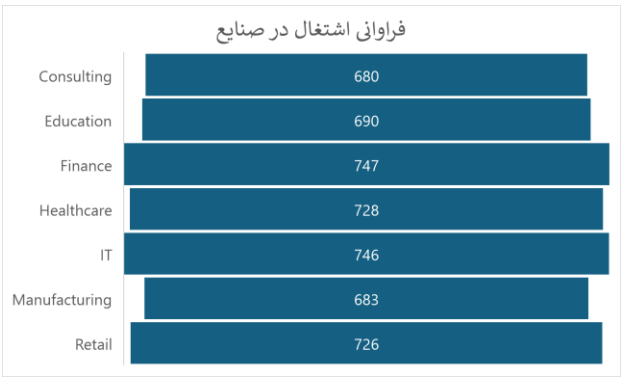
در این بخش، نمودارهای توصیف داده‌ها جهت بررسی فراوانی و توزیع مقادیر ویژگی های مختلف ارائه شدهاند. شکل ۱ فراوانی ویژگی جنسیت کارمندان را نشان می دهد، که شامل آمار متناسبی از جنسیت‌های متفاوت است. شکل ۲ به نحوه توزیع فراوانی موقعیت شغلی کارمندان اختصاص دارد که نمایانگر نسبت بیشتر مدیران پروژه می‌باشد. شکل ۳ و ۴ به توصیف فراوانی صنایعی که کارمندان در آن مشغول به کار هستند میپردازد. شکل ۵ نمایانگر توزیع فراوانی سه شیوه کاری از حیث محل کار می‌باشد که مشخصا با فاصله کمی تعداد موقعیت‌های دورکاری بیشتر می‌باشد. نمودارهای ۶، ۷ و ۸ به ترتیب توزیع گسسته سطح استرس، وضعیت روحی و روان و میزان رضایت از شیوه دورکاری پرداخته‌است.



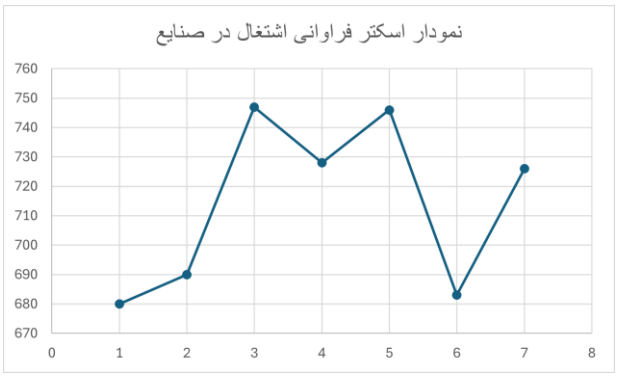
شکل 1: فراوانی ویژگی جنیست در مجموعه داده



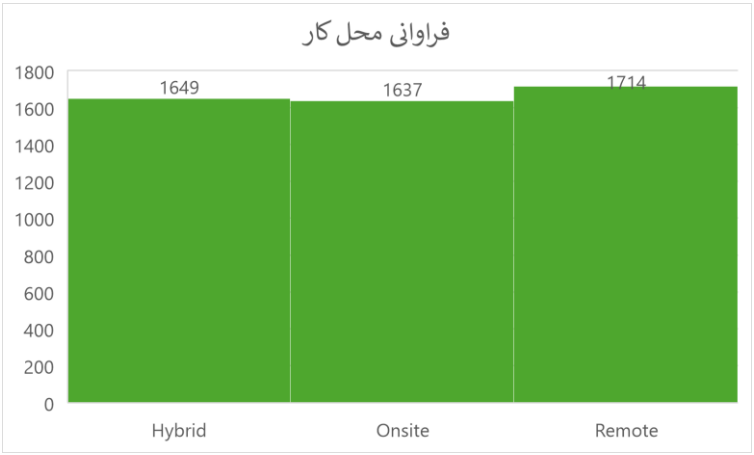
شکل 2: فراوانی موقعیت کاری در مجموعه داده



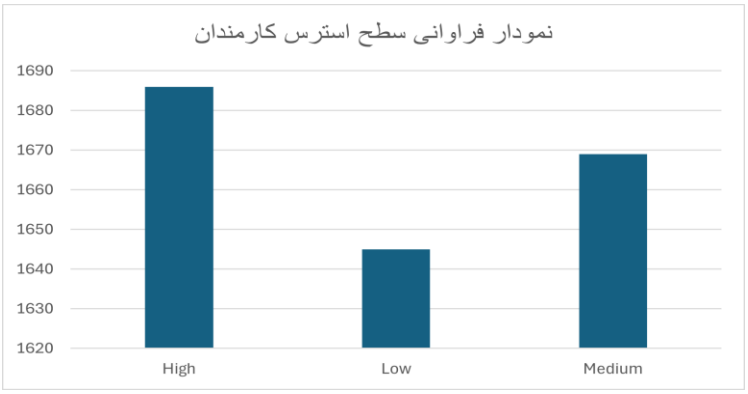
شکل 3: نمودار فراوانی اشتغال در صنایع



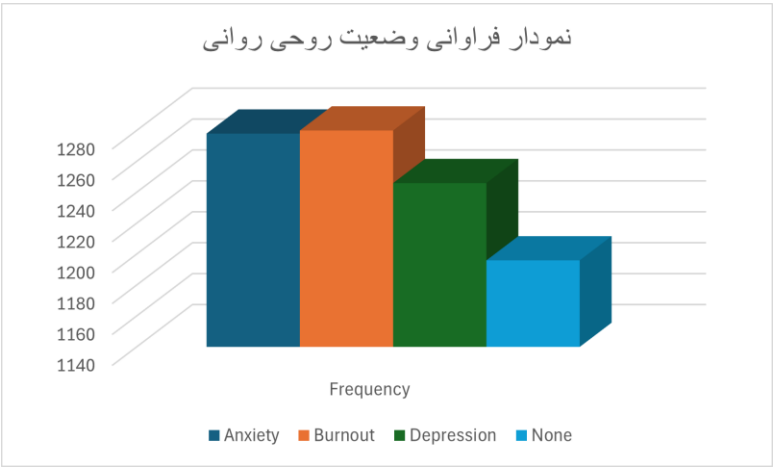
شکل 4: نمودار اسکتر اشتغال در صنایع



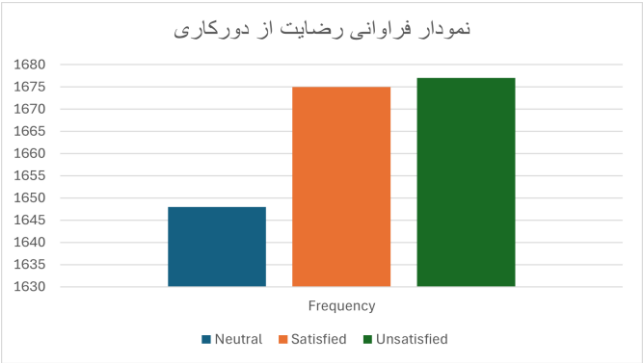
شکل 5: نمودار فراوانی محل کار



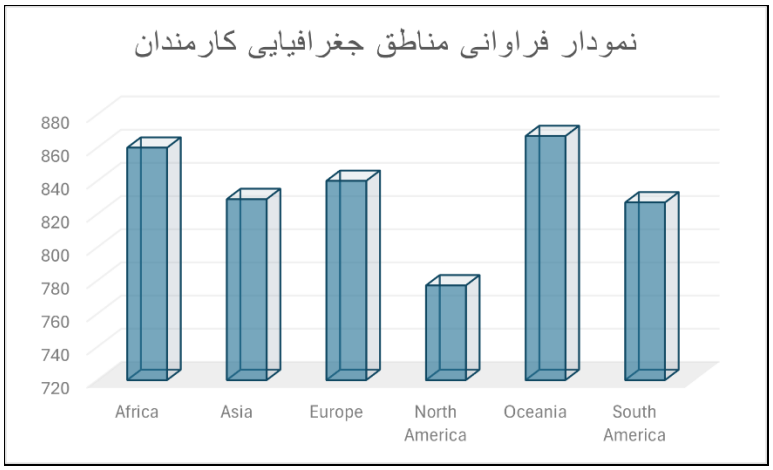
شکل 6: نمودار فراوانی سطح استرس کارمندان



شکل 7: نمودار فراوانی وضعیت روحی روانی



شکل 8: نمودار فراوانی رضایت از دورکاری



شکل 9: نمودار فراوانی مناطق جغرافیایی کارمندان

5-1- نقشه حرارتی

نقشه حرارتی زیر نشان‌دهنده همبستگی میان ویژگی‌های مختلف در مجموعه‌داده‌ای مرتبط با دورکاری و سلامت روان است. در این هیتمپ، هر خانه نمایانگر ضریب همبستگی بین دو ویژگی است که می‌تواند از -1 تا 1 متغیر باشد:

- همبستگی نزدیک به 1 نشان‌دهنده یک رابطه مثبت قوی است، به این معنا که افزایش یک ویژگی به طور قابل توجهی با افزایش ویژگی دیگر همراه است.
- همبستگی نزدیک به -1 نشان‌دهنده یک رابطه منفی قوی است، یعنی افزایش یک ویژگی با کاهش محسوس ویژگی دیگر همراه می‌شود.
- همبستگی نزدیک به 0 نشان‌دهنده نبود رابطه معنادار بین دو ویژگی است.

شکل 10: نقشه حرارتی برای همه ویژگی های اثر دورکاری بر سلامت روان

2- پیش‌پردازش

- در مرحله اول، با استفاده از کتابخانه‌های ضروری شروع کردیم. برای کار با داده‌ها، از `pandas` و برای تحلیل آماری و نمایش تصویری از `numpy`، `seaborn` و `matplotlib` استفاده شد. ابتدا داده‌ها را با تابع `pd.read_csv()` از `pandas` بارگذاری کردیم تا فایل CSV را به یک DataFrame تبدیل کنیم. سپس، با استفاده از دستور `head()`، پنج ردیف اول داده را چاپ کردیم تا یک نگاه کلی به داده‌ها داشته باشیم. همچنین، برای شناسایی نوع هر ستون و تعداد مقادیر خالی، از دستور `info()` استفاده کردیم. این مرحله به ما کمک کرد تا مشخصات کلی داده‌ها را مشاهده کنیم و به‌طور دقیق بدانیم که در کدام بخش‌ها نیاز به اصلاح و پاکسازی داریم.

```
import pandas as pd

# Load the dataset
df = pd.read_csv('/kaggle/input/remote-work-and-mental-health/Impact_of_Remote_Work_on_Mental_Health.csv')

# Check the first few rows and basic info
print("First few rows of the dataset:")
print(df.head())
```

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv('/kaggle/input/remote-work-and-mental-health/Impact_of_Remote_Work_on_Mental_Health.csv')

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

# Select only numerical columns for correlation
numerical_df = df.select_dtypes(include='number')

# Calculate the correlation matrix
corr_matrix = numerical_df.corr()

# Set the figure size for better visibility
plt.figure(figsize=(10, 8))
```

- در مرحله دوم، مدیریت مقادیر خالی را شروع کردیم. وجود مقادیر خالی می‌تواند مشکلاتی در تحلیل داده‌ها و مدلسازی ایجاد کند، بنابراین باید این مقادیر را با دقت مدیریت کنیم. برای ستون‌های عددی که مقادیر خالی داشتند، از میانگین ستون‌ها استفاده کردیم تا مقادیر خالی را پر کنیم. برای این کار از تابع `fillna()` همراه با `mean()` استفاده کردیم. همچنین، برای ستون‌های دسته‌بندی‌شده، مقادیر خالی را با مد (مقداری که بیشترین تکرار را دارد) پر کردیم. این روش کمک می‌کند که داده‌ها همگن و منسجم شوند و از دست رفتن اطلاعات جلوگیری شود. این کار با استفاده از `df[column].mode()[0]` برای یافتن مد و `fillna()` برای جایگزینی مقادیر خالی انجام شد.

```
# Check for missing values
print("\nMissing values per column:")
print(df.isnull().sum())

# Fill missing values with mean (for numerical) or mode (for categorical)
df_filled = df.copy()  # Copy for comparison purposes
for column in df.columns:
    if df[column].dtype == 'object':
        df_filled[column].fillna(df[column].mode()[0], inplace=True)
    else:
        df_filled[column].fillna(df[column].mean(), inplace=True)

# Display sample before and after filling missing values
print("\nSample data before filling missing values:")
print(df.head())
print("\nSample data after filling missing values:")
print(df_filled.head())
```

- در مرحله سوم، به سراغ حذف ردیف‌های تکراری رفتیم. وجود داده‌های تکراری می‌تواند به طور غیرمستقیم بر دقت مدل‌های یادگیری ماشین تأثیر منفی بگذارد، زیرا این تکرارها ممکن است وزن ناعادلانه‌ای به برخی نمونه‌ها دهند. برای شناسایی ردیف‌های تکراری، از تابع `duplicated()` استفاده کردیم و سپس با استفاده از `drop_duplicates()` این ردیف‌ها را حذف کردیم. این کار نه تنها دقت تحلیل‌ها را افزایش می‌دهد، بلکه باعث کاهش حجم داده‌ها می‌شود و مدل‌های ما را کارآمدتر می‌کند.

```
# Check for duplicates
print(f"\nNumber of duplicate rows: {df_filled.duplicated().sum()}")

# Remove duplicate rows
df_no_duplicates = df_filled.drop_duplicates()

# Show sample data after removing duplicates
print("\nSample data after removing duplicates:")
print(df_no_duplicates.head())
```

- در مرحله چهارم، کدگذاری داده‌های دسته‌بندی‌شده انجام شد. برای کدگذاری این ستون‌ها از دو روش اصلی استفاده کردیم: `Label Encoding` و `One-Hot Encoding`. ستون‌هایی که فقط دو دسته داشتند (مانند بله/خیر)، با استفاده از `LabelEncoder`از کتابخانه `sklearn.preprocessing`به صورت عددی کدگذاری شدند. برای این کار، ابتدا یک نمونه از `LabelEncoder`ساختیم و سپس با استفاده از `fit_transform()` ستون موردنظر را به مقادیر عددی تبدیل کردیم. در مقابل، ستون‌هایی که بیش از دو دسته داشتند، با استفاده از `pd.get_dummies()` کدگذاری شدند. این روش به این صورت است که به هر دسته یک ستون جدید اختصاص می‌دهد و از مقادیر ۰ و ۱ برای نشان دادن حضور یا عدم حضور آن دسته در هر ردیف استفاده می‌کند.

```
from sklearn.preprocessing import LabelEncoder

# Identify categorical columns
categorical_columns = df_no_duplicates.select_dtypes(include='object').columns

# Apply Label Encoding on binary categorical columns
df_encoded = df_no_duplicates.copy()
label_encoders = {}
for col in categorical_columns:
    # Label Encoding for binary categories
    if df_encoded[col].nunique() == 2:
        le = LabelEncoder()
        df_encoded[col] = le.fit_transform(df_encoded[col])
        label_encoders[col] = le
    # One-Hot Encoding for non-binary categories
    else:
        df_encoded = pd.get_dummies(df_encoded, columns=[col], drop_first=True)

# Show sample data after encoding
print("\nSample data after encoding:")
print(df_encoded.head())
```

- در مرحله پنجم، به استانداردسازی داده‌های عددی پرداختیم تا همه ویژگی‌ها در یک مقیاس مشابه قرار گیرند. برای این کار از `StandardScaler` در کتابخانه `sklearn.preprocessing` استفاده کردیم. ابتدا یک نمونه از `StandardScaler`ساختیم و سپس با استفاده از `fit_transform()` روی ستون‌های عددی، داده‌ها را استاندارد کردیم. استانداردسازی باعث می‌شود که تمام مقادیر داده‌ها در محدوده میانگین صفر و انحراف معیار یک قرار بگیرند و تأثیر ویژگی‌های عددی بزرگ‌تر بر مدل‌های یادگیری ماشین کاهش یابد.

```
from sklearn.preprocessing import StandardScaler

# Select numerical columns
numerical_columns = df_encoded.select_dtypes(include='number').columns

# Apply standard scaling
scaler = StandardScaler()
df_scaled = df_encoded.copy()
df_scaled[numerical_columns] = scaler.fit_transform(df_encoded[numerical_columns])

# Show sample data after scaling
print("\nSample data after scaling:")
print(df_scaled.head())
```

- در نهایت، برای نمایش تفاوت‌های هر مرحله، نمونه‌هایی از داده‌های قبل و بعد از هر مرحله پیش‌پردازش را چاپ کردیم. این کار به ما امکان داد که تاثیر هر مرحله از پیش‌پردازش را به خوبی مشاهده کنیم و ببینیم چگونه داده‌ها با انجام این مراحل تمیز و آماده استفاده در مدل‌های یادگیری ماشین شدند. این نمایش نمونه‌ها با استفاده از `print()` به ما کمک کرد که به‌طور شفاف تغییرات در داده‌ها را درک کنیم و به استاد یا همکارانمان نشان دهیم که چگونه داده‌ها بهبود یافته‌اند و آماده‌ی تحلیل‌های دقیق‌تر و مدل‌سازی شده‌اند.

2-1- نمایش نمونه‌هایی از پیش‌پردازش

```
First few rows of the dataset:
  Employee_ID  Age      Gender          Job_Role    Industry  \
0    EMP0001   32  Non-binary                HR  Healthcare
1    EMP0002   40      Female     Data Scientist          IT
2    EMP0003   59  Non-binary  Software Engineer   Education
3    EMP0004   27        Male  Software Engineer     Finance
4    EMP0005   49        Male             Sales  Consulting

   Years_of_Experience Work_Location  Hours_Worked_Per_Week  \
0                   13       Hybrid                     47
1                    3       Remote                     52
2                   22       Hybrid                     46
3                   20       Onsite                     32
4                   32       Onsite                     35

   Number_of_Virtual_Meetings  Work_Life_Balance_Rating Stress_Level  \
0                           7                         2       Medium
1                           4                         1       Medium
2                          11                         5       Medium
3                           8                         4         High
4                          12                         2         High

  Mental_Health_Condition Access_to_Mental_Health_Resources  \
0             Depression                                No
1                Anxiety                                No
2                Anxiety                                No
3             Depression                               Yes
4                    NaN                               Yes

  Productivity_Change  Social_Isolation_Rating Satisfaction_with_Remote_Work  \
0            Decrease                        1                    Unsatisfied
1            Increase                        3                      Satisfied
2           No Change                        4                    Unsatisfied
3            Increase                        3                    Unsatisfied
4            Decrease                        3                    Unsatisfied

   Company_Support_for_Remote_Work Physical_Activity Sleep_Quality  \
0                                1           Weekly          Good
1                                2           Weekly          Good
2                                5              NaN          Poor
3                                3              NaN          Poor
4                                3           Weekly       Average

         Region
0        Europe
1          Asia
2  North America
3        Europe
4  North America
```

```
Basic info of the dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 20 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Employee_ID                       5000 non-null   object
 1   Age                               5000 non-null   int64
 2   Gender                            5000 non-null   object
 3   Job_Role                          5000 non-null   object
 4   Industry                          5000 non-null   object
 5   Years_of_Experience               5000 non-null   int64
 6   Work_Location                     5000 non-null   object
 7   Hours_Worked_Per_Week             5000 non-null   int64
 8   Number_of_Virtual_Meetings        5000 non-null   int64
 9   Work_Life_Balance_Rating          5000 non-null   int64
 10  Stress_Level                      5000 non-null   object
 11  Mental_Health_Condition           3804 non-null   object
 12  Access_to_Mental_Health_Resources 5000 non-null   object
 13  Productivity_Change               5000 non-null   object
 14  Social_Isolation_Rating           5000 non-null   int64
 15  Satisfaction_with_Remote_Work     5000 non-null   object
 16  Company_Support_for_Remote_Work   5000 non-null   int64
 17  Physical_Activity                 3371 non-null   object
 18  Sleep_Quality                     5000 non-null   object
 19  Region                            5000 non-null   object
dtypes: int64(7), object(13)
memory usage: 781.4+ KB
None

Missing values per column:
Employee_ID                            0
Age                                    0
Gender                                 0
Job_Role                               0
Industry                               0
Years_of_Experience                    0
Work_Location                          0
Hours_Worked_Per_Week                  0
Number_of_Virtual_Meetings             0
Work_Life_Balance_Rating               0
Stress_Level                           0
Mental_Health_Condition             1196
Access_to_Mental_Health_Resources      0
Productivity_Change                    0
Social_Isolation_Rating                0
Satisfaction_with_Remote_Work          0
Company_Support_for_Remote_Work        0
Physical_Activity                   1629
Sleep_Quality                          0
Region                                 0
dtype: int64
```

```
Sample data after filling missing values:
  Employee_ID  Age      Gender          Job_Role    Industry  \
0     EMP0001   32  Non-binary                HR  Healthcare
1     EMP0002   40      Female     Data Scientist          IT
2     EMP0003   59  Non-binary  Software Engineer   Education
3     EMP0004   27        Male  Software Engineer     Finance
4     EMP0005   49        Male              Sales  Consulting

   Years_of_Experience Work_Location  Hours_Worked_Per_Week  \
0                   13        Hybrid                     47
1                    3        Remote                     52
2                   22        Hybrid                     46
3                   20        Onsite                     32
4                   32        Onsite                     35

   Number_of_Virtual_Meetings  Work_Life_Balance_Rating Stress_Level  \
0                           7                         2       Medium
1                           4                         1       Medium
2                          11                         5       Medium
3                           8                         4         High
4                          12                         2         High

  Mental_Health_Condition Access_to_Mental_Health_Resources  \
0              Depression                                No
1                 Anxiety                                No
2                 Anxiety                                No
3              Depression                               Yes
4                 Burnout                               Yes

  Productivity_Change  Social_Isolation_Rating Satisfaction_with_Remote_Work  \
0            Decrease                       1                   Unsatisfied
1            Increase                       3                     Satisfied
2           No Change                       4                   Unsatisfied
3            Increase                       3                   Unsatisfied
4            Decrease                       3                   Unsatisfied

   Company_Support_for_Remote_Work Physical_Activity Sleep_Quality  \
0                                1           Weekly          Good
1                                2           Weekly          Good
2                                5           Weekly          Poor
3                                3           Weekly          Poor
4                                3           Weekly       Average

         Region
0        Europe
1          Asia
2  North America
3        Europe
4  North America

Number of duplicate rows: 0
```

```
Sample data after encoding:
   Age  Years_of_Experience  Hours_Worked_Per_Week  \
0   32                   13                     47
1   40                    3                     52
2   59                   22                     46
3   27                   20                     32
4   49                   32                     35

   Number_of_Virtual_Meetings  Work_Life_Balance_Rating  \
0                           7                         2
1                           4                         1
2                          11                         5
3                           8                         4
4                          12                         2

   Access_to_Mental_Health_Resources  Social_Isolation_Rating  \
0                                  0                        1
1                                  0                        3
2                                  0                        4
3                                  1                        3
4                                  1                        3

   Company_Support_for_Remote_Work  Physical_Activity  Employee_ID_EMP0002  \
0                                1                  1                False
1                                2                  1                 True
2                                5                  1                False
3                                3                  1                False
4                                3                  1                False

   ...  Productivity_Change_No Change  \
0  ...                          False
1  ...                          False
2  ...                           True
3  ...                          False
4  ...                          False

   Satisfaction_with_Remote_Work_Satisfied  \
0                                    False
1                                     True
2                                    False
3                                    False
4                                    False

   Satisfaction_with_Remote_Work_Unsatisfied  Sleep_Quality_Good  \
0                                       True               True
1                                      False               True
2                                       True              False
3                                       True              False
4                                       True              False

   Sleep_Quality_Poor  Region_Asia  Region_Europe  Region_North America  \
0               False        False           True                 False
1               False         True          False                 False
2                True        False          False                  True
3                True        False           True                 False
4               False        False          False                  True
```

3- پردازش و اجرای الگوریتم‌ها

برای اجرای درخت تصمیم روی دیتاست مربوط به تأثیر کار از راه دور بر سلامت روان، ابتدا باید داده‌ها را پردازش کرد. این فرآیند شامل حذف مقادیر گمشده، کدگذاری متغیرهای دسته‌ای و نرمال‌سازی ویژگی‌های عددی است. در این مرحله، مقادیر گمشده در متغیرهای عددی با میانگین و در متغیرهای دسته‌ای با مد پر می‌شوند. سپس، متغیر هدف که در اینجا «شرایط سلامت روان» است، به دسته‌های عددی کدگذاری می‌شود. پس از آن، داده‌ها به ویژگی‌ها و متغیر هدف تقسیم می‌شوند و مجموعه داده به دو بخش آموزش و آزمون تقسیم می‌گردد.

پس از آماده‌سازی داده‌ها، مدل درخت تصمیم با استفاده از مجموعه آموزشی آموزش داده می‌شود. درخت تصمیم بر اساس ویژگی‌های موجود در داده‌ها، ساختاری را ایجاد می‌کند که می‌تواند به پیش‌بینی وضعیت سلامت روان افراد بر اساس متغیرهای مختلف مانند سن، جنسیت، و میزان کارکردی آنها بپردازد. پس از آموزش مدل، با استفاده از مجموعه آزمون، پیش‌بینی‌ها انجام می‌شود و دقت مدل با محاسبه معیارهایی مانند دقت و گزارش طبقه‌بندی ارزیابی می‌شود. همچنین می‌توان قوانین استخراج‌شده از درخت تصمیم را برای درک بهتر ویژگی‌های کلیدی مؤثر بر پیش‌بینی‌ها مشاهده کرد.

3-1- آموزش Classifier با استفاده از درخت تصمیم

ابتدا داده‌ها بارگذاری و پیش‌پردازش شده‌اند. در مرحله پیش‌پردازش، مقادیر گمشده به وسیله مقدار متداول هر ویژگی پر شده‌اند و همچنین ویژگی «کد شغلی (Employee_ID)» حذف شده است، زیرا به عنوان یک ویژگی شناسایی نیاز نیست. سپس ویژگی «شرایط سلامت روان (Mental_Health_Condition)» به صورت عددی کدگذاری شده و دیگر ویژگی‌های دسته‌ای نیز با استفاده از تکنیک‌های کدگذاری متناسب (Label Encoding و One-Hot Encoding) تبدیل شده‌اند. پس از آن، مقیاس ویژگی‌های عددی با استفاده از **StandardScaler** نرمال‌سازی شده است تا مدل بهتر بتواند داده‌ها را تحلیل کند.

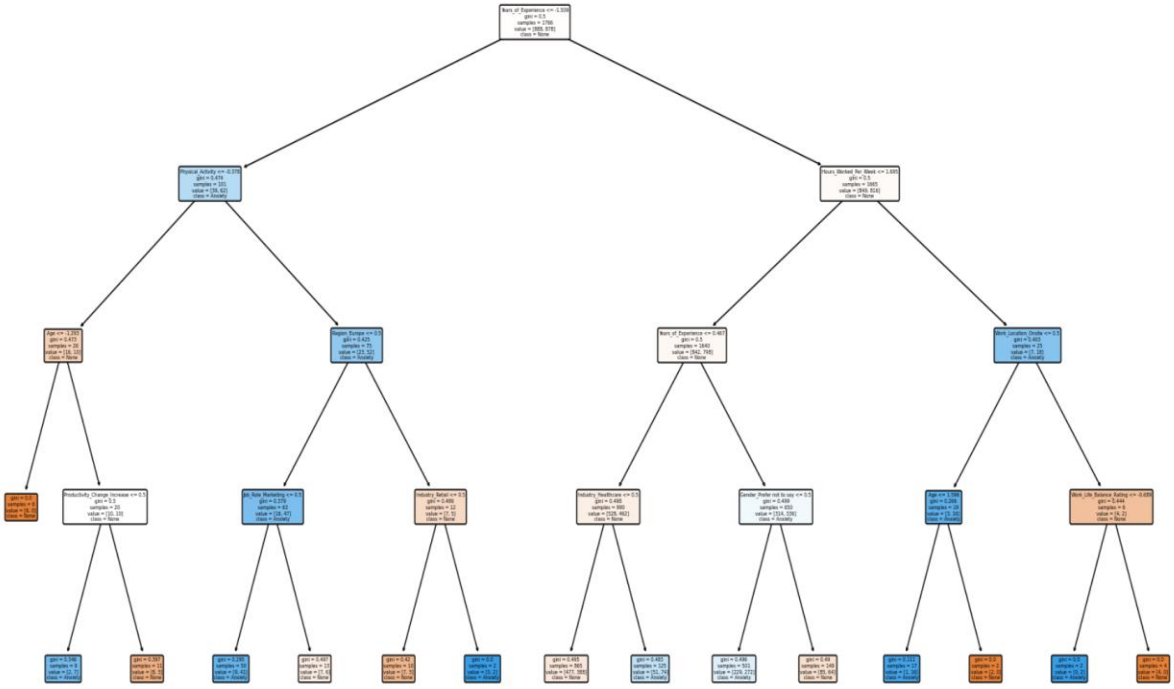
پس از آماده‌سازی داده‌ها، مدل درخت تصمیم با استفاده از **DecisionTreeClassifier** ایجاد و آموزش داده می‌شود. داده‌ها به دو مجموعه آموزشی و آزمایشی تقسیم می‌شوند و مدل بر روی مجموعه آموزشی آموزش داده می‌شود. سپس مدل بر روی مجموعه آزمایشی پیش‌بینی انجام می‌دهد و دقت آن با استفاده از معیارهای مختلف از جمله دقت کلی و گزارش طبقه‌بندی ارزیابی می‌شود. در انتها، درخت تصمیم ترسیم شده و قوانین استخراج‌شده از آن نمایش داده می‌شود. این قوانین به شفاف‌سازی روابط میان ویژگی‌ها و وضعیت سلامت روان کمک می‌کنند و می‌توانند در تصمیم‌گیری‌های بالینی و مدیریتی مفید باشند.

```python
# 8. Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# 9. Initialize and train the Decision Tree model
model = DecisionTreeClassifier(max_depth=4, random_state=42)
model.fit(X_train, y_train)

# 10. Make predictions and evaluate the model
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy of Decision Tree: {accuracy * 100:.2f}%")
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# 11. Display the Decision Tree
plt.figure(figsize=(20, 10))
plot_tree(model, feature_names=X.columns, class_names=['None', 'Anxiety', 'Depression'], filled=True, rounded=True)
plt.show()
```



3-2- قوانین استخراج شده

قوانین استخراجی از درخت تصمیم، به وضوح الگوهای موجود در داده‌ها را برای پیش‌بینی شرایط سلامت روان نشان می‌دهند. این قوانین، به شکل سلسله‌مراتبی و با استفاده از ویژگی‌های مختلف، به تجزیه و تحلیل عوامل مؤثر بر وضعیت سلامت روان افراد کمک می‌کنند. به عنوان مثال، در اولین قانون، اگر سال‌های تجربه کاری کمتر از -1.51 باشد، و فعالیت بدنی نیز کمتر از -0.38 باشد، سن فرد (Age) نقش مهمی ایفا می‌کند. اگر سن زیر -1.29 باشد، مدل پیش‌بینی می‌کند که فرد به وضعیت سلامت روان «اضطراب (class: 1.0)» دچار است. اما اگر سن بیشتر از -1.29 باشد و تغییر در بهره‌وری (Productivity Change)به صورت افزایش باشد، پیش‌بینی می‌شود که فرد به وضعیت «افسردگی (class: 2.0)» مبتلا خواهد شد.

در ادامه، قوانین نشان می‌دهند که چطور ویژگی‌هایی مانند مکان کار(Work Location) ، ساعات کاری در هفته (Hours Worked Per Week)، و نقش شغلی (Job Role) بر نتایج تأثیر دارند. برای نمونه، اگر سال‌های تجربه کاری بیشتر از -1.51 باشد و ساعات کاری در هفته کمتر از 1.69 باشد، جنسیت فرد (Gender) می‌تواند در تعیین وضعیت سلامت روان او تأثیرگذار باشد. همچنین، این قوانین می‌توانند به شناسایی گروه‌های پرخطر در محیط‌های کاری کمک کنند. برای مثال، در یک سناریو دیگر، اگر فردی در صنعت «بهداشت و درمان (Industry: Healthcare)» کار کند و سال‌های تجربه کاری‌اش کمتر از 0.47 باشد، مدل به احتمال زیاد پیش‌بینی می‌کند که او به وضعیت «افسردگی (class: 2.0)» مبتلا خواهد شد. این نوع اطلاعات می‌تواند به کارفرمایان و مشاوران کمک کند تا اقدامات مناسب‌تری برای بهبود سلامت روان کارکنان خود انجام دهند.

```
# 12. Extract and print rules from the Decision Tree
tree_rules = export_text(model, feature_names=list(X.columns))
print("\nExtracted Rules from the Decision Tree:\n")
print(tree_rules)
```

```
Extracted Rules from the Decision Tree:

|--- Years_of_Experience <= -1.51
|   |--- Physical_Activity <= -0.38
|   |   |--- Age <= -1.29
|   |   |   |--- class: 1.0
|   |   |--- Age >  -1.29
|   |   |   |--- Productivity_Change_Increase <= 0.50
|   |   |   |   |--- class: 2.0
|   |   |   |--- Productivity_Change_Increase >  0.50
|   |   |   |   |--- class: 1.0
|   |--- Physical_Activity >  -0.38
|   |   |--- Region_Europe <= 0.50
|   |   |   |--- Job_Role_Marketing <= 0.50
|   |   |   |   |--- class: 2.0
|   |   |   |--- Job_Role_Marketing >  0.50
|   |   |   |   |--- class: 1.0
|   |   |--- Region_Europe >  0.50
|   |   |   |--- Industry_Retail <= 0.50
|   |   |   |   |--- class: 1.0
|   |   |   |--- Industry_Retail >  0.50
|   |   |   |   |--- class: 2.0
|--- Years_of_Experience >  -1.51
|   |--- Hours_Worked_Per_Week <= 1.69
|   |   |--- Years_of_Experience <= 0.47
|   |   |   |--- Industry_Healthcare <= 0.50
|   |   |   |   |--- class: 1.0
|   |   |   |--- Industry_Healthcare >  0.50
|   |   |   |   |--- class: 2.0
|   |   |--- Years_of_Experience >  0.47
|   |   |   |--- Gender_Prefer not to say <= 0.50
|   |   |   |   |--- class: 2.0
|   |   |   |--- Gender_Prefer not to say >  0.50
|   |   |   |   |--- class: 1.0
|   |--- Hours_Worked_Per_Week >  1.69
|   |   |--- Work_Location_Onsite <= 0.50
|   |   |   |--- Age <= 1.60
|   |   |   |   |--- class: 2.0
|   |   |   |--- Age >  1.60
|   |   |   |   |--- class: 1.0
|   |   |--- Work_Location_Onsite >  0.50
|   |   |   |--- Work_Life_Balance_Rating <= -0.69
|   |   |   |   |--- class: 2.0
|   |   |   |--- Work_Life_Balance_Rating >  -0.69
|   |   |   |   |--- class: 1.0
```

3-3- تقسیم مجموعه داده به داده‌های آموزش و داده‌های آزمون

داده‌ها از یک دیتاست مربوط به تأثیر کار از راه دور بر سلامت روان استخراج شده‌اند و پس از بارگذاری، به دو مجموعه آموزش و آزمون تقسیم شده‌اند. این تقسیم به نسبت ۷۰ به ۳۰ انجام شده است، به این معنی که ۷۰ درصد از داده‌ها به عنوان مجموعه آموزش (X_train و y_train) و ۳۰ درصد به عنوان مجموعه آزمون (X_test و y_test) در نظر گرفته شده است.

مجموعه آموزش شامل ویژگی‌های مختلف مانند سال‌های تجربه، فعالیت بدنی، سن، تغییرات تولیدی، و سایر متغیرهای مربوط به شغل و سلامت روان است. مدل تصمیم‌گیری (درخت تصمیم) با استفاده از این داده‌ها آموزش می‌بیند تا الگوهای مربوط به وضعیت سلامت روان افراد را شناسایی کند. سپس، با استفاده از مجموعه آزمون، عملکرد مدل ارزیابی می‌شود. این مرحله شامل پیش‌بینی وضعیت سلامت روان (Mental_Health_Condition) بر اساس ویژگی‌های موجود در مجموعه آزمون و مقایسه نتایج پیش‌بینی شده با مقادیر واقعی است. این فرایند به ما این امکان را می‌دهد که بفهمید مدل چقدر خوب عمل می‌کند و آیا می‌تواند به درستی وضعیت سلامت روان را پیش‌بینی کند یا خیر.

```python
# Separate features and target (you need to specify the target column; assume 'Mental_Health' for this example)
X = df.drop(columns=['Mental_Health'])  # Drop target column
y = df['Mental_Health']  # Target variable

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

3-4- اجرای الگوریتم Random Forest

الگوریتم **Random Forest** به عنوان یک روش پیشرفته یادگیری ماشین برای پیش‌بینی وضعیت سلامت روانی افراد در نتیجه کار از راه دور پیاده‌سازی شده است. این الگوریتم بر پایه مجموعه‌ای از درخت‌های تصمیم‌گیری کار می‌کند و از رأی‌گیری بین این درخت‌ها برای ارائه پیش‌بینی نهایی استفاده می‌کند. در ابتدا، داده‌ها به دو بخش آموزشی و آزمایشی تقسیم می‌شوند، به طوری که 70 درصد داده‌ها برای آموزش و 30 درصد برای ارزیابی مدل در نظر گرفته می‌شوند. این تقسیم‌بندی به مدل این امکان را می‌دهد تا بر اساس داده‌های آموزشی یاد بگیرد و سپس بر روی داده‌های ناشناخته (آزمایشی) عملکرد خود را ارزیابی کند.

اجرای الگوریتم Random Forest در کد شامل چند مرحله کلیدی است. ابتدا، مدل با استفاده از `RandomForestClassifier` از کتابخانه `sklearn` ایجاد می‌شود، که تعداد درخت‌ها (n_estimators) به 100 و حالت تصادفی (random_state) به 42 تنظیم شده است. سپس با استفاده از داده‌های آموزشی `X_train` و `y_train`، مدل آموزش داده می‌شود. پس از آموزش، پیش‌بینی‌هایی بر اساس داده‌های آزمایشی (`X_test`) انجام می‌شود و دقت مدل با استفاده از معیار **دقت** (accuracy)و **گزارش طبقه‌بندی** (classification report) اندازه‌گیری می‌شود.

معیارهای ارزیابی مدل شامل دقت(Accuracy) ، دقت (Precision)، یادآوری (Recall) و نمره F1 (F1 Score) است.

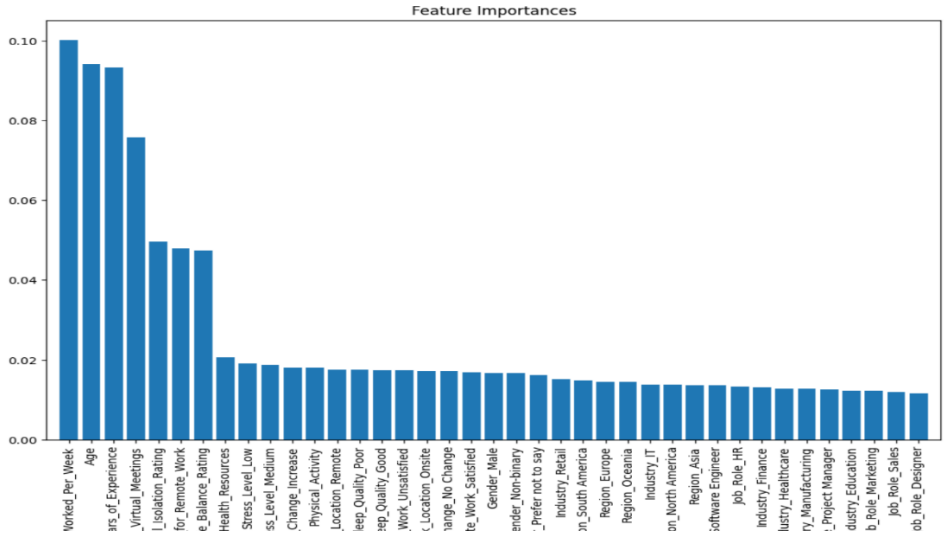
- **دقت** نشان‌دهنده درصد کل پیش‌بینی‌های صحیح نسبت به کل پیش‌بینی‌ها است.
- **دقت** به ما می‌گوید که از تمام پیش‌بینی‌های مثبت، چند مورد صحیح بوده‌اند، و **یادآوری** تعداد موارد مثبت صحیح را نسبت به کل موارد مثبت واقعی اندازه‌گیری می‌کند.
- نمره **F1** نیز یک معیار ترکیبی است که دقت و یادآوری را در نظر می‌گیرد و در مواقعی که عدم توازن بین کلاس‌ها وجود دارد، بسیار مفید است. این معیارها به تجزیه و تحلیل عملکرد مدل کمک می‌کنند و این امکان را می‌دهند که نقاط قوت و ضعف مدل خود را شناسایی کنیم.

در نهایت، این الگوریتم با نمایش اهمیت ویژگی‌ها (feature importance) این امکان را می‌دهد که بفهمیم کدام ویژگی‌ها بیشترین تأثیر را در پیش‌بینی‌ها دارند و بنابراین می‌توانند در تصمیم‌گیری‌های آینده نقش مهمی ایفا کنند.

```
Classification Report:
              precision    recall  f1-score   support

         1.0       0.54      0.57      0.56       390
         2.0       0.51      0.48      0.50       368

    accuracy                           0.53       758
   macro avg       0.53      0.53      0.53       758
weighted avg       0.53      0.53      0.53       758
```



3-5- ارزیابی و تفاوت کارآیی دو الگوریتم

مقایسه عملکرد الگوریتم‌های درخت تصمیم و جنگل تصادفی (Random Forest) در تحلیل داده‌ها نشان‌دهنده تفاوت‌های جالبی در معیارهای مختلف است. ابتدا به دقت (Accuracy) این دو الگوریتم اشاره می‌کنیم. جنگل تصادفی با دقت 52.77 درصد کمی بهتر از درخت تصمیم با دقت 51.00 درصد عمل کرده است. این تفاوت کوچک نشان می‌دهد که جنگل تصادفی توانسته است در پیش‌بینی نتایج کمی موفق‌تر عمل کند و می‌تواند به عنوان گزینه بهتری برای این مجموعه داده‌ها در نظر گرفته شود.

از نظر دقت پیش‌بینی(Precision) ، جنگل تصادفی برای کلاس 1.0 با دقت 0.54 در مقایسه با درخت تصمیم (0.52) اندکی بهتر عمل کرده است. برای کلاس 2.0 نیز جنگل تصادفی با دقت 0.51 عملکرد بهتری نسبت به درخت تصمیم (0.50) داشت. این امر نشان‌دهنده این است که جنگل تصادفی به طور کلی قادر به کاهش خطاهای پیش‌بینی در هر دو کلاس بوده و در شناسایی موارد صحیح بهتر عمل کرده است.

در مورد معیار فراخوانی (Recall)، هر دو الگوریتم برای کلاس 1.0 دارای مقدار مشابه 0.57 هستند، اما جنگل تصادفی برای کلاس 2.0 با فراخوانی 0.48 عملکرد بهتری نسبت به درخت تصمیم (0.45) داشته است. این نشان می‌دهد که جنگل تصادفی قادر است به طور موثرتری موارد مثبت کلاس 2.0 را شناسایی کند.

معیار F1-Score که ترکیبی از دقت و فراخوانی است، نیز در جنگل تصادفی نسبت به درخت تصمیم بهتر عمل کرده است . F1-Scoreبرای کلاس 1.0 در جنگل تصادفی 0.56 و برای کلاس 2.0 0.50 بوده که هر دو در مقایسه با درخت تصمیم (0.55 برای کلاس 1.0 و 0.47 برای کلاس 2.0) بهتر است. به‌طور کلی، معیارهای میانگین ماکرو و وزن‌دار نیز نشان‌دهنده برتری جنگل تصادفی در تمامی ابعاد است.

به‌طور خلاصه، با وجود این که هر دو الگوریتم عملکرد مشابهی دارند، جنگل تصادفی به‌طور کلی در شناسایی و پیش‌بینی دقیق‌تر از درخت تصمیم عمل کرده و می‌تواند به عنوان یک گزینه مناسب برای تحلیل این داده‌ها در نظر گرفته شود. این نتیجه می‌تواند ناشی از قابلیت جنگل تصادفی در مدیریت تنوع داده‌ها و بهبود قابلیت‌های پیش‌بینی باشد، به‌ویژه در مواجهه با داده‌های پیچیده و غیرخطی.

Metric	Decision Tree	Random Forest
Accuracy	51.00%	52.77%
Precision (1.0)	0.52	0.54
Recall (1.0)	0.57	0.57

F1-Score (1.0)	0.55	0.56
Support (1.0)	390	390
Precision (2.0)	0.50	0.51
Recall (2.0)	0.45	0.48
F1-Score (2.0)	0.47	0.50
Support (2.0)	368	368
Macro Average		
- Precision	0.51	0.53
- Recall	0.51	0.53
- F1-Score	0.51	0.53
Weighted Average		
- Precision	0.51	0.53
- Recall	0.51	0.53
- F1-Score	0.51	0.53