ML_Project

February 15, 2026

[ ]:
```
# Import Required Libraries
# ================================

# Basic libraries
import numpy as np
import pandas as pd

# Visualization libraries
import matplotlib.pyplot as plt
import seaborn as sns

# Sklearn - Preprocessing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder, LabelEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

# Sklearn - Models
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC

# Sklearn - Evaluation Metrics
from sklearn.metrics import (
    accuracy_score,
    confusion_matrix,
    classification_report,
    roc_auc_score,
    roc_curve
)
```

[8]:
```
#Load the dataset
data = pd.read_csv('Data/alzheimers_disease_data.csv')
print(data.head())
print(data.info())
print(data.describe())

# 2) Missing values (overall + per column)
```

```
missing_per_col = data.isna().sum().sort_values(ascending=False)
print("\nMissing values per column (top 15):")
print(missing_per_col.head(15))

# 3) Basic types
print("\nDtypes:\n", data.dtypes)

# 4) Quick numeric summary
display(data.describe(include="number").T)

# 5) Quick categorical summary (top categories)
cat_cols_guess = data.select_dtypes(include=["object", "category"]).columns
for c in cat_cols_guess[:10]:  # show first 10 only
    print(f"\nColumn: {c}")
    print(data[c].value_counts(dropna=False).head(10))
```

```
   PatientID  Age  Gender  Ethnicity  EducationLevel        BMI  Smoking  \
0       4751   73       0          0               2  22.927749        0
1       4752   89       0          0               0  26.827681        0
2       4753   73       0          3               1  17.795882        0
3       4754   74       1          0               1  33.800817        1
4       4755   89       0          0               0  20.716974        0

   AlcoholConsumption  PhysicalActivity  DietQuality  …  MemoryComplaints  \
0           13.297218          6.327112     1.347214  …                 0
1            4.542524          7.619885     0.518767  …                 0
2           19.555085          7.844988     1.826335  …                 0
3           12.209266          8.428001     7.435604  …                 0
4           18.454356          6.310461     0.795498  …                 0

   BehavioralProblems       ADL  Confusion  Disorientation  \
0                   0  1.725883          0               0
1                   0  2.592424          0               0
2                   0  7.119548          0               1
3                   1  6.481226          0               0
4                   0  0.014691          0               0

   PersonalityChanges  DifficultyCompletingTasks  Forgetfulness  Diagnosis  \
0                   0                          1              0          0
1                   0                          0              1          0
2                   0                          1              0          0
3                   0                          0              0          0
4                   1                          1              0          0

   DoctorInCharge
0       XXXConfid
1       XXXConfid
2       XXXConfid
```

```
3       XXXConfid
4       XXXConfid

[5 rows x 35 columns]
<class 'pandas.DataFrame'>
RangeIndex: 2149 entries, 0 to 2148
Data columns (total 35 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   PatientID                  2149 non-null   int64
 1   Age                        2149 non-null   int64
 2   Gender                     2149 non-null   int64
 3   Ethnicity                  2149 non-null   int64
 4   EducationLevel             2149 non-null   int64
 5   BMI                        2149 non-null   float64
 6   Smoking                    2149 non-null   int64
 7   AlcoholConsumption         2149 non-null   float64
 8   PhysicalActivity           2149 non-null   float64
 9   DietQuality                2149 non-null   float64
 10  SleepQuality               2149 non-null   float64
 11  FamilyHistoryAlzheimers    2149 non-null   int64
 12  CardiovascularDisease      2149 non-null   int64
 13  Diabetes                   2149 non-null   int64
 14  Depression                 2149 non-null   int64
 15  HeadInjury                 2149 non-null   int64
 16  Hypertension               2149 non-null   int64
 17  SystolicBP                 2149 non-null   int64
 18  DiastolicBP                2149 non-null   int64
 19  CholesterolTotal           2149 non-null   float64
 20  CholesterolLDL             2149 non-null   float64
 21  CholesterolHDL             2149 non-null   float64
 22  CholesterolTriglycerides   2149 non-null   float64
 23  MMSE                       2149 non-null   float64
 24  FunctionalAssessment       2149 non-null   float64
 25  MemoryComplaints           2149 non-null   int64
 26  BehavioralProblems         2149 non-null   int64
 27  ADL                        2149 non-null   float64
 28  Confusion                  2149 non-null   int64
 29  Disorientation             2149 non-null   int64
 30  PersonalityChanges         2149 non-null   int64
 31  DifficultyCompletingTasks  2149 non-null   int64
 32  Forgetfulness              2149 non-null   int64
 33  Diagnosis                  2149 non-null   int64
 34  DoctorInCharge             2149 non-null   str
dtypes: float64(12), int64(22), str(1)
memory usage: 587.7 KB
None
         PatientID          Age       Gender    Ethnicity  EducationLevel  \
```

count	2149.000000	2149.000000	2149.000000	2149.000000	2149.000000
mean	5825.000000	74.908795	0.506282	0.697534	1.286645
std	620.507185	8.990221	0.500077	0.996128	0.904527
min	4751.000000	60.000000	0.000000	0.000000	0.000000
25%	5288.000000	67.000000	0.000000	0.000000	1.000000
50%	5825.000000	75.000000	1.000000	0.000000	1.000000
75%	6362.000000	83.000000	1.000000	1.000000	2.000000
max	6899.000000	90.000000	1.000000	3.000000	3.000000

	BMI	Smoking	AlcoholConsumption	PhysicalActivity	\
count	2149.000000	2149.000000	2149.000000	2149.000000	
mean	27.655697	0.288506	10.039442	4.920202	
std	7.217438	0.453173	5.757910	2.857191	
min	15.008851	0.000000	0.002003	0.003616	
25%	21.611408	0.000000	5.139810	2.570626	
50%	27.823924	0.000000	9.934412	4.766424	
75%	33.869778	1.000000	15.157931	7.427899	
max	39.992767	1.000000	19.989293	9.987429	

	DietQuality	…	FunctionalAssessment	MemoryComplaints	\
count	2149.000000	…	2149.000000	2149.000000	
mean	4.993138	…	5.080055	0.208004	
std	2.909055	…	2.892743	0.405974	
min	0.009385	…	0.000460	0.000000	
25%	2.458455	…	2.566281	0.000000	
50%	5.076087	…	5.094439	0.000000	
75%	7.558625	…	7.546981	0.000000	
max	9.998346	…	9.996467	1.000000	

	BehavioralProblems	ADL	Confusion	Disorientation	\
count	2149.000000	2149.000000	2149.000000	2149.000000	
mean	0.156817	4.982958	0.205212	0.158213	
std	0.363713	2.949775	0.403950	0.365026	
min	0.000000	0.001288	0.000000	0.000000	
25%	0.000000	2.342836	0.000000	0.000000	
50%	0.000000	5.038973	0.000000	0.000000	
75%	0.000000	7.581490	0.000000	0.000000	
max	1.000000	9.999747	1.000000	1.000000	

	PersonalityChanges	DifficultyCompletingTasks	Forgetfulness	\
count	2149.000000	2149.000000	2149.000000	
mean	0.150768	0.158678	0.301536	
std	0.357906	0.365461	0.459032	
min	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	1.000000	
max	1.000000	1.000000	1.000000	

```
         Diagnosis
count  2149.000000
mean      0.353653
std       0.478214
min       0.000000
25%       0.000000
50%       0.000000
75%       1.000000
max       1.000000

[8 rows x 34 columns]

Missing values per column (top 15):
PatientID                    0
BehavioralProblems           0
CholesterolLDL               0
CholesterolHDL               0
CholesterolTriglycerides     0
MMSE                         0
FunctionalAssessment         0
MemoryComplaints             0
ADL                          0
DiastolicBP                  0
Confusion                    0
Disorientation               0
PersonalityChanges           0
DifficultyCompletingTasks    0
Forgetfulness                0
dtype: int64

Dtypes:
 PatientID                     int64
Age                            int64
Gender                         int64
Ethnicity                      int64
EducationLevel                 int64
BMI                          float64
Smoking                        int64
AlcoholConsumption           float64
PhysicalActivity             float64
DietQuality                  float64
SleepQuality                 float64
FamilyHistoryAlzheimers        int64
CardiovascularDisease          int64
Diabetes                       int64
Depression                     int64
HeadInjury                     int64
```

```
Hypertension                  int64
SystolicBP                    int64
DiastolicBP                   int64
CholesterolTotal            float64
CholesterolLDL              float64
CholesterolHDL              float64
CholesterolTriglycerides    float64
MMSE                        float64
FunctionalAssessment        float64
MemoryComplaints              int64
BehavioralProblems            int64
ADL                         float64
Confusion                     int64
Disorientation                int64
PersonalityChanges            int64
DifficultyCompletingTasks     int64
Forgetfulness                 int64
Diagnosis                     int64
DoctorInCharge                  str
dtype: object
```

	count	mean	std	min	\
PatientID	2149.0	5825.000000	620.507185	4751.000000	
Age	2149.0	74.908795	8.990221	60.000000	
Gender	2149.0	0.506282	0.500077	0.000000	
Ethnicity	2149.0	0.697534	0.996128	0.000000	
EducationLevel	2149.0	1.286645	0.904527	0.000000	
BMI	2149.0	27.655697	7.217438	15.008851	
Smoking	2149.0	0.288506	0.453173	0.000000	
AlcoholConsumption	2149.0	10.039442	5.757910	0.002003	
PhysicalActivity	2149.0	4.920202	2.857191	0.003616	
DietQuality	2149.0	4.993138	2.909055	0.009385	
SleepQuality	2149.0	7.051081	1.763573	4.002629	
FamilyHistoryAlzheimers	2149.0	0.252210	0.434382	0.000000	
CardiovascularDisease	2149.0	0.144253	0.351428	0.000000	
Diabetes	2149.0	0.150768	0.357906	0.000000	
Depression	2149.0	0.200558	0.400511	0.000000	
HeadInjury	2149.0	0.092601	0.289940	0.000000	
Hypertension	2149.0	0.148906	0.356079	0.000000	
SystolicBP	2149.0	134.264774	25.949352	90.000000	
DiastolicBP	2149.0	89.847836	17.592496	60.000000	
CholesterolTotal	2149.0	225.197519	42.542233	150.093316	
CholesterolLDL	2149.0	124.335944	43.366584	50.230707	
CholesterolHDL	2149.0	59.463533	23.139174	20.003434	
CholesterolTriglycerides	2149.0	228.281496	101.986721	50.407194	
MMSE	2149.0	14.755132	8.613151	0.005312	
FunctionalAssessment	2149.0	5.080055	2.892743	0.000460	
MemoryComplaints	2149.0	0.208004	0.405974	0.000000	

BehavioralProblems	2149.0	0.156817	0.363713	0.000000
ADL	2149.0	4.982958	2.949775	0.001288
Confusion	2149.0	0.205212	0.403950	0.000000
Disorientation	2149.0	0.158213	0.365026	0.000000
PersonalityChanges	2149.0	0.150768	0.357906	0.000000
DifficultyCompletingTasks	2149.0	0.158678	0.365461	0.000000
Forgetfulness	2149.0	0.301536	0.459032	0.000000
Diagnosis	2149.0	0.353653	0.478214	0.000000

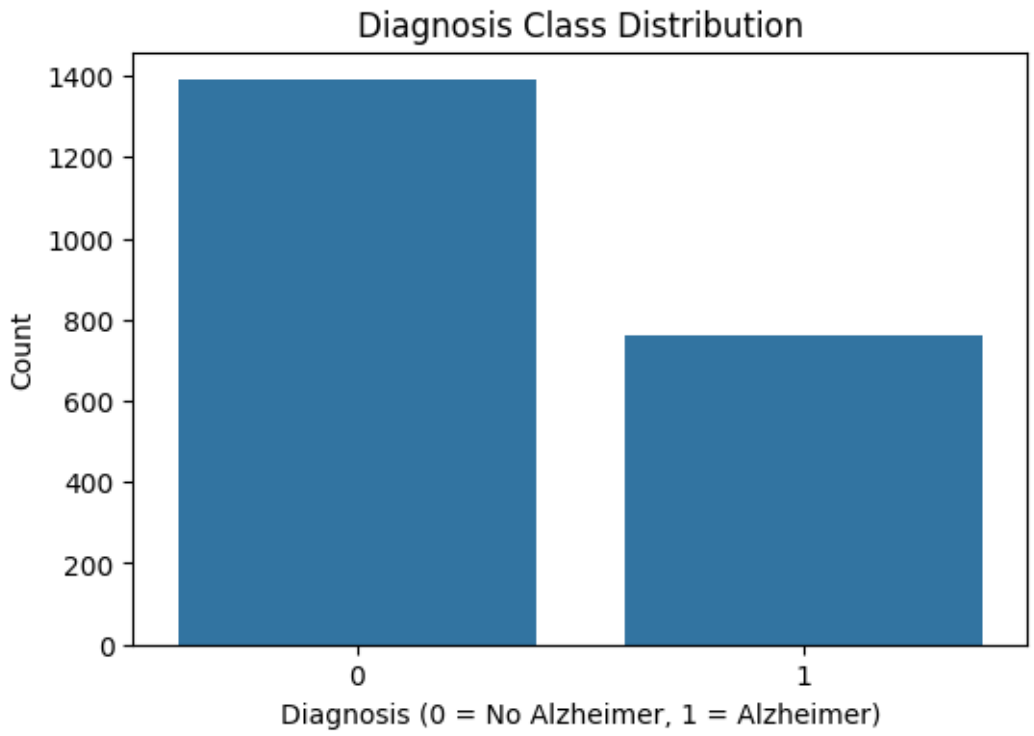
	25%	50%	75%	max
PatientID	5288.000000	5825.000000	6362.000000	6899.000000
Age	67.000000	75.000000	83.000000	90.000000
Gender	0.000000	1.000000	1.000000	1.000000
Ethnicity	0.000000	0.000000	1.000000	3.000000
EducationLevel	1.000000	1.000000	2.000000	3.000000
BMI	21.611408	27.823924	33.869778	39.992767
Smoking	0.000000	0.000000	1.000000	1.000000
AlcoholConsumption	5.139810	9.934412	15.157931	19.989293
PhysicalActivity	2.570626	4.766424	7.427899	9.987429
DietQuality	2.458455	5.076087	7.558625	9.998346
SleepQuality	5.482997	7.115646	8.562521	9.999840
FamilyHistoryAlzheimers	0.000000	0.000000	1.000000	1.000000
CardiovascularDisease	0.000000	0.000000	0.000000	1.000000
Diabetes	0.000000	0.000000	0.000000	1.000000
Depression	0.000000	0.000000	0.000000	1.000000
HeadInjury	0.000000	0.000000	0.000000	1.000000
Hypertension	0.000000	0.000000	0.000000	1.000000
SystolicBP	112.000000	134.000000	157.000000	179.000000
DiastolicBP	74.000000	91.000000	105.000000	119.000000
CholesterolTotal	190.252963	225.086430	262.031657	299.993352
CholesterolLDL	87.195798	123.342593	161.733733	199.965665
CholesterolHDL	39.095698	59.768237	78.939050	99.980324
CholesterolTriglycerides	137.583222	230.301983	314.839046	399.941862
MMSE	7.167602	14.441660	22.161028	29.991381
FunctionalAssessment	2.566281	5.094439	7.546981	9.996467
MemoryComplaints	0.000000	0.000000	0.000000	1.000000
BehavioralProblems	0.000000	0.000000	0.000000	1.000000
ADL	2.342836	5.038973	7.581490	9.999747
Confusion	0.000000	0.000000	0.000000	1.000000
Disorientation	0.000000	0.000000	0.000000	1.000000
PersonalityChanges	0.000000	0.000000	0.000000	1.000000
DifficultyCompletingTasks	0.000000	0.000000	0.000000	1.000000
Forgetfulness	0.000000	0.000000	1.000000	1.000000
Diagnosis	0.000000	0.000000	1.000000	1.000000

Column: DoctorInCharge
DoctorInCharge

```
XXXConfid    2149
Name: count, dtype: int64

/var/folders/_r/1mtvh9nn1ns69vtxm47ygl1c0000gn/T/ipykernel_10713/2441348960.py:1
9: Pandas4Warning: For backward compatibility, 'str' dtypes are included by
select_dtypes when 'object' dtype is specified. This behavior is deprecated and
will be removed in a future version. Explicitly pass 'str' to `include` to
select them, or to `exclude` to remove them and silence this warning.
See https://pandas.pydata.org/docs/user_guide/migration-3-strings.html#string-
migration-select-dtypes for details on how to write code that works with pandas
2 and 3.
  cat_cols_guess = data.select_dtypes(include=["object", "category"]).columns
```

[ ]:
```python
#Target distribution
# Count values
print(data["Diagnosis"].value_counts())

# Percentage distribution
print("\nPercentage Distribution:")
print(data["Diagnosis"].value_counts(normalize=True) * 100)

# Plot distribution
plt.figure(figsize=(6,4))
sns.countplot(x="Diagnosis", data=data)
plt.title("Diagnosis Class Distribution")
plt.xlabel("Diagnosis (0 = No Alzheimer, 1 = Alzheimer)")
plt.ylabel("Count")
plt.show()
```

```
Diagnosis
0    1389
1     760
Name: count, dtype: int64

Percentage Distribution:
Diagnosis
0    64.634714
1    35.365286
Name: proportion, dtype: float64
```

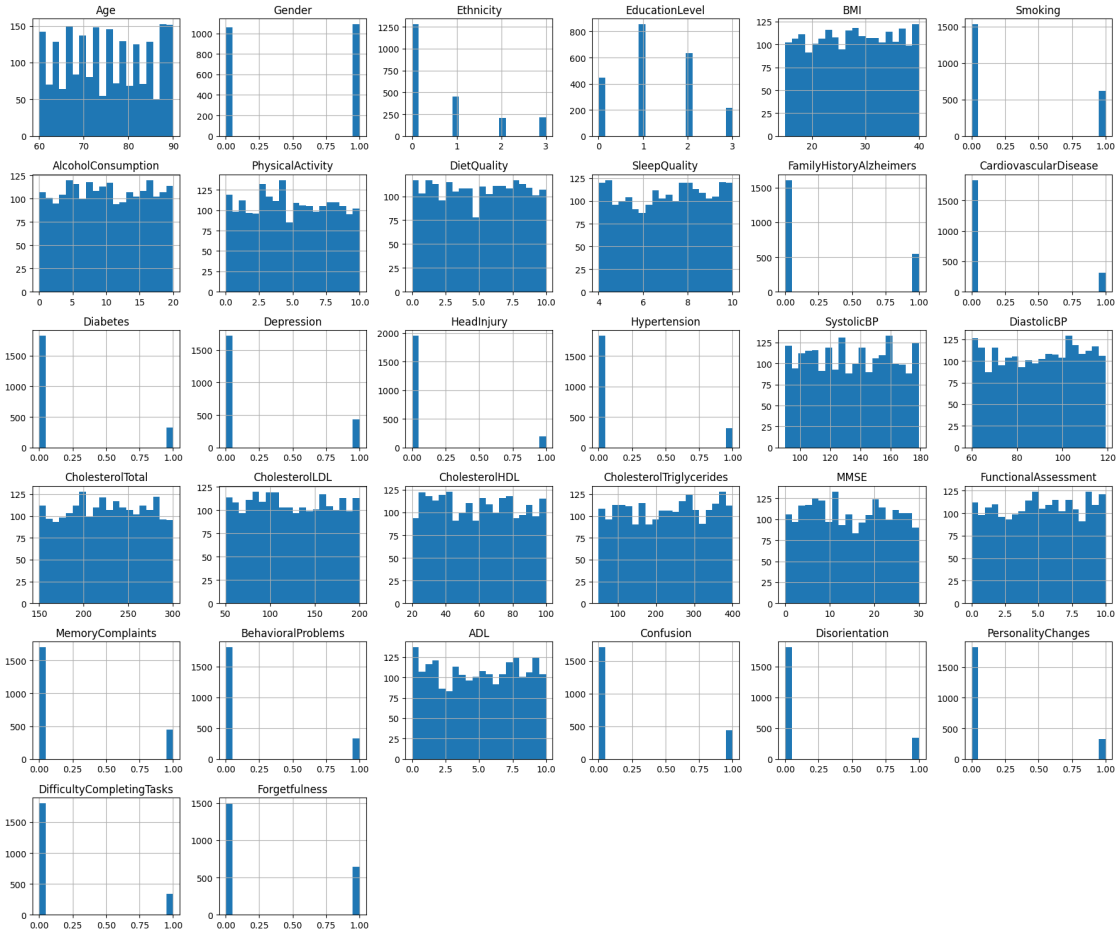


[25]:
```python
# Define Features (X) and Target (y)

print(data.columns.tolist())
# Define target
y = data["Diagnosis"]

# Define features
X = data.drop(columns=["Diagnosis","DoctorInCharge"])

print("X shape:", X.shape)
print("y shape:", y.shape)

print("Unique target values:", y.unique())
y = y.astype(str).str.strip().str.lower()

#check data

# 2. Numerical Feature Distribution

numerical_cols = X.select_dtypes(include=["int64", "float64"]).columns

print("Numerical Features:", numerical_cols)
X[numerical_cols].hist(figsize=(18,15), bins=20)
```

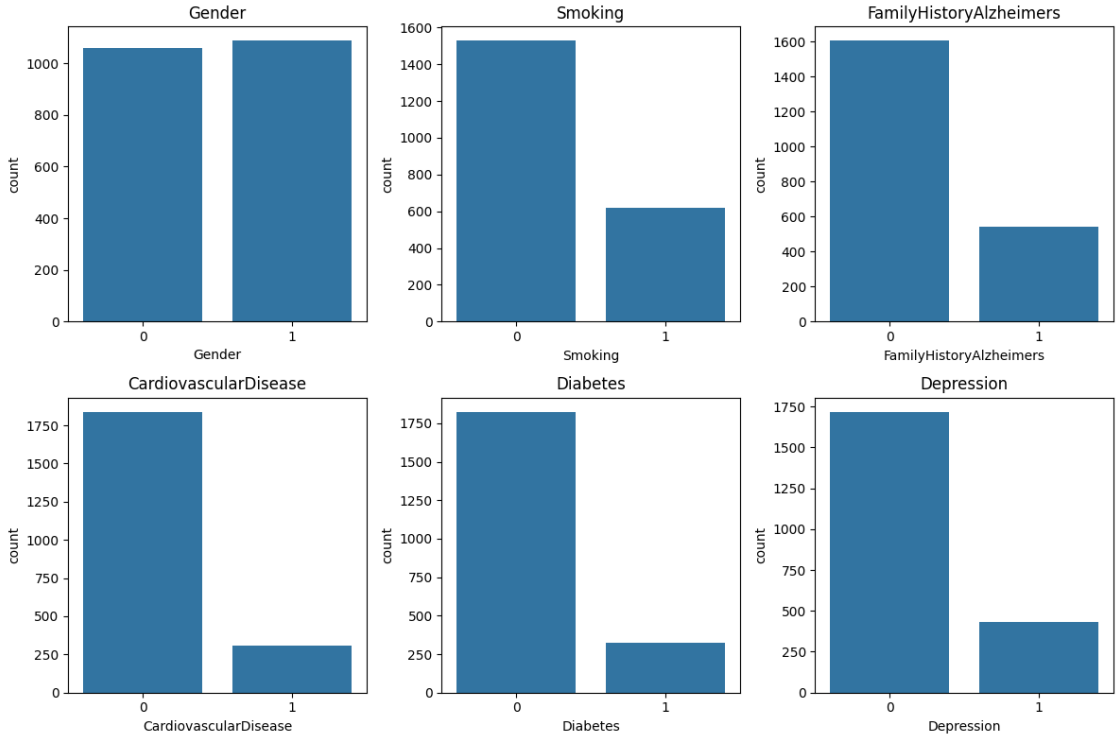
```
plt.tight_layout()
plt.show()

binary_cols = [col for col in X.columns if X[col].nunique() == 2]

print("Binary Features:", binary_cols)

plt.figure(figsize=(12,8))

for i, col in enumerate(binary_cols[:6], 1):
    plt.subplot(2,3,i)
    sns.countplot(x=col, data=data)
    plt.title(col)

plt.tight_layout()
plt.show()

#correlation with target
# Select only numeric columns
numeric_data = data.select_dtypes(include=["int64", "float64"])

# Correlation with target
corr_with_target = numeric_data.corr()["Diagnosis"].sort_values(ascending=False)


print(corr_with_target)

plt.figure(figsize=(14,10))
sns.heatmap(numeric_data.corr(), cmap="coolwarm", center=0)
plt.title("Correlation Heatmap (Numeric Only)")
plt.show()

# ================================
# Chi-Square Test for Categorical Features
# ================================

from scipy.stats import chi2_contingency

categorical_cols = [
    'Gender', 'Ethnicity', 'EducationLevel', 'Smoking',
    'FamilyHistoryAlzheimers', 'CardiovascularDisease',
    'Diabetes', 'Depression', 'HeadInjury', 'Hypertension',
    'MemoryComplaints', 'BehavioralProblems',
    'Confusion', 'Disorientation', 'PersonalityChanges',
    'DifficultyCompletingTasks', 'Forgetfulness'
]
```

```python
chi_results = []

for col in categorical_cols:
    contingency_table = pd.crosstab(data[col], data['Diagnosis'])
    chi2, p, dof, expected = chi2_contingency(contingency_table)

    chi_results.append((col, chi2, p))

# Convert to DataFrame
chi_df = pd.DataFrame(chi_results, columns=['Feature', 'Chi2', 'p-value'])

# Sort by p-value (most significant first)
chi_df = chi_df.sort_values(by='p-value')

print(chi_df)

plt.figure(figsize=(6,4))
sns.countplot(x="MemoryComplaints", hue="Diagnosis", data=data)
plt.title("MemoryComplaints vs Diagnosis")
plt.show()
```

```
['Age', 'Gender', 'Ethnicity', 'EducationLevel', 'BMI', 'Smoking',
'AlcoholConsumption', 'PhysicalActivity', 'DietQuality', 'SleepQuality',
'FamilyHistoryAlzheimers', 'CardiovascularDisease', 'Diabetes', 'Depression',
'HeadInjury', 'Hypertension', 'SystolicBP', 'DiastolicBP', 'CholesterolTotal',
'CholesterolLDL', 'CholesterolHDL', 'CholesterolTriglycerides', 'MMSE',
'FunctionalAssessment', 'MemoryComplaints', 'BehavioralProblems', 'ADL',
'Confusion', 'Disorientation', 'PersonalityChanges',
'DifficultyCompletingTasks', 'Forgetfulness', 'Diagnosis', 'DoctorInCharge']
X shape: (2149, 32)
y shape: (2149,)
Unique target values: [0 1]
Numerical Features: Index(['Age', 'Gender', 'Ethnicity', 'EducationLevel',
'BMI', 'Smoking',
       'AlcoholConsumption', 'PhysicalActivity', 'DietQuality', 'SleepQuality',
       'FamilyHistoryAlzheimers', 'CardiovascularDisease', 'Diabetes',
       'Depression', 'HeadInjury', 'Hypertension', 'SystolicBP', 'DiastolicBP',
       'CholesterolTotal', 'CholesterolLDL', 'CholesterolHDL',
       'CholesterolTriglycerides', 'MMSE', 'FunctionalAssessment',
       'MemoryComplaints', 'BehavioralProblems', 'ADL', 'Confusion',
       'Disorientation', 'PersonalityChanges', 'DifficultyCompletingTasks',
       'Forgetfulness'],
      dtype='str')
```

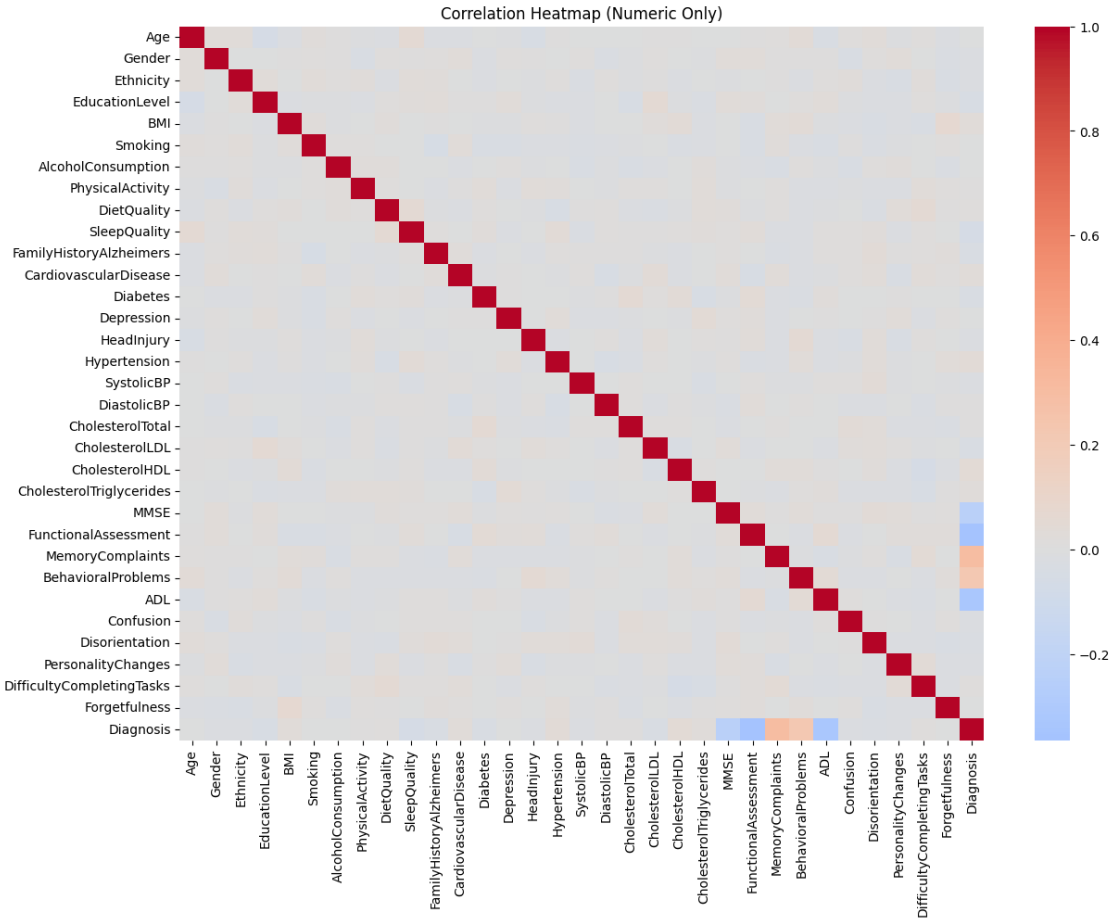



```
Binary Features: ['Gender', 'Smoking', 'FamilyHistoryAlzheimers',
'CardiovascularDisease', 'Diabetes', 'Depression', 'HeadInjury', 'Hypertension',
'MemoryComplaints', 'BehavioralProblems', 'Confusion', 'Disorientation',
'PersonalityChanges', 'DifficultyCompletingTasks', 'Forgetfulness']
```



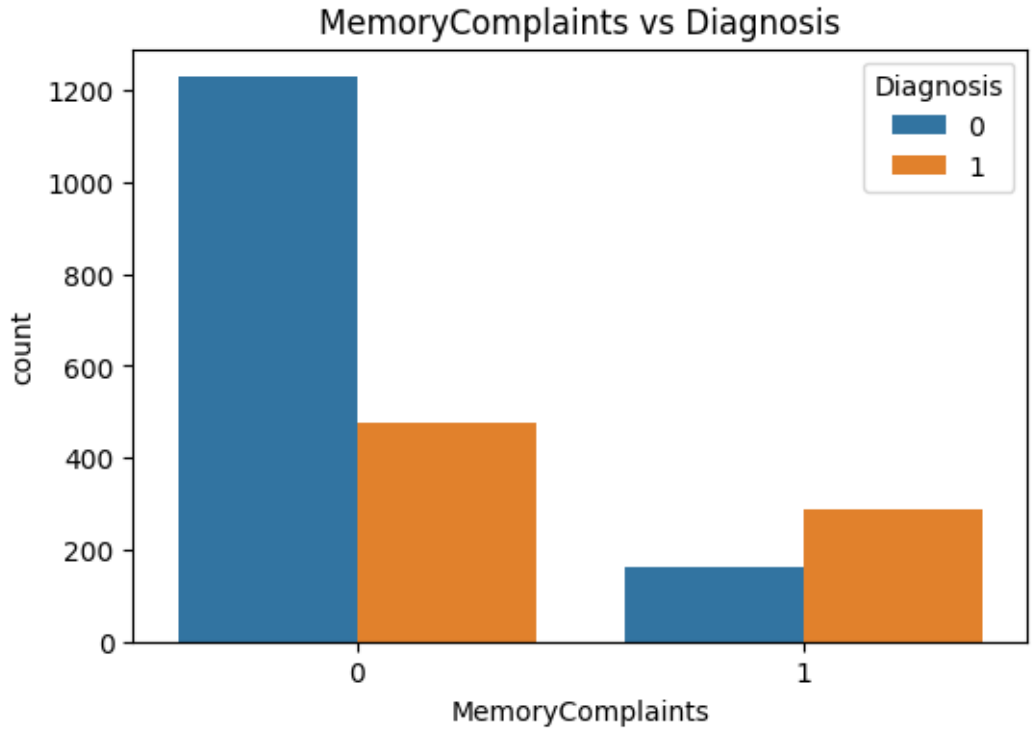

```
Diagnosis                  1.000000
MemoryComplaints           0.306742
BehavioralProblems         0.224350
CholesterolHDL             0.042584
Hypertension               0.035080
CardiovascularDisease      0.031490
BMI                        0.026343
CholesterolTriglycerides   0.022672
DifficultyCompletingTasks  0.009069
DietQuality                0.008506
CholesterolTotal           0.006394
PhysicalActivity           0.005945
DiastolicBP                0.005293
Forgetfulness             -0.000354
Smoking                   -0.004865
Age                       -0.005488
Depression                -0.005893
AlcoholConsumption        -0.007618
Ethnicity                 -0.014782
SystolicBP                -0.015615
Confusion                 -0.019186
PersonalityChanges        -0.020627
Gender                    -0.020975
HeadInjury                -0.021411
```

```
Disorientation              -0.024648
Diabetes                    -0.031508
CholesterolLDL              -0.031976
FamilyHistoryAlzheimers     -0.032900
EducationLevel              -0.043966
SleepQuality                -0.056548
MMSE                        -0.237126
ADL                         -0.332346
FunctionalAssessment        -0.364898
Name: Diagnosis, dtype: float64
```



	Feature	Chi2	p-value
10	MemoryComplaints	200.623704	1.526605e-45
11	BehavioralProblems	106.879217	4.731447e-25
1	Ethnicity	6.302089	9.780307e-02
9	Hypertension	2.442487	1.180889e-01
4	FamilyHistoryAlzheimers	2.170309	1.406980e-01
6	Diabetes	1.953177	1.622450e-01
5	CardiovascularDisease	1.947683	1.628367e-01

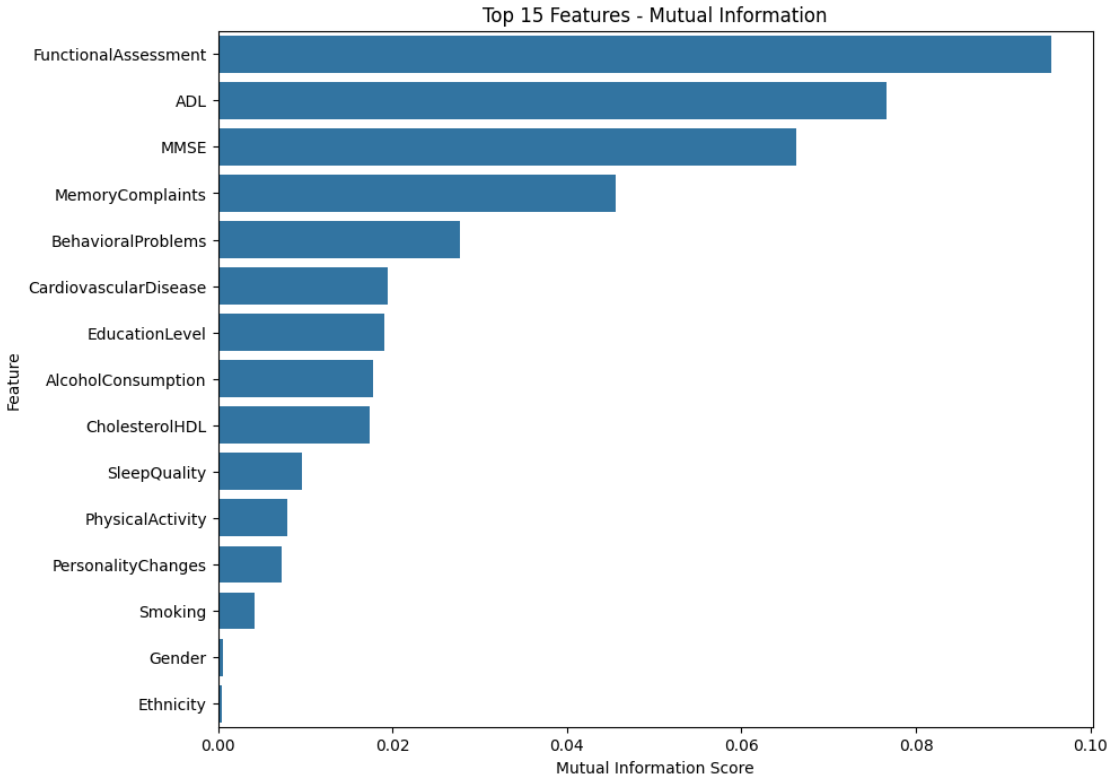
```
2              EducationLevel    4.453147  2.165077e-01
13             Disorientation    1.168140  2.797838e-01
0                      Gender    0.859716  3.538183e-01
8                  Headinjury    0.836768  3.603227e-01
14         PersonalityChanges    0.797783  3.717571e-01
12                  Confusion    0.694786  4.045414e-01
15  DifficultyCompletingTasks    0.128631  7.198557e-01
7                  Depression    0.047019  8.283335e-01
3                     Smoking    0.030887  8.604932e-01
16              Forgetfulness    0.000000  1.000000e+00
```



[29]:
```python
#run Mutual Information
from sklearn.feature_selection import mutual_info_classif

mi_scores = mutual_info_classif(X, y)

mi_df = pd.DataFrame({
    "Feature": X.columns,
    "MI Score": mi_scores
}).sort_values(by="MI Score", ascending=False)

print(mi_df)
```

```
plt.figure(figsize=(10,8))

sns.barplot(
    x="MI Score",
    y="Feature",
    data=mi_df.head(15)
)

plt.title("Top 15 Features - Mutual Information")
plt.xlabel("Mutual Information Score")
plt.ylabel("Feature")
plt.show()
```

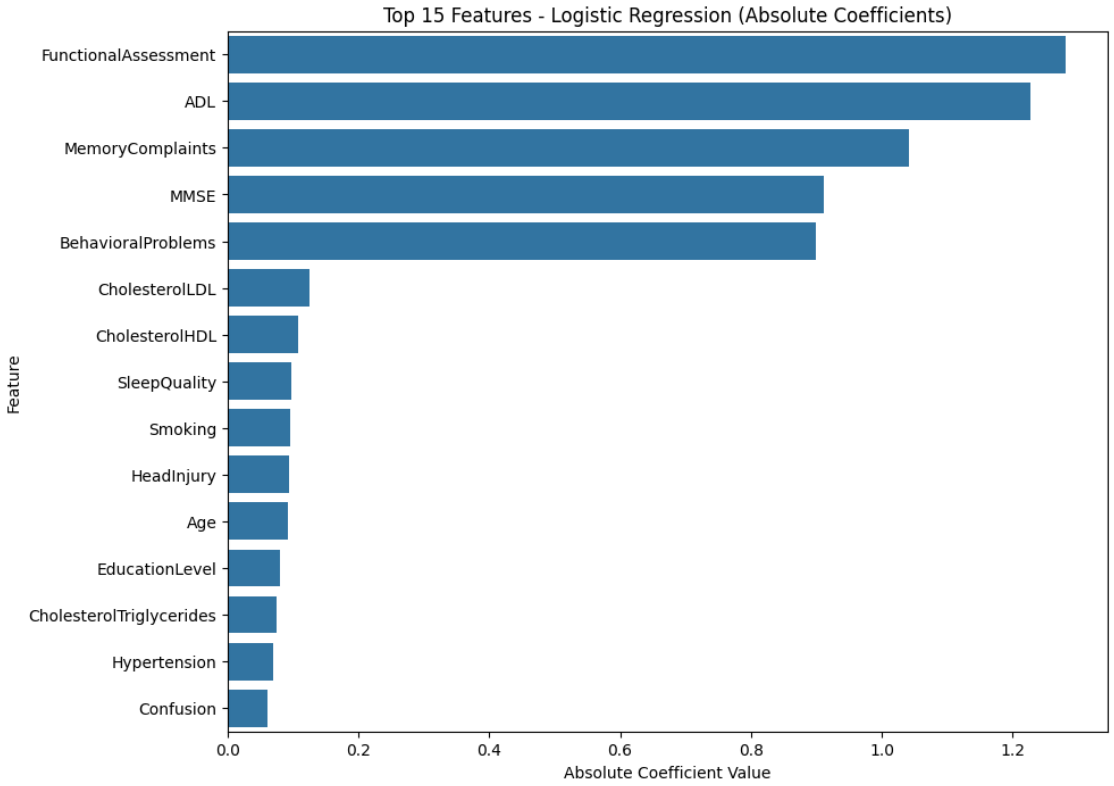
	Feature	MI Score
23	FunctionalAssessment	0.095454
26	ADL	0.076564
22	MMSE	0.066278
24	MemoryComplaints	0.045584
25	BehavioralProblems	0.027777
11	CardiovascularDisease	0.019441
3	EducationLevel	0.019135
6	AlcoholConsumption	0.017780
20	CholesterolHDL	0.017349
9	SleepQuality	0.009691
7	PhysicalActivity	0.007947
29	PersonalityChanges	0.007328
5	Smoking	0.004248
1	Gender	0.000574
2	Ethnicity	0.000478
18	CholesterolTotal	0.000338
0	Age	0.000000
27	Confusion	0.000000
21	CholesterolTriglycerides	0.000000
30	DifficultyCompletingTasks	0.000000
28	Disorientation	0.000000
16	SystolicBP	0.000000
19	CholesterolLDL	0.000000
17	DiastolicBP	0.000000
15	Hypertension	0.000000
14	HeadInjury	0.000000
13	Depression	0.000000
12	Diabetes	0.000000
10	FamilyHistoryAlzheimers	0.000000
8	DietQuality	0.000000
4	BMI	0.000000
31	Forgetfulness	0.000000



[30]:
```python
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

model = LogisticRegression(max_iter=5000)
model.fit(X_scaled, y)

coef_df = pd.DataFrame({
    "Feature": X.columns,
    "Coefficient": model.coef_[0]
}).sort_values(by="Coefficient", key=abs, ascending=False)

print(coef_df)

# Add absolute coefficient column
coef_df["Abs_Coefficient"] = coef_df["Coefficient"].abs()

coef_sorted = coef_df.sort_values(
    by="Abs_Coefficient", ascending=False
).head(15)
```

```
plt.figure(figsize=(10,8))

sns.barplot(
    x="Abs_Coefficient",
    y="Feature",
    data=coef_sorted
)

plt.title("Top 15 Features - Logistic Regression (Absolute Coefficients)")
plt.xlabel("Absolute Coefficient Value")
plt.ylabel("Feature")
plt.show()
```

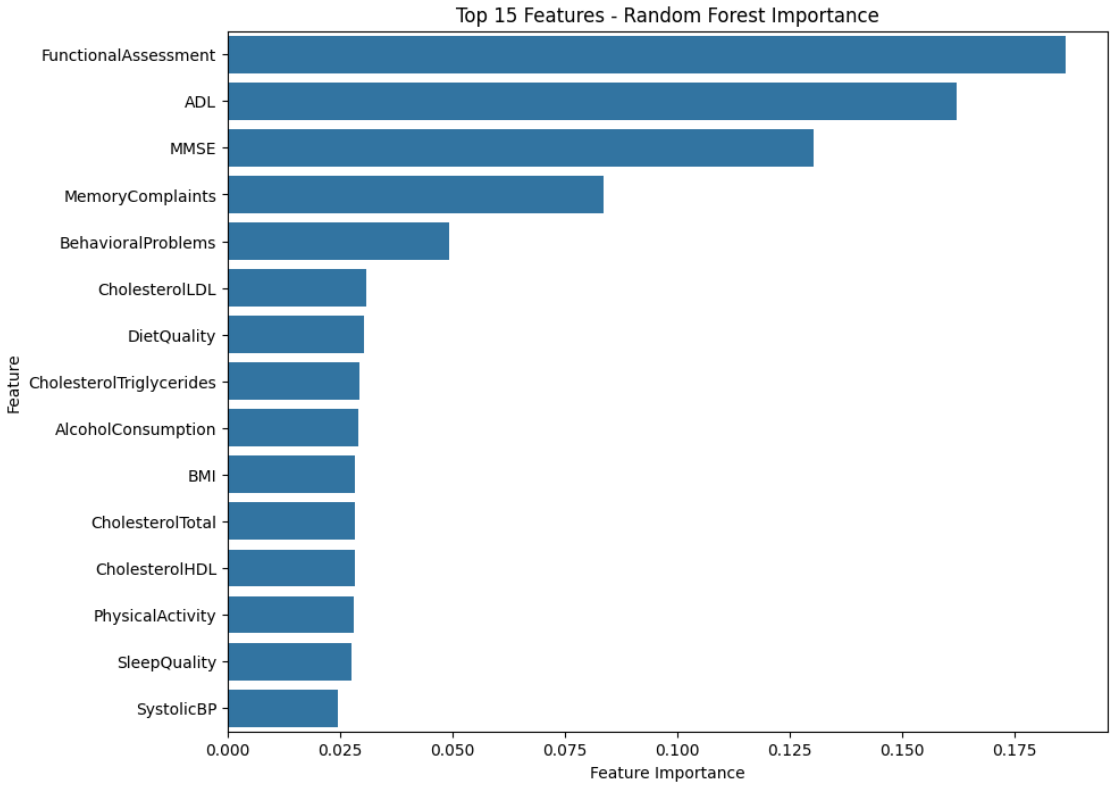
	Feature	Coefficient
23	FunctionalAssessment	-1.281453
26	ADL	-1.228264
24	MemoryComplaints	1.042513
22	MMSE	-0.911615
25	BehavioralProblems	0.899950
19	CholesterolLDL	-0.125779
20	CholesterolHDL	0.108061
9	SleepQuality	-0.096909
5	Smoking	-0.095608
14	HeadInjury	-0.093224
0	Age	-0.091350
3	EducationLevel	-0.079040
21	CholesterolTriglycerides	0.074490
15	Hypertension	0.070219
27	Confusion	-0.061393
11	CardiovascularDisease	0.058572
6	AlcoholConsumption	-0.049467
10	FamilyHistoryAlzheimers	-0.047632
28	Disorientation	-0.044182
2	Ethnicity	-0.036897
30	DifficultyCompletingTasks	0.036018
17	DiastolicBP	0.032241
4	BMI	-0.030573
8	DietQuality	0.028768
13	Depression	0.025975
29	PersonalityChanges	-0.024919
1	Gender	-0.023533
7	PhysicalActivity	-0.020372
16	SystolicBP	-0.018220
18	CholesterolTotal	0.009948
12	Diabetes	0.004981
31	Forgetfulness	0.001607



[31]:
```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(random_state=42)
rf.fit(X, y)

importance_df = pd.DataFrame({
    "Feature": X.columns,
    "Importance": rf.feature_importances_
}).sort_values(by="Importance", ascending=False)

print(importance_df)

plt.figure(figsize=(10,8))

sns.barplot(
    x="Importance",
    y="Feature",
    data=importance_df.head(15)
)

plt.title("Top 15 Features - Random Forest Importance")
plt.xlabel("Feature Importance")
```

```
plt.ylabel("Feature")
plt.show()
```

	Feature	Importance
23	FunctionalAssessment	0.186329
26	ADL	0.161961
22	MMSE	0.130136
24	MemoryComplaints	0.083670
25	BehavioralProblems	0.049218
19	CholesterolLDL	0.030777
8	DietQuality	0.030307
21	CholesterolTriglycerides	0.029223
6	AlcoholConsumption	0.029117
4	BMI	0.028366
18	CholesterolTotal	0.028358
20	CholesterolHDL	0.028295
7	PhysicalActivity	0.028094
9	SleepQuality	0.027388
16	SystolicBP	0.024561
17	DiastolicBP	0.022586
0	Age	0.021369
3	EducationLevel	0.009381
2	Ethnicity	0.007772
1	Gender	0.003923
15	Hypertension	0.003919
5	Smoking	0.003884
11	CardiovascularDisease	0.003634
31	Forgetfulness	0.003540
29	PersonalityChanges	0.003487
30	DifficultyCompletingTasks	0.003449
13	Depression	0.003212
28	Disorientation	0.003057
27	Confusion	0.002971
10	FamilyHistoryAlzheimers	0.002886
14	HeadInjury	0.002580
12	Diabetes	0.002553



[32]:
```
#Comparte Top features from all methods
top_mi = set(mi_df.head(10)["Feature"])
top_rf = set(importance_df.head(10)["Feature"])
top_lr = set(coef_sorted["Feature"])

print("Common Important Features:")
print(top_mi & top_rf & top_lr)
```

```
Common Important Features:
{'ADL', 'FunctionalAssessment', 'MMSE', 'BehavioralProblems',
'MemoryComplaints'}
```

[33]:
```
#Preprocessing Pipeline

# Define Feature Groups
# ================================

numerical_cols = [
    'Age', 'BMI', 'AlcoholConsumption', 'PhysicalActivity',
    'DietQuality', 'SleepQuality', 'SystolicBP', 'DiastolicBP',
    'CholesterolTotal', 'CholesterolLDL', 'CholesterolHDL',
    'CholesterolTriglycerides', 'MMSE',
```

```
    'FunctionalAssessment', 'ADL'
]

binary_cols = [
    'Gender', 'Smoking', 'FamilyHistoryAlzheimers',
    'CardiovascularDisease', 'Diabetes', 'Depression',
    'HeadInjury', 'Hypertension', 'MemoryComplaints',
    'BehavioralProblems', 'Confusion', 'Disorientation',
    'PersonalityChanges', 'DifficultyCompletingTasks',
    'Forgetfulness'
]

categorical_cols = ['Ethnicity']

ordinal_cols = ['EducationLevel']

# Build Preprocessing Transformer
# ================================

from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer

preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical_cols),
        ('cat', OneHotEncoder(drop='first'), categorical_cols),
        ('bin', 'passthrough', binary_cols),
        ('ord', 'passthrough', ordinal_cols)
    ]
)

from sklearn.pipeline import Pipeline

pipeline = Pipeline(steps=[
    ('preprocessing', preprocessor)
])
```

[34]:
```
#Train Test split
from sklearn.model_selection import train_test_split

X = data.drop(columns=["Diagnosis", "DoctorInCharge"])
y = data["Diagnosis"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42,
```

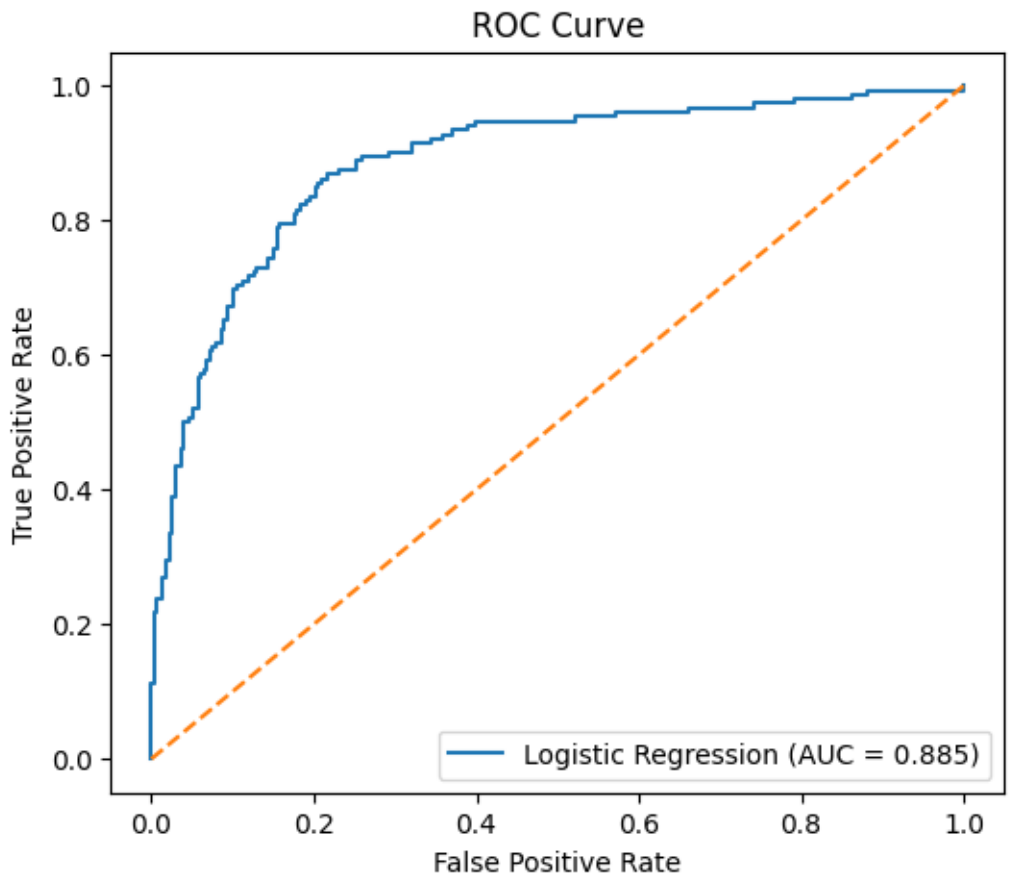
```
    stratify=y
)

print("Train shape:", X_train.shape)
print("Test shape:", X_test.shape)
```

```
Train shape: (1719, 32)
Test shape: (430, 32)
```

[39]:
```
#Logistic Regression Model
class_weight="balanced"

from sklearn.linear_model import LogisticRegression

log_model = Pipeline(steps=[
    ('preprocessing', preprocessor),
    ('model', LogisticRegression(
        max_iter=5000,
        class_weight="balanced",
        random_state=42
    ))
])

# Train
log_model.fit(X_train, y_train)

# Predict class labels
y_pred = log_model.predict(X_test)

# Predict probabilities (for ROC)
y_prob = log_model.predict_proba(X_test)[:, 1]

from sklearn.metrics import accuracy_score, classification_report,␣
 ↪confusion_matrix, roc_auc_score

# Accuracy
acc = accuracy_score(y_test, y_pred)
print("Accuracy:", acc)

# Confusion Matrix
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))

# Classification Report
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```python
# ROC-AUC
roc_auc = roc_auc_score(y_test, y_prob)
print("\nROC-AUC Score:", roc_auc)

from sklearn.metrics import roc_curve

fpr, tpr, thresholds = roc_curve(y_test, y_prob)

plt.figure(figsize=(6,5))
plt.plot(fpr, tpr, label=f"Logistic Regression (AUC = {roc_auc:.3f})")
plt.plot([0,1], [0,1], linestyle="--")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve")
plt.legend()
plt.show()
```

```
Accuracy: 0.813953488372093

Confusion Matrix:
[[219  59]
 [ 21 131]]

Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.79      0.85       278
           1       0.69      0.86      0.77       152

    accuracy                           0.81       430
   macro avg       0.80      0.82      0.81       430
weighted avg       0.83      0.81      0.82       430


ROC-AUC Score: 0.8852943960620977
```



[ ]:
```
#Random Forest Model

from sklearn.ensemble import RandomForestClassifier

rf_model = Pipeline(steps=[
    ('preprocessing', preprocessor),
    ('model', RandomForestClassifier(
        n_estimators=200,
        class_weight="balanced",
        random_state=42
    ))
])
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)
y_prob_rf = rf_model.predict_proba(X_test)[:, 1]

# Evaluation
```

```
from sklearn.metrics import accuracy_score, classification_report,␣
 ↪confusion_matrix, roc_auc_score

# Accuracy
acc_rf = accuracy_score(y_test, y_pred_rf)
print("Accuracy:", acc_rf)

# Confusion Matrix
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred_rf))

# Classification Report
print("\nClassification Report:")
print(classification_report(y_test, y_pred_rf))

# ROC-AUC
roc_auc_rf = roc_auc_score(y_test, y_prob_rf)
print("\nROC-AUC Score:", roc_auc_rf)
```

```
Accuracy: 0.9325581395348838

Confusion Matrix:
[[271   7]
 [ 22 130]]

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.97      0.95       278
           1       0.95      0.86      0.90       152

    accuracy                           0.93       430
   macro avg       0.94      0.92      0.92       430
weighted avg       0.93      0.93      0.93       430


ROC-AUC Score: 0.9428246876183264
```

[42]:
```
#comparing ROC curves

# Logistic already computed earlier

fpr_rf, tpr_rf, _ = roc_curve(y_test, y_prob_rf)

plt.figure(figsize=(6,5))

# Logistic curve
```

```
plt.plot(fpr, tpr, label=f"Logistic (AUC = {roc_auc:.3f})")

# Random Forest curve
plt.plot(fpr_rf, tpr_rf, label=f"Random Forest (AUC = {roc_auc_rf:.3f})")

plt.plot([0,1], [0,1], linestyle="--")

plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve Comparison")
plt.legend()
plt.show()
```