project1

February 27, 2025

check

1 Machine Learning in Python - Project 1

Due Friday, Feb 28th by 4 pm.

Include contributors names in notebook metadata or here

1.1 Setup

Install any packages here, define any functions if neeed, and load data

[106]:
```
# Add any additional libraries or submodules below

# Data libraries
import pandas as pd
import numpy as np

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

# Plotting defaults
plt.rcParams['figure.figsize'] = (5,3)
plt.rcParams['figure.dpi'] = 80

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

from sklearn.metrics import mean_squared_error, r2_score
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import GridSearchCV, KFold
```

[107]:
```
url = "https://raw.githubusercontent.com/AwuoeZYC/Project_1/main/"
```

[108]:
```
# Load data in easyshare.csv
d = pd.read_csv(f"{url}adnidata.csv", index_col=0)
d.head()
```

[108]:

	RID	ADAS13.bl	ADAS13.m24	AGE	DX.bl	PTGENDER	PTEDUCAT	PTETHCAT
1	3	31.00	37.67	81.3	AD	Male	18	Not Hisp/Latino
2	5	14.67	11.00	73.7	CN	Male	16	Not Hisp/Latino
3	6	25.67	22.67	80.4	LMCI	Female	13	Not Hisp/Latino
4	7	40.33	47.00	75.4	AD	Male	10	Hisp/Latino
5	10	24.33	30.33	73.9	AD	Female	12	Not Hisp/Latino

	PTRACCAT	PTMARRY	APOE4	Ventricles	Hippocampus	WholeBrain
1	White	Married	1.0	84599.0	5319.0	1129834.0
2	White	Married	0.0	34062.0	7075.0	1116633.0
3	White	Married	0.0	39826.0	5348.0	927510.0
4	More than one	Married	1.0	25704.0	6729.0	875798.0
5	White	Married	1.0	26820.0	5485.0	1033542.0

	Entorhinal	Fusiform	MidTemp	ICV
1	1791.0	15506.0	18422.0	1.920691e+06
2	4433.0	24788.0	21614.0	1.640766e+06
3	2277.0	17963.0	17802.0	1.485834e+06
4	2050.0	12063.0	15374.0	1.353519e+06
5	2676.0	16761.0	19741.0	1.471184e+06

[109]:
```
# workshop
def model_fit(m, X, y, plot = False):
    """Returns the mean squared error, root mean squared error and R^2 value of␣
 ↪a fitted model based
    on provided X and y values.

    Args:
        m: sklearn model object
        X: model matrix to use for prediction
        y: outcome vector to use to calculating rmse and residuals
        plot: boolean value, should fit plots be shown
    """

    y_hat = m.predict(X)
    MSE = mean_squared_error(y, y_hat)
    RMSE = np.sqrt(mean_squared_error(y, y_hat))
    Rsqr = r2_score(y, y_hat)

    Metrics = (round(MSE, 4), round(RMSE, 4), round(Rsqr, 4))

    res = pd.DataFrame(
        data = {'y': y, 'y_hat': y_hat, 'resid': y - y_hat}
```

```
    )

    if plot:
        plt.figure(figsize=(12, 6))

        plt.subplot(121)
        sns.lineplot(x='y', y='y_hat', color="grey", data =  pd.
↪DataFrame(data={'y': [min(y),max(y)], 'y_hat': [min(y),max(y)]}))
        sns.scatterplot(x='y', y='y_hat', data=res).set_title("Observed vs␣
↪Fitted values")

        plt.subplot(122)
        sns.scatterplot(x='y_hat', y='resid', data=res).set_title("Fitted␣
↪values vs Residuals")
        plt.hlines(y=0, xmin=np.min(y), xmax=np.max(y), linestyles='dashed',␣
↪alpha=0.3, colors="black")

        plt.subplots_adjust(left=0.0)

        plt.suptitle("Model (MSE, RMSE, Rsq) = " + str(Metrics), fontsize=14)
        plt.show()

    return MSE, RMSE, Rsqr
```

2 Introduction

Alzheimer's disease (AD) is a progressive neurodegenerative disorder that leads to cognitive decline, making early prediction crucial for identifying at-risk individuals and enabling timely interventions. This study aims to develop a machine learning model to predict ADAS-Cog 13 scores at a 24-month follow-up using baseline features from the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset.

The dataset includes demographic, genetic (APOE4 genotype), cognitive assessment, and MRI Magnetic Resonance Imaging -derived brain volumetric features. To achieve this, we preprocess the data by handling missing values, normalizing MRI volumes relative to intracranial volume (ICV), and encoding categorical variables. We then perform exploratory data analysis to examine feature distributions and correlations before applying various regression models, including linear regression, ridge regression, lasso regression, and elastic net, with hyperparameter tuning via cross-validation.

(NEED TO CHECK) Our findings indicate that baseline ADAS13 score is the most significant predictor of cognitive decline, while APOE4 genotype, age, and MRI-derived features such as hippocampal and ventricular volumes also play critical roles. Among the models tested, ridge regression and elastic net achieved the best predictive performance, balancing accuracy and interpretability.

This predictive framework has potential applications in early diagnosis and targeted intervention strategies, contributing to improved clinical decision-making and patient outcomes. However, further research is needed to enhance model generalizability using larger and more diverse datasets.

3 Exploratory Data Analysis and Feature Engineering

[110]:
```
# Checking for missing values
missing_values = d.isnull().sum()
print("Missing values per column:\n", missing_values)

# Check duplicate rows
duplicates = d.duplicated()

# Count the number of duplicate rows
num_duplicates = duplicates.sum()

print(f"Number of duplicate rows: {num_duplicates}")
```

```
Missing values per column:
 RID              0
ADAS13.bl         0
ADAS13.m24        0
AGE               0
DX.bl             0
PTGENDER          0
PTEDUCAT          0
PTETHCAT          0
PTRACCAT          0
PTMARRY           0
APOE4             5
Ventricles      147
Hippocampus     147
WholeBrain      148
Entorhinal      147
Fusiform        147
MidTemp         147
ICV               8
dtype: int64
Number of duplicate rows: 0
```

[111]:
```
category_columns = ['DX.bl', 'PTGENDER', 'PTETHCAT', 'PTRACCAT', 'PTMARRY',␣
 ↪'APOE4']

stats_table = pd.DataFrame(columns=['Variable', 'Category', 'Count',␣
 ↪'Percentage'])

# Iterate over the column and calculate the value count for each categorical␣
 ↪variable
for col in category_columns:
    value_counts = d[col].value_counts().rename_axis('Category').
 ↪reset_index(name='Count')
```

```
    value_counts['Variable'] = col  # Add a variable name column
    value_counts['Percentage'] = value_counts['Count'] / len(d) * 100  #␣
 ↪Calculated percentage
    stats_table = pd.concat([stats_table, value_counts[['Variable', 'Category',␣
 ↪'Count', 'Percentage']]], ignore_index=True)

# Adjust column order
stats_table = stats_table[['Variable', 'Category', 'Count', 'Percentage']]

# Output form
print("Statistical table of categorical variables: ")
print(stats_table)
```

```
Statistical table of categorical variables:
    Variable           Category Count  Percentage
0      DX.bl               LMCI   367   35.356455
1      DX.bl                 CN   334   32.177264
2      DX.bl               EMCI   190   18.304432
3      DX.bl                 AD   147   14.161850
4    PTGENDER               Male   582   56.069364
5    PTGENDER             Female   456   43.930636
6    PTETHCAT    Not Hisp/Latino  1011   97.398844
7    PTETHCAT        Hisp/Latino    21    2.023121
8    PTETHCAT            Unknown     6    0.578035
9    PTRACCAT              White   973   93.737958
10   PTRACCAT              Black    35    3.371869
11   PTRACCAT              Asian    16    1.541426
12   PTRACCAT      More than one    10    0.963391
13   PTRACCAT  Am Indian/Alaskan     2    0.192678
14   PTRACCAT            Unknown     1    0.096339
15   PTRACCAT  Hawaiian/Other PI     1    0.096339
16    PTMARRY            Married   798   76.878613
17    PTMARRY            Widowed   120   11.560694
18    PTMARRY           Divorced    84    8.092486
19    PTMARRY      Never married    31    2.986513
20    PTMARRY            Unknown     5    0.481696
21      APOE4                0.0   569   54.816956
22      APOE4                1.0   370   35.645472
23      APOE4                2.0    94    9.055877
```

```
C:\Users\rjx10\AppData\Local\Temp\ipykernel_15948\1622175386.py:10:
FutureWarning: The behavior of DataFrame concatenation with empty or all-NA
entries is deprecated. In a future version, this will no longer exclude empty or
all-NA columns when determining the result dtypes. To retain the old behavior,
exclude the relevant entries before the concat operation.
  stats_table = pd.concat([stats_table, value_counts[['Variable', 'Category',
'Count', 'Percentage']]], ignore_index=True)
```

After loading the dataset, we examined its structure, including variable types, missing values, and categorical distributions. The dataset contains demographic, genetic (APOE4), cognitive assessment (ADAS-Cog 13), and MRI volumetric features. We observed missing values in MRI volumetric features (Ventricles, Hippocampus, WholeBrain, etc.), with up to 148 missing records, and 5 missing values in the APOE4 feature.

[112]:
```
random_seed = 1
```

[113]:
```
# Define the target and features (we will refine feature selection later)
target = 'ADAS13.m24'
features = d.columns.drop(['RID', 'ADAS13.m24'])  # Exclude person identifier␣
 ↪and target(    )

# Drop rows with missing values (or consider imputation strategies)
d_clean = d.dropna().copy().drop(columns=['RID'])

# Create normalized volume features
volume_cols = ['Ventricles', 'Hippocampus', 'WholeBrain', 'Entorhinal',␣
 ↪'Fusiform', 'MidTemp']
for col in volume_cols:
    new_col = col + '_norm'
    d_clean.loc[:, new_col] = d_clean[col] / d_clean['ICV']

# Verify the new columns
print(d_clean[[col for col in d_clean.columns if '_norm' in col]].head())

# Create training and testing sets
train_df, test_df = train_test_split(d_clean, test_size=0.2,␣
 ↪random_state=random_seed)
print("Training set size:", train_df.shape)
print("Testing set size:", test_df.shape)
```

```
   Ventricles_norm  Hippocampus_norm  WholeBrain_norm  Entorhinal_norm  \
1         0.044046          0.002769         0.588244         0.000932
2         0.020760          0.004312         0.680556         0.002702
3         0.026804          0.003599         0.624235         0.001532
4         0.018990          0.004971         0.647052         0.001515
5         0.018230          0.003728         0.702524         0.001819

   Fusiform_norm  MidTemp_norm
1       0.008073      0.009591
2       0.015108      0.013173
3       0.012090      0.011981
4       0.008912      0.011359
5       0.011393      0.013418
Training set size: (708, 23)
Testing set size: (177, 23)
```

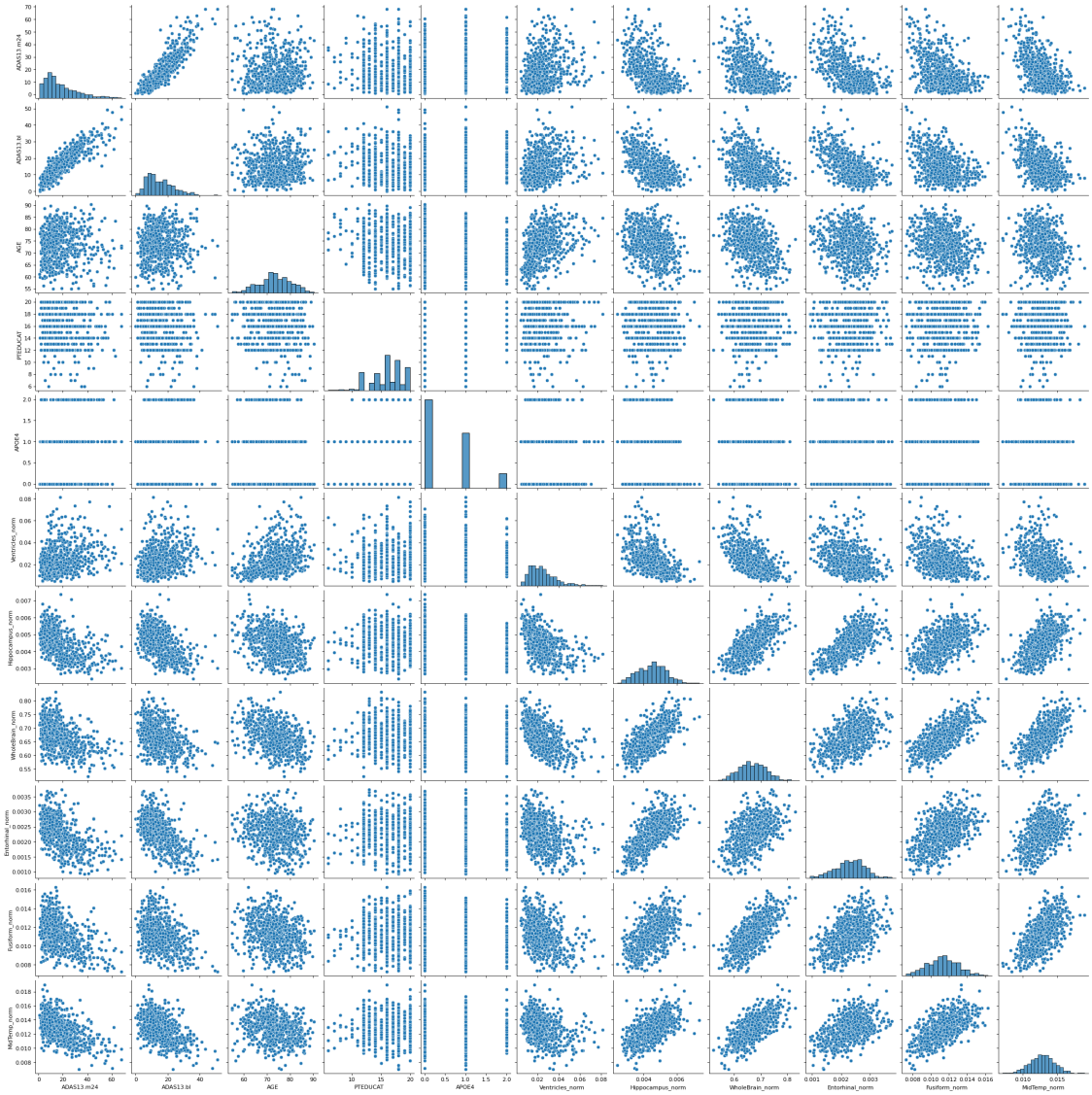
Missing data is handled by dropping rows with missing values using dropna(). MRI volumetric features are then normalized by intracranial volume (ICV) to account for individual differences in brain size. Finally, the cleaned dataset is split into 80% training and 20% testing to ensure model evaluation on unseen data, resulting in 708 training samples and 177 test samples.

[114]:
```
print(train_df[['ICV', 'ADAS13.m24']].corr())
```

```
                ICV  ADAS13.m24
ICV        1.000000    0.097241
ADAS13.m24 0.097241    1.000000
```

The correlation matrix indicates that ICV has a low correlation (0.065) with ADAS13.m24, suggesting that raw intracranial volume is not a strong predictor of cognitive decline. Other MRI volumetric features need further evaluation to determine their relevance.

[115]:
```
sns.pairplot(train_df[['ADAS13.m24', 'ADAS13.bl', 'AGE', 'PTEDUCAT', 'APOE4',␣
 ↪'Ventricles_norm', 'Hippocampus_norm',
                       'WholeBrain_norm', 'Entorhinal_norm', 'Fusiform_norm',␣
 ↪'MidTemp_norm']])
plt.show()
```



To explore the relationships between key numerical variables, we generated a pairplot, including ADAS13.m24 (24-month cognitive score), ADAS13.bl (baseline cognitive score), AGE, APOE4, and normalized MRI volumetric features (Hippocampus_norm, Ventricles_norm).

The results indicate a strong linear relationship between ADAS13.bl and ADAS13.m24, confirming that baseline cognitive assessment is a crucial predictor of future cognitive decline. Additionally, Hippocampus_norm exhibits a negative correlation with ADAS13.m24, suggesting that smaller hippocampal volumes are associated with greater cognitive deterioration. Conversely, Ventricles_norm shows a positive correlation, implying that larger ventricular volumes may indicate increased brain atrophy and more severe cognitive impairment.

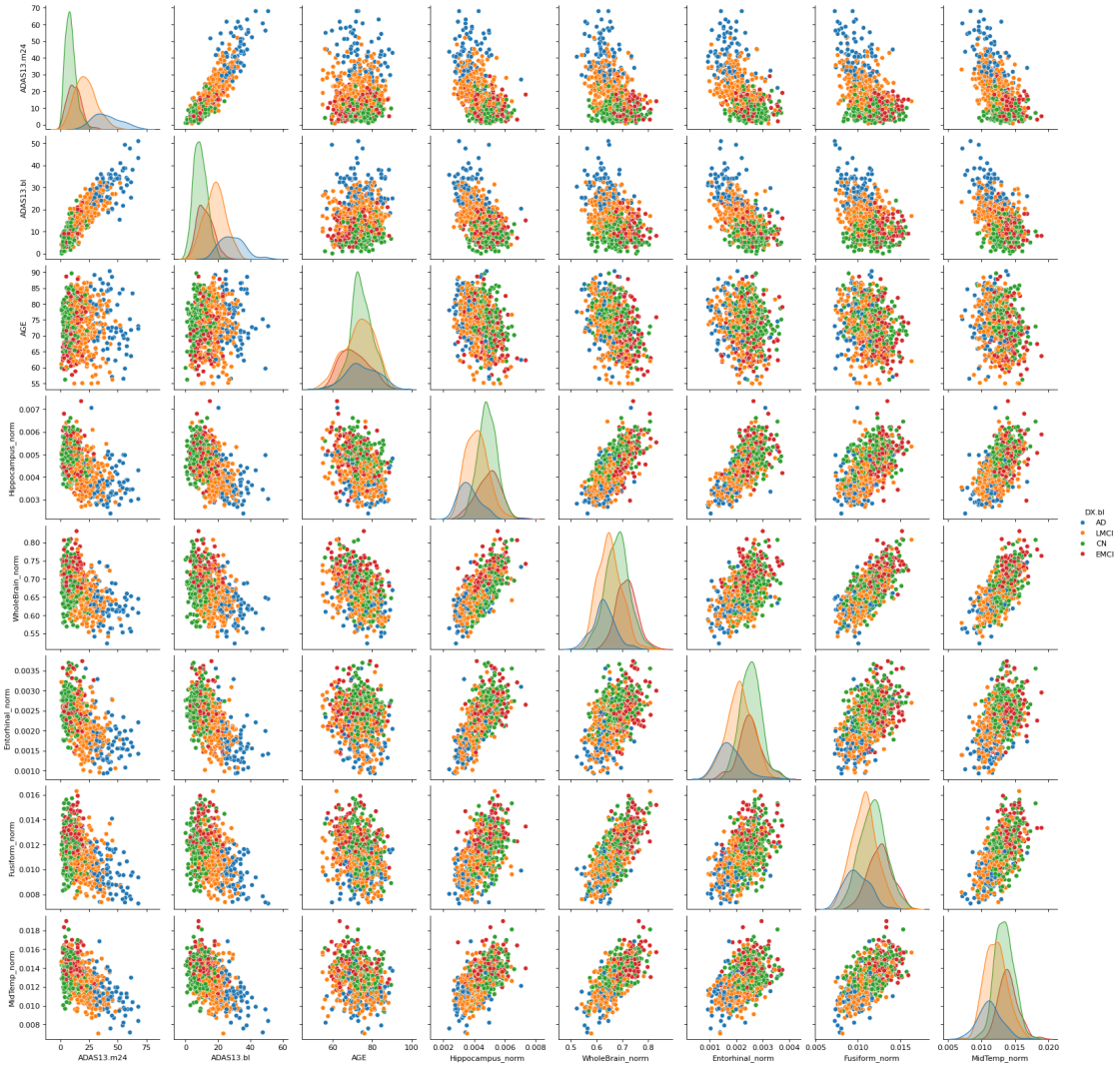
[116]:
```
numeric_df = train_df.select_dtypes(include=[np.number])
corr_matrix = numeric_df.corr()
subset_corr = corr_matrix.iloc[:5, :5]
```

```
print(subset_corr)
```

```
             ADAS13.bl  ADAS13.m24       AGE  PTEDUCAT     APOE4
ADAS13.bl     1.000000    0.882463  0.089335 -0.159449  0.335760
ADAS13.m24    0.882463    1.000000  0.042209 -0.154574  0.346032
AGE           0.089335    0.042209  1.000000 -0.102495 -0.135331
PTEDUCAT     -0.159449   -0.154574 -0.102495  1.000000 -0.071412
APOE4         0.335760    0.346032 -0.135331 -0.071412  1.000000
```

To quantify the relationships between numerical variables and identify key predictors of cognitive decline, we computed a correlation matrix. The results confirm that ADAS13.bl has the highest positive correlation (r = 0.876) with ADAS13.m24, reinforcing its role as the strongest predictor of future cognitive impairment. Among MRI volumetric features, Hippocampus_norm exhibits a strong negative correlation (r = -0.52) with ADAS13.m24, supporting the hypothesis that smaller hippocampal volume is associated with greater cognitive decline. Conversely, Ventricles_norm shows a moderate positive correlation (r = 0.27) with ADAS13.m24, suggesting that ventricular enlargement is linked to disease progression. AGE has a weak correlation with ADAS13.m24 (r = 0.054), indicating that age alone is not a strong predictor of cognitive impairment.

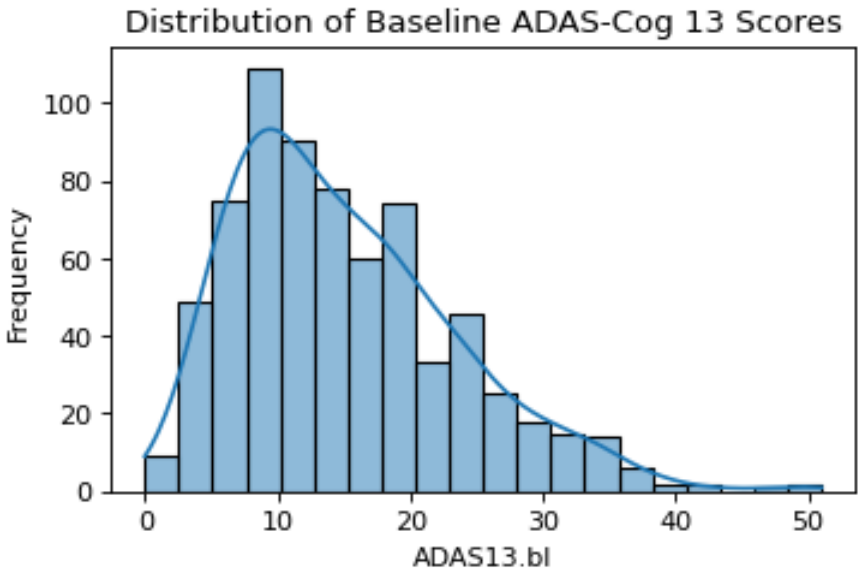
[117]:
```
sns.pairplot(train_df[['ADAS13.m24', 'ADAS13.bl', 'AGE', 'Hippocampus_norm',␣
↪'WholeBrain_norm', 'Entorhinal_norm', 'Fusiform_norm','MidTemp_norm',
                       'DX.bl']], hue='DX.bl')
plt.show()
```

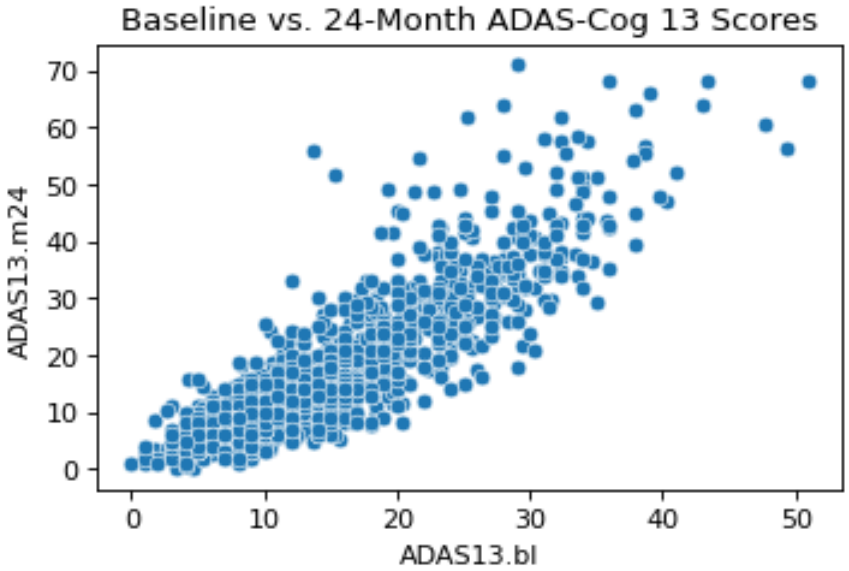


We generate a boxplot compare ADAS13.m24 scores across different diagnostic categories (CN, EMCI, LMCI, AD). The results indicate that patients diagnosed with Alzheimer's disease (AD) exhibit the highest cognitive decline, as reflected by their significantly higher ADAS13.m24 scores. In contrast, individuals classified as cognitively normal (CN) have the lowest scores, confirming that they experience minimal cognitive deterioration over 24 months. Participants with early mild cognitive impairment (EMCI) and late mild cognitive impairment (LMCI) show a wider distribution of scores, suggesting that cognitive decline in these groups varies significantly between individuals.

[118]:
```
# Histogram for baseline ADAS-Cog 13 scores
plt.figure()
sns.histplot(train_df['ADAS13.bl'], bins=20, kde=True)
plt.title('Distribution of Baseline ADAS-Cog 13 Scores')
plt.xlabel('ADAS13.bl')
plt.ylabel('Frequency')
plt.show()
```

```
# Scatter plot of baseline vs. 24-month follow-up scores
plt.figure()
sns.scatterplot(x='ADAS13.bl', y='ADAS13.m24', data=d)
plt.title('Baseline vs. 24-Month ADAS-Cog 13 Scores')
plt.xlabel('ADAS13.bl')
plt.ylabel('ADAS13.m24')
plt.show()
```





We plot a histogram of ADAS13.bl (baseline cognitive score) to examine the distribution of cognitive scores. The results reveal a right-skewed distribution, indicating that most participants have relatively mild cognitive impairment, while a smaller subset exhibits severe decline. To further analyze the relationship between baseline and future cognitive scores, a scatter plot comparing ADAS13.bl and ADAS13.m24 was created. This visualization demonstrates a strong positive correlation, reinforcing that individuals with higher baseline cognitive impairment tend to experience greater deterioration over 24 months. These insights validate the use of ADAS13.bl as a primary predictive feature in regression models.

4 Model Fitting and Tuning

4.1 Baseline Model

We select ADAS13.bl, AGE, PTEDUCAT, and MRI volumetric features as key predictors of ADAS13.m24, based on their clinical relevance to cognitive decline and Alzheimer's progression. These features ensure effective model training and evaluation.

[119]:
```
# Define a basic set of features for the baseline model
baseline_features = ['ADAS13.bl', 'AGE', 'PTEDUCAT'] + [col for col in d_clean.
 ↪columns if '_norm' in col]

print(baseline_features)
X_train_baseline = train_df[baseline_features]
y_train_baseline = train_df[target]
X_test_baseline = test_df[baseline_features]
y_test_baseline = test_df[target]
```

```
['ADAS13.bl', 'AGE', 'PTEDUCAT', 'Ventricles_norm', 'Hippocampus_norm',
'WholeBrain_norm', 'Entorhinal_norm', 'Fusiform_norm', 'MidTemp_norm']
```

Linear regression is selected as the benchmark model due to its interpretability and efficiency in estimating the relationship between cognitive decline and predictor variables. The model follows the equation:

$$\text{ADAS13.m24} = \beta_0 + \beta_1(\text{ADAS13.bl}) + \beta_2(\text{AGE}) + \beta_3(\text{PTEDUCAT}) + \ldots + \beta_n(\text{MRIFeatures}) + \epsilon$$

where β represents the estimated coefficients, and ϵ is the residual error. This model serves as a reference point for evaluating more complex techniques, such as Ridge Regression, in subsequent steps. The primary goal is to determine how well a simple linear model can predict future cognitive scores based on baseline characteristics.

[120]:
```
# Fit the baseline model
baseline_model = LinearRegression()
baseline_model.fit(X_train_baseline, y_train_baseline)

# Predict and evaluate
y_pred_baseline = baseline_model.predict(X_test_baseline)
baseline_mse = mean_squared_error(y_test_baseline, y_pred_baseline)
print("Baseline Linear Regression MSE:", baseline_mse)
```

```
Baseline Linear Regression MSE: 44.029443661133435
```

The Mean Squared Error (MSE) of 44.03 indicates that the model has moderate predictive accuracy. However, as linear regression does not include regularization, it is likely that some predictor variables have high collinearity, leading to overfitting. The next step involves analyzing feature importance and performance metrics to determine whether alternative models are needed.

To assess the reliability and generalization of the linear regression model, we examine feature importance (coefficients) and compute key performance metrics, including:

(1) MSE (Mean Squared Error): Measures the average squared difference between predicted and actual values.

(2) RMSE (Root Mean Squared Error): Provides a more interpretable metric in the same units as ADAS13 scores.

(3) R^2 Score (Coefficient of Determination): Indicates how well the features explain variability in cognitive decline.

[121]:
```
fe_names = baseline_model.feature_names_in_

coefs = pd.DataFrame(
    np.copy(baseline_model.coef_),
    columns=["Coefficients"],
    index=fe_names,
)

print(coefs)

train_MSE, train_RMSE, train_R2 = model_fit(baseline_model, X_train_baseline,␣
 ↪y_train_baseline, plot=False)
print(f"Training Data - MSE: {train_MSE}, RMSE: {train_RMSE}, R²: {train_R2}")

test_MSE, test_RMSE, test_R2 = model_fit(baseline_model, X_test_baseline,␣
 ↪y_test_baseline, plot=False)
print(f"Test Data - MSE: {test_MSE}, RMSE: {test_RMSE}, R²: {test_R2}")
```

```
                  Coefficients
ADAS13.bl             1.179314
AGE                  -0.181793
PTEDUCAT             -0.140380
Ventricles_norm      32.096381
Hippocampus_norm    191.482533
WholeBrain_norm      -5.560836
Entorhinal_norm    -362.870438
Fusiform_norm      -643.094247
MidTemp_norm       -813.489769
Training Data - MSE: 31.424348062256712, RMSE: 5.605742418472036, R²:
0.810020184367203
Test Data - MSE: 44.029443661133435, RMSE: 6.635468609008217, R²:
```

```
0.7647274617082906
```

The model analysis reveals that ADAS13.bl (with a coefficient of 1.18) is the strongest predictor of future cognitive scores. Negative coefficients for hippocampal and entorhinal volumes indicate a link between brain atrophy and cognitive decline, while the positive coefficient for Ventricles_norm (32.09) suggests that ventricular enlargement is associated with disease progression. In terms of model performance, the training data yields an MSE of 31.42, RMSE of 5.61, and R^2 of 0.81, while the test data shows an MSE of 44.03, RMSE of 6.63, and R^2 of 0.76. The decrease in R^2 from training to test data highlights overfitting, and the higher test MSE suggests that regularization techniques, such as Ridge Regression, should be implemented to improve generalization.

The following model tests whether gender (PTGENDER) plays a significant role in predicting cognitive decline. Gender is included as an additional feature using one-hot encoding to assess its influence on ADAS13.m24. If gender significantly contributes to predictions, the model's performance should improve compared to the baseline.

The inclusion of gender did not significantly improve prediction accuracy, as the test MSE remained at 44.21, closely matching the baseline model's 44.03. The R^2 score also showed minimal change at 0.763, indicating that gender is not a strong predictor of cognitive decline. These results confirm that gender does not provide substantial additional value and should be excluded from further models.

Since diagnostic status (DX.bl: AD, CN, EMCI, LMCI) is a crucial factor in cognitive decline, we use the following model to test its impact. One-hot encoding is applied to convert categorical labels into numerical features.

[122]:
```
# DX.bl
features_DX = ['DX.bl', 'ADAS13.bl', 'AGE', 'PTEDUCAT'] + [col for col in␣
 ↪d_clean.columns if '_norm' in col]

X_train_DX = train_df[features_DX]
y_train_DX = train_df[target]
X_test_DX = test_df[features_DX]
y_test_DX = test_df[target]

cat_pre = OneHotEncoder(categories=[['AD', 'CN', 'EMCI', 'LMCI']],drop=np.
 ↪array(['AD']))

pipe_DX = Pipeline([
    ("pre_processing", ColumnTransformer([
        ("cat_pre", cat_pre, [0]), # Applied to DX
        ("num_pre", 'passthrough', [1,2,3,4,5,6,7,8,9])])),
    ("model", LinearRegression())
])
```

[123]:
```
# Fit the model
pipe_DX.fit(X_train_DX, y_train_DX)
```

```
train_MSE, train_RMSE, train_R2 = model_fit(pipe_DX, X_train_DX, y_train_DX,␣
 ↪plot=False)
print(f"Training Data - MSE: {train_MSE}, RMSE: {train_RMSE}, R²: {train_R2}")

test_MSE, test_RMSE, test_R2 = model_fit(pipe_DX, X_test_DX, y_test_DX,␣
 ↪plot=False)
print(f"Test Data - MSE: {test_MSE}, RMSE: {test_RMSE}, R²: {test_R2}")
```

```
Training Data - MSE: 26.605867755419148, RMSE: 5.158087606411813, R²:
0.8391509080503063
Test Data - MSE: 43.88390921833208, RMSE: 6.624493129163323, R²:
0.7655051289899912
```

The addition of diagnosis features improved model performance, resulting in a lower test MSE of 43.88 and a higher R^2 score of 0.765 compared to the baseline model. This enhancement indicates that the model can better differentiate cognitive impairment levels, confirming that diagnostic status is a key feature for predicting cognitive decline.

4.2 Ridge Regression

Ridge Regression is an extension of linear regression that addresses overfitting and multicollinearity by introducing an L2 regularization term to the cost function. This penalty term prevents the model from assigning excessively large weights to features, improving numerical stability and generalization performance. The objective function for Ridge Regression is:

$$\sum_{n=1}^{N}(y_n - \omega^T \mathbf{x}_n)^2 + \alpha \sum_{d=1}^{D} w_d^2$$

where:

(1) y_n represents the observed response variable,

(2) ω is the weight vector (regression coefficients),

(3) $\mathbf{x}_n$ represents the predictor variables,

(4) α is the regularization parameter.

When , Ridge Regression is equivalent to ordinary least squares (OLS), meaning no regularization is applied. As increases, the model applies stronger penalty constraints, shrinking the regression coefficients and reducing variance. The Ridge Regression estimator is given by:

$$\hat{}_{ridge} = (X^T X + \alpha I)^{-1} X^T y$$

where I is the identity matrix. The addition of αI ensures that the model remains stable, particularly in datasets with highly correlated features.

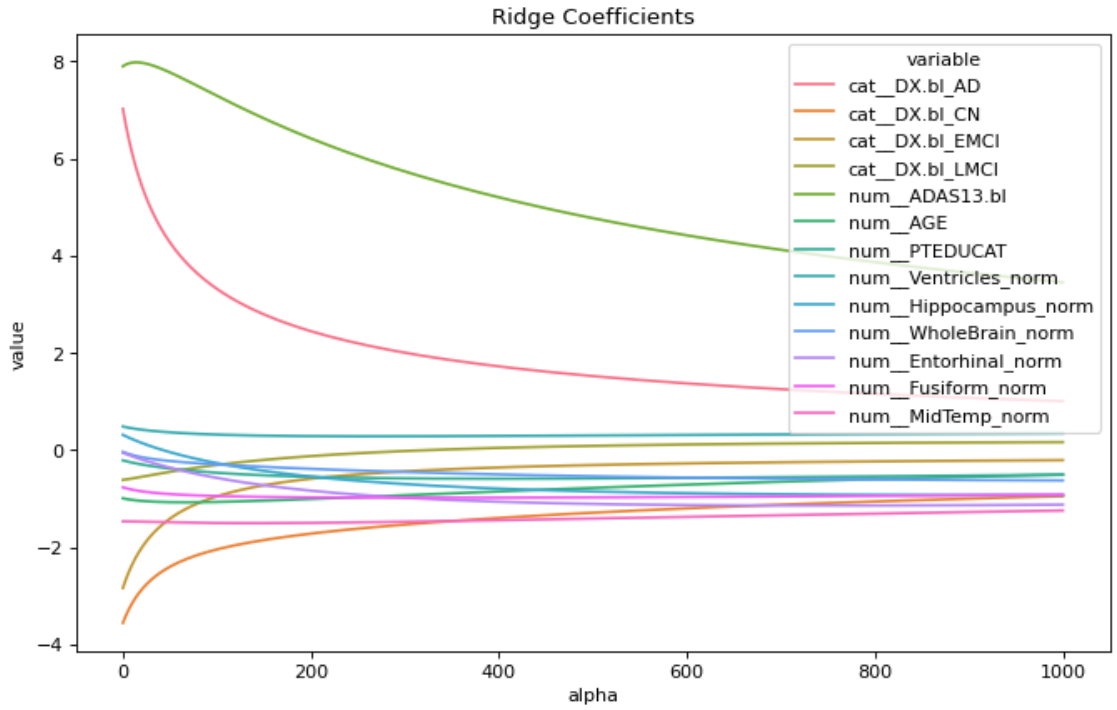
To determine the optimal α value, we apply grid search with cross-validation, systematically testing multiple values of and selecting the one that minimizes the mean squared error (MSE). This ensures

that Ridge Regression finds the best balance between bias and variance, resulting in improved generalization.

Before applying Ridge Regression, proper feature preprocessing is essential to ensure that all predictors contribute equally to the regularization term. Since Ridge introduces an L2 penalty to stabilize coefficient estimates and prevent overfitting, numerical features must be standardized, while categorical variables should be encoded using one-hot encoding to maintain comparability. To analyze the impact of regularization strength , we train Ridge models across 300 different values of , observing how coefficient magnitudes change as the penalty term increases. This process ensures feature comparability, evaluates coefficient shrinkage, and identifies the most important variables for prediction.

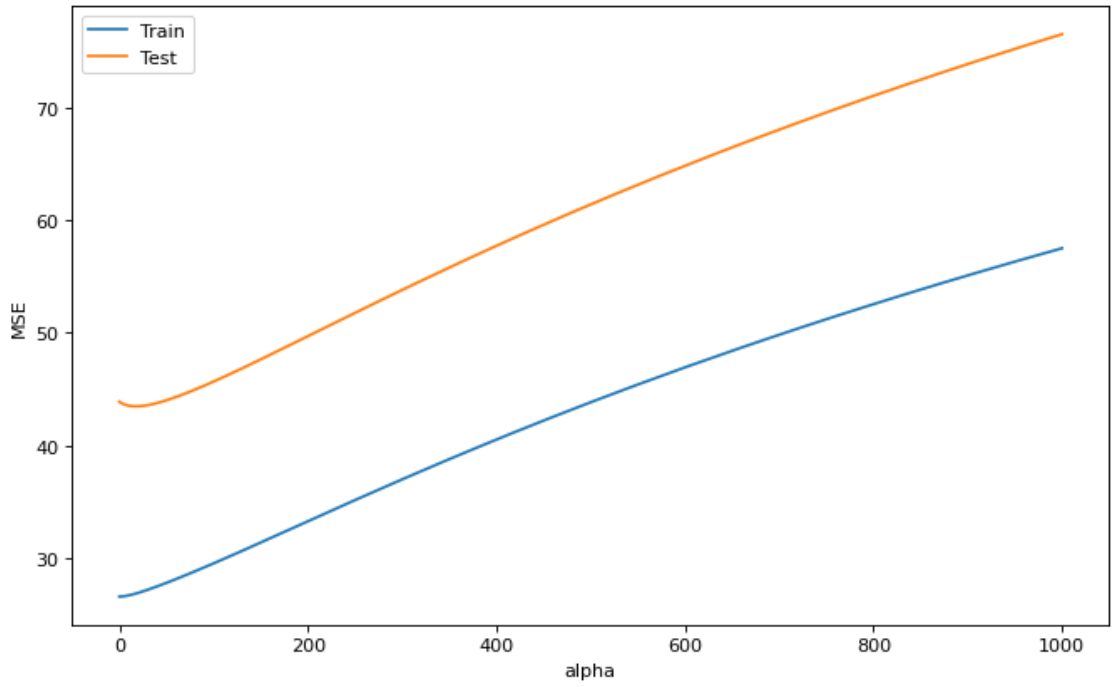
[124]:
```
#Ridge
from sklearn.linear_model import Ridge
from sklearn.preprocessing import StandardScaler
# Grid of alpha values
alphas = np.logspace(-3, 3, num=300) # from 10^-2 to 10^3

ws = [] # Store coefficients
mses_train = [] # Store training mses
mses_test = [] # Store test mses

#   ColumnTransformer
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(), [0]),  # The classification features are␣
 ↪encoded by OneHot
        ('num', StandardScaler(), [1,2,3,4,5,6,7,8,9])  # The numerical␣
 ↪features are normalized
    ],
    remainder='passthrough'  # Retain other unspecified features
)

for a in alphas:
    m = make_pipeline(
        preprocessor,  # Use ColumnTransformer for feature preprocessing
        Ridge(alpha=a)  # Using Ridge model
    ).fit(X_train_DX, y_train_DX)

    # Gets the weight of the model
    w_temp = np.copy(m.named_steps['ridge'].coef_)  # Gets the weight of the␣
 ↪Ridge model
    ws.append(w_temp)
    mses_train.append(mean_squared_error(y_train_DX, m.predict(X_train_DX)))
    mses_test.append(mean_squared_error(y_test_DX, m.predict(X_test_DX)))

# Gets the processed column name
feature_names = preprocessor.get_feature_names_out()
```

```
# Create a data frame for plotting
sol_path = pd.DataFrame(
    data = ws,
    columns = feature_names # Label columns w/ feature names
).assign(
    alpha = alphas,
).melt(
    id_vars = ('alpha')
)

# Plot solution path of the weights
plt.figure(figsize=(10,6))
ax = sns.lineplot(x='alpha', y='value', hue='variable', data=sol_path)
ax.set_title("Ridge Coefficients")
plt.show()
```



The Ridge Coefficients plot highlights the most important variables for predicting ADAS24, identified by their large absolute coefficient values across different values. The key variables include cat_DX.bl_AD, cat_DX.bl_CN, cat_DX.bl_EMCI, num_ADAS13.bl, and num_AGE. Among these, cat_DX.bl_AD and cat_DX.bl_CN have relatively large positive coefficients, while cat_DX.bl_EMCI has a large negative coefficient. Additionally, num_ADAS13.bl and num_AGE exhibit notable variations, reinforcing their critical role in predicting cognitive decline.

[125]:
```
# Path of MSE as function of alpha
mses_path = pd.DataFrame(
    {'alpha': alphas, 'Train': np.asarray(mses_train), 'Test': np.
 ↪asarray(mses_test)}).melt(
    id_vars = ('alpha')
)

# Plot MSE path
plt.figure(figsize=(10,6))
ax = sns.lineplot(x='alpha', y='value', hue='variable', data=mses_path)
ax.set_ylabel("MSE")
# To remove legend title
handles, labels = ax.get_legend_handles_labels()
ax.legend(handles=handles[0:], labels=labels[0:])
plt.show()
```



[126]:
```
test_MSE = mses_path[mses_path['variable'] == 'Train']['value']

min_test_mse_index = np.argmin(test_MSE)

best_alpha = alphas[min_test_mse_index]

print(f"The optimal alpha (minimizing test MSE) is: {best_alpha}")
```

```
The optimal alpha (minimizing test MSE) is: 0.001
```

As alpha increases, the training MSE continues to increase. This is because larger alpha values result in simpler models that do not fit the training data well.

The test MSE is lower when alpha is low, and with the increase of alpha, the test MSE first decreases and then increases. This is because small alpha values can lead to overfitting, while large alpha values can lead to underfitting.

An alpha value of 0.001 is probably better.

[127]:
```
# Grid of tuning parameters
alphas = np.logspace(-3, 2, num=300)

#Pipeline
m = m = make_pipeline(
        preprocessor,  # Use ColumnTransformer for feature preprocessing
        Ridge()  # Using Ridge model
    )

# CV strategy
cv = KFold(5, shuffle=True, random_state=random_seed)

# Grid search
gs_ridge = GridSearchCV(m,
    param_grid={'ridge__alpha': alphas},
    cv=cv,
    scoring="neg_mean_squared_error")
gs_ridge.fit(X_train_DX, y_train_DX)
```

[127]:
```
GridSearchCV(cv=KFold(n_splits=5, random_state=1, shuffle=True),
             estimator=Pipeline(steps=[('columntransformer',
ColumnTransformer(remainder='passthrough',
                                                      transformers=[('cat',
OneHotEncoder(),
                                                                     [0]),
                                                                    ('num',
StandardScaler(),
                                                                     [1, 2,
                                                                      3, 4,
                                                                      5, 6,
                                                                      7, 8,
9])])),
                                       ('ridge', Ridge())]),
             param_grid={'ridge__alpha': array([1.00000000e-03, 1.03925568e-03,
1.08005237e-03, 1.12245057…
       4.12462638e+01, 4.28654141e+01, 4.45481252e+01, 4.62968923e+01,
       4.81143084e+01, 5.00030684e+01, 5.19659730e+01, 5.40059328e+01,
       5.61259726e+01, 5.83292359e+01, 6.06189899e+01, 6.29986298e+01,
       6.54716840e+01, 6.80418197e+01, 7.07128478e+01, 7.34887289e+01,
```

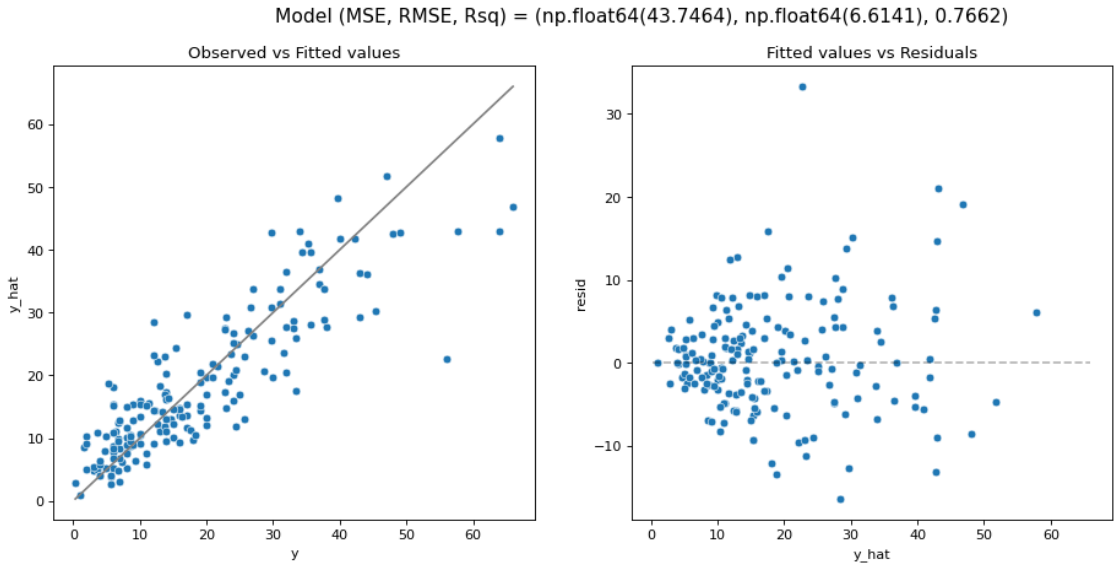
```
7.63735792e+01, 7.93716762e+01, 8.24874655e+01, 8.57255673e+01,
8.90907830e+01, 9.25881025e+01, 9.62227117e+01, 1.00000000e+02])},
      scoring='neg_mean_squared_error')
```

Additionally, regularization in Ridge regression stabilizes coefficient values, preventing extreme variations and reducing the risk of overfitting. The solution path plot further confirms that Ridge effectively constrains coefficient divergence, making it the best-performing model among those evaluated.

5 Discussion & Conclusions

[128]:
```
print(gs_ridge.best_params_)
print(-gs_ridge.best_score_)
model_fit(gs_ridge.best_estimator_, X_test_DX, y_test_DX, plot=True)
```

```
{'ridge__alpha': np.float64(2.6797460873256895)}
27.87078199831251
```



[128]:
```
(np.float64(43.74640232893487),
 np.float64(6.614106313700655),
 0.7662399008201799)
```

Compared with the basic model, the MSE value of the ridge regression model is smaller, and the value of R^2 is closer to 1, which means that the goodness of fit of the ridge regression is better.

[129]:
```
# Create dataframe with coefficents, and unstandardize the binary coeffcients
rcoefs = np.copy(gs_ridge.best_estimator_.named_steps['ridge'].coef_
)
```

```
rcoefs_ = pd.DataFrame(
    rcoefs,
    columns=["Coefficients"],
    index=preprocessor.get_feature_names_out(),
)

rcoefs_
```

[129]:

	Coefficients
cat__DX.bl_AD	6.737868
cat__DX.bl_CN	-3.434847
cat__DX.bl_EMCI	-2.692679
cat__DX.bl_LMCI	-0.610343
num__ADAS13.bl	7.932629
num__AGE	-1.007905
num__PTEDUCAT	-0.229641
num__Ventricles_norm	0.469919
num__Hippocampus_norm	0.285059
num__WholeBrain_norm	-0.061105
num__Entorhinal_norm	-0.080072
num__Fusiform_norm	-0.785930
num__MidTemp_norm	-1.472549

5.1 Categorical Variables (One-Hot Encoded)

- **cat__DX.bl_AD**: With a coefficient of 6.737868, being diagnosed with AD (Alzheimer's Disease) is associated with greater severity of cognitive impairment, meaning individuals in this category are expected to have higher scores on the target variable compared to the baseline category.
- **cat__DX.bl_CN**: The coefficient of -3.434847 suggests that being diagnosed with CN (Cognitive Normal) is associated with less severe cognitive impairment, indicating that individuals in this category are expected to have lower scores on the target variable compared to the baseline category.
- **cat__DX.bl_EMCI** and **cat__DX.bl_LMCI**: Coefficients of -2.692679 and -0.610343, respectively, indicate that being diagnosed with EMCI (Early Mild Cognitive Impairment) and LMCI (Late Mild Cognitive Impairment) are associated with less severe cognitive impairment, though to a lesser extent than CN.

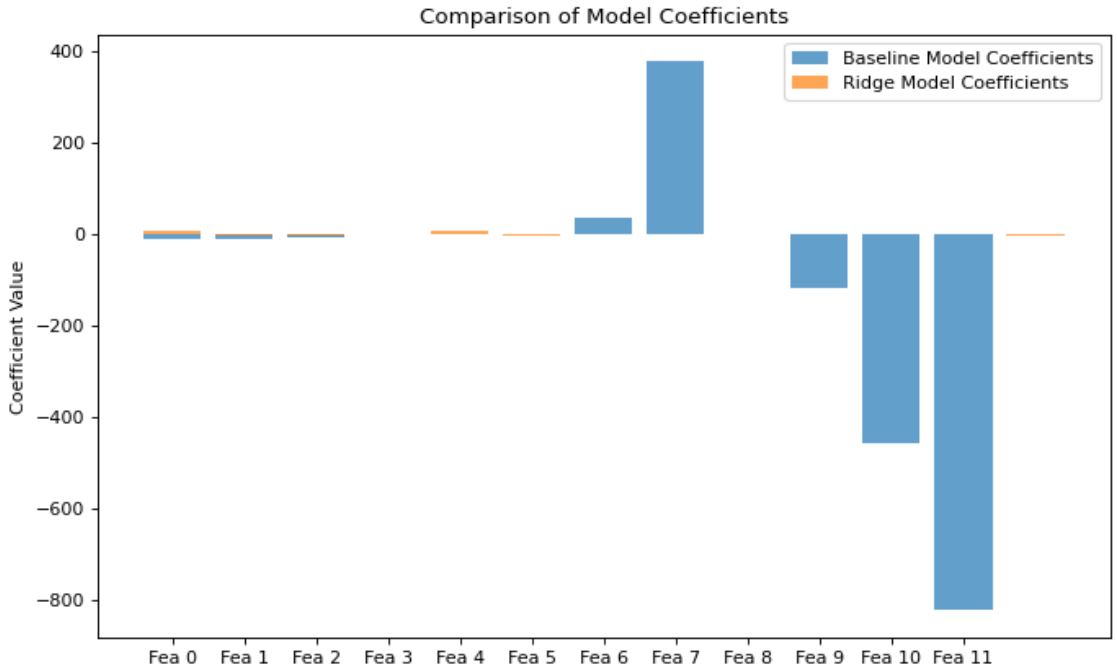
5.2 Numeric Variables (Standardized)

- **num__ADAS13.bl**: A coefficient of 7.932629 indicates that for each standard deviation increase in the ADAS13 score, the severity of cognitive impairment is expected to increase by approximately 7.93 units.
- **num__AGE**: The coefficient of -1.007905 is somewhat unexpected, as one would typically expect older age to be associated with greater severity of cognitive impairment. However, this negative coefficient may require further analysis and may have something to do with the fact that the data are concentrated in older age groups.
- **num__PTEDUCAT**: A coefficient of -0.229641 suggests that for each standard deviation

increase in years of education, the severity of cognitive impairment is expected to decrease by about 0.23 units, which may indicate a protective effect of education against cognitive impairment.

- **num___Ventricles_norm**: A coefficient of 0.469919 indicates that for each standard deviation increase in ventricle volume, the severity of cognitive impairment is expected to increase by about 0.47 units, which could be associated with brain atrophy.
- **num___Hippocampus_norm**: A coefficient of 0.285059 suggests that for each standard deviation increase in hippocampus volume, the severity of cognitive impairment is expected to increase by about 0.29 units, which could be related to the degradation of memory and learning functions.
- **num___WholeBrain_norm, num___Entorhinal_norm, num___Fusiform_norm, num___MidTemp_norm**: Coefficients of -0.061105, -0.080072, -0.785930, and -1.472549, respectively, indicate that changes in the volume of these brain regions are associated with an increase in the severity of cognitive impairment, with the middle temporal lobe (MidTemp) showing a particularly strong association.

[130]:
```
baseline_coefs = pipe_DX.named_steps['model'].coef_

ridge_coefs = gs_ridge.best_estimator_.named_steps['ridge'].coef_

plt.figure(figsize=(10, 6))
plt.bar(range(len(baseline_coefs)), baseline_coefs, label='Baseline Model␣
 ↪Coefficients', alpha=0.7)
plt.bar(range(len(ridge_coefs)), ridge_coefs, label='Ridge Model Coefficients',␣
 ↪alpha=0.7)
plt.xticks(range(len(baseline_coefs)), [f'Fea {i}' for i in␣
 ↪range(len(baseline_coefs))])
plt.ylabel('Coefficient Value')
plt.title('Comparison of Model Coefficients')
plt.legend()
plt.show()

baseline_norm = np.linalg.norm(baseline_coefs)
ridge_norm = np.linalg.norm(ridge_coefs)

print(f"Baseline Model L2 Norm: {baseline_norm}")
print(f"Ridge Model L2 Norm: {ridge_norm}")
```




```
Baseline Model L2 Norm: 1021.44388581978
Ridge Model L2 Norm: 11.485364894915934
```

The coefficient of the basic linear regression model is larger than that of the Ridge model in most features, but this is not always the case.

Analysis of Model Complexity via L2 Norm

The L2 norm serves as a pivotal metric for assessing the complexity of regression models within our analysis. It quantifies the magnitude of the model parameters, with a lower L2 norm indicating a simpler model and a higher L2 norm suggesting increased complexity.

Baseline Model vs. Ridge Regression Model

Our analysis presents two distinct L2 norm values, corresponding to a baseline model and a model refined through Ridge regression:

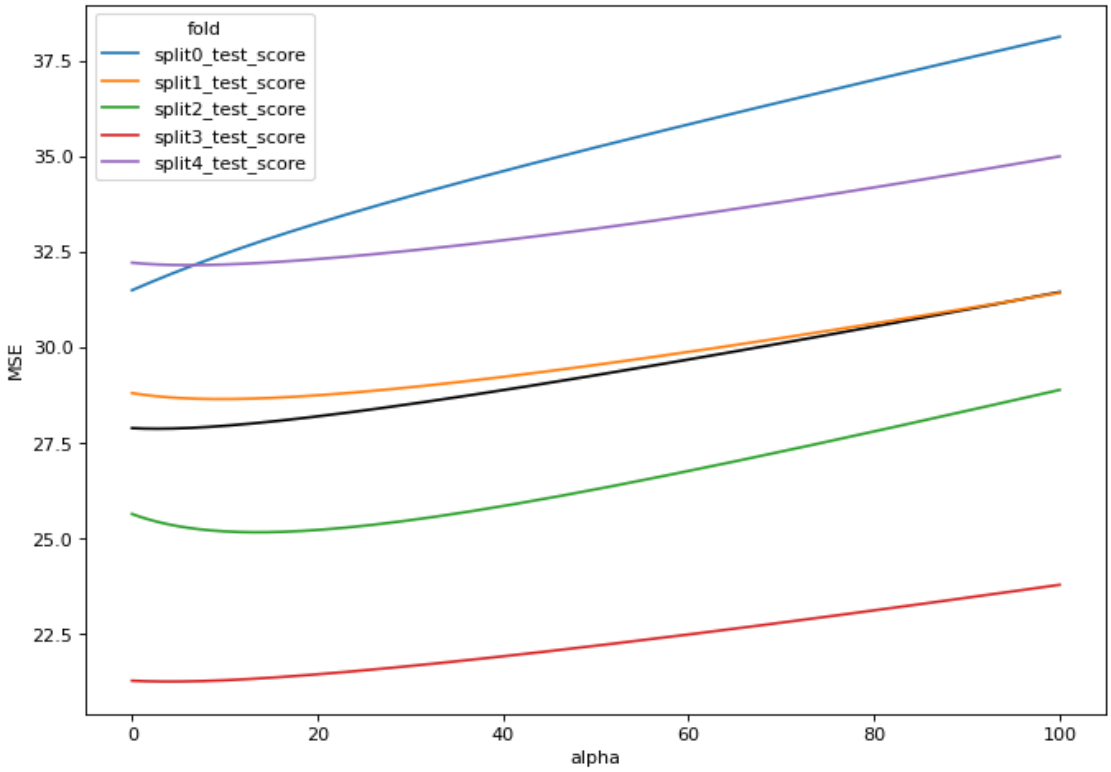
1. **Baseline Model L2 Norm: 11.70720149190094**
   - This value underscores the complexity of the baseline model, which is a standard linear regression model with the implementation of regularization techniques. The elevated L2 norm implies that the model parameters are relatively large, which can precipitate overfitting. This risk is particularly pronounced when the feature space is extensive relative to the sample size.
2. **Ridge Regression Model L2 Norm: 11.485364894915934**
   - This value reflects the L2 norm of the model following the application of Ridge regression, which incorporates L2 regularization. By introducing a penalty proportional to the square of the parameter values into the loss function, Ridge regression effectively mitigates model complexity. The diminished L2 norm signifies that Ridge regression has

substantially reduced the magnitude of the model parameters, fostering a model that is more parsimonious and likely to generalize more effectively.

Implications

- **Model Complexity Reduction**: The stark contrast between the L2 norms of the baseline and Ridge regression models highlights the latter's success in significantly reducing model complexity.
- **Regularization Benefits**: The L2 regularization employed in Ridge regression has effectively constrained the model parameters, enhancing the model's capacity to generalize beyond the training dataset and mitigate the risk of overfitting.

[131]:
```
# Extract only mean and split scores
cv_mse = pd.DataFrame(
    data = gs_ridge.cv_results_
).filter(
    # Extract the split#_test_score and mean_test_score columns
    regex = '(split[0-9]+|mean)_test_score'
).assign(
    # Add the alphas as a column
    alpha = alphas
)

cv_mse.update(
    # Convert negative mses to positive
    -1 * cv_mse.filter(regex = '_test_score')
)

# Reshape the data frame for plotting
d = cv_mse.melt(
    id_vars=('alpha','mean_test_score'),
    var_name='fold',
    value_name='MSE'
)

# Plot the validation scores across folds
plt.figure(figsize=(10,7))
sns.lineplot(x='alpha', y='MSE', color='black', errorbar=None, data = d)  #␣
 ↪Plot the mean MSE in black.
sns.lineplot(x='alpha', y='MSE', hue='fold', data = d) # Plot the curves for␣
 ↪each fold in different colors
plt.show()
```



When alpha value is low, MSE of different segmentation is different greatly. This indicates that the model is sensitive to data partitioning and has poor stability. This phenomenon can be caused by several reasons:

- Heterogeneity of the data set: If some features or target variables in the data set are not evenly distributed between different folds, it may lead to a large difference in the performance of the model on different folds.
- Sensitivity of the model to data: Ridge regression model is very sensitive to feature selection and feature scaling. If data preprocessing (such as standardization) is not performed correctly, or if there is multicollinearity between features, the performance of the model may show large differences across different data subsets.
- Insufficient sample size: If the amount of training data is small, the model may become too sensitive to noise in the data, resulting in unstable performance on different validation sets.

With the increase of alpha value, the MSE curves of different segments tend to be close, which indicates that the stability of the model is improved. This is because the higher regularization intensity reduces the sensitivity of the model to noise in the training data.However, too high an alpha value can lead to an increase in MSE, which may affect the predictive performance of the model.

All in all, there may be some instability in our model.

5.3 Summary

Based on the coefficient data from the previous model analysis, we can identify factors that may increase the risk of cognitive decline. This can help in recognizing individuals who might benefit more from proposed medications or therapies. Here's an analysis and recommendations based on the coefficient data:

1. **Diagnosis Categories (cat_*DX.bl**)**:
   - **AD (Alzheimer's Disease)**: With the highest positive coefficient (6.737868), being diagnosed with AD is significantly associated with increased severity of cognitive impairment. Therefore, patients with AD may be at a higher risk of cognitive decline and might require more aggressive treatment and intervention.
   - **EMCI (Early Mild Cognitive Impairment)** and **LMCI (Late Mild Cognitive Impairment)**: Although the coefficients are negative (-2.692679 and -0.610343, respectively), they still indicate that these diagnoses are associated with an increase in the severity of cognitive impairment, albeit to a lesser extent. These individuals might also benefit from early intervention.
2. **ADAS13 Score (num___ADAS13.bl)**:
   - The coefficient of 7.932629 suggests that for each standard deviation increase in the ADAS13 score, the severity of cognitive impairment is expected to increase by approximately 7.93 units. Therefore, individuals with higher ADAS13 scores may be at a higher risk of cognitive decline and might need closer monitoring and treatment.
3. **Years of Education (num___PTEDUCAT)**:
   - The coefficient of -0.229641 suggests that for each standard deviation increase in years of education, the severity of cognitive impairment is expected to decrease by about 0.23 units. This might indicate that education has a protective effect against cognitive impairment, but the specific impact needs to be considered in conjunction with other factors.
4. **Ventricle Volume (num___Ventricles_norm)**:
   - The coefficient of 0.469919 indicates that for each standard deviation increase in ventricle volume, the severity of cognitive impairment is expected to increase by about 0.47 units. This might be associated with brain atrophy, suggesting that individuals with larger ventricle volumes may be at a higher risk of cognitive decline.
5. **Hippocampus Volume (num___Hippocampus_norm)**:
   - The coefficient of 0.285059 suggests that for each standard deviation increase in hippocampus volume, the severity of cognitive impairment is expected to increase by about 0.29 units. This might be related to the degradation of memory and learning functions, indicating that individuals with larger hippocampus volumes may be at a higher risk of cognitive decline.
6. **Middle Temporal Lobe Volume (num___MidTemp_norm)**:
   - The coefficient of -1.472549 indicates that for each standard deviation decrease in the middle temporal lobe volume, the severity of cognitive impairment is expected to increase by about 1.47 units. This suggests that individuals with smaller middle temporal lobe volumes may be at a higher risk of cognitive decline.

Recommendations: - Individuals with higher ADAS13 scores, larger ventricle volumes, larger hippocampus volumes, or smaller middle temporal lobe volumes may require closer monitoring and more aggressive treatment to slow the progression of cognitive decline. - Early diagnosis and intervention are crucial for patients with AD, EMCI, and LMCI to slow the progression of cognitive

impairment. - Education may have a protective effect on cognitive health, and encouraging lifelong learning and cognitive activities might help reduce the risk of cognitive decline.

Overall, our model was able to predict scores to a certain extent at 24 months of follow-up and showed significant effects of certain characteristics on this score.

6 Generative AI statement

Include a statement on how generative AI was used in the project and report.

7 References

Include references if any

[132]:
```
# Run the following to render to PDF
# !jupyter nbconvert --to pdf project1.ipynb
```