MLPy - Project 1:

Modelling Risk Factors Associated With Dementia using SHARE dataset - A Report.

Authors: Anagha Indulal Nair (S2596158), Navya Thampan (S2602687), Nayem Dewan (S2613404), Shon Kurian George (S2572046)

1. Introduction

The word "Dementia" has long been misunderstood and mischaracterized by stigmatizing cultural beliefs as a curse or punishment and is considered to be neither treatable nor preventable. In today's reality, it is simply an umbrella term for the impaired ability to remember, think, or make decisions that interfere with doing everyday activities [3]. It is not a normal part of aging, although it is known to predominantly affect individuals over 65 years of age [4]. Dementia manifests in various forms, with Alzheimer’s Disease being the most common [8]. The other major types include Vascular dementia, dementia with Lewy bodies, Mixed dementia, and Huntington’s disease (inherited) [9].

It isn’t an understatement to say that dementia is the greatest health and social care challenge globally in the 21st Century. In 2020, around 50 million people live with dementia, and projections estimate that this number will rise to 152 million by 2050 [7]. In 2015, the estimated global cost of dementia was US$818 billion [10]. The point of interest is that 85% of costs are related to family and social aspects instead of medical expenses or care [7]. Appropriate social and lifestyle interventions focused on preventing or reducing the associated impact of risk factors by even a small amount could result in substantial monetary savings for governments and other bodies [9].

Research has established that neurocognitive decline and hence lower cognitive reserve, is linked to a higher risk of developing dementia [8]. This association has allowed the team to hypothesize the use of cognitive ability (cogscore variable in easyshare.csv dataset) as a proxy to determine the severity of dementia in an individual. The cogscore metric was generated by combining the results from five cognitive function tests available in the easyshare all.csv dataset. The tests that were conducted could be considered as parallels to the ones that were shown by [5] to have good properties for detecting dementia.

Dementia is heterogeneous, with great variation in risk factors, some of which coexist for different types. Livingston et al [9] [10] present the case that the 12 modifiable risk factors that they have identified account for 40% of all worldwide dementias. This offers encouraging potential to consequently prevent and delay these cases. A machine learning model can be designed which uses existing data and takes these risk factors into account to help determine what are the most effective and amenable interventions which can be enforced to tackle dementia. Limitations of this model have also been addressed in the discussion section for a realistic expectation of the scope of the model.

The team endeavored to build a model which could predict and capture the underlying relationship between cognitive score and the risk factors associated with dementia, which are available in the SHARE (Survey of Health, Ageing and Retirement in Europe) dataset provided to us. The Lancet Commissions for 2017 [9] and 2020 [10] which focused on the prevention, intervention, and care of dementia, were utilized as the main foundation to gain domain knowledge to achieve the task at hand. The focal point of the Lancet Commissions was on the estimation of the Population Attributable Factor (PAF) [9]. It is a measure of the percentage reduction in new cases over a period of time with the elimination of a particular risk factor. Livingston et al have described the calculation of PAF in their 2017 Lancet Commission. This metric will be the main basis for the interpretation of the model findings and predictions.

Based on the type and the intensity of the relationships that exist between cognitive function and various modifiable risk variables, conclusions can certainly be made from this analysis. For example, the analysis may indicate that specific lifestyle factors such as physical health, daily nutrition, social life, etc., substantially prevent cognitive loss. Additionally, other factors such as obesity, hypertension, and smoking could contribute to the higher possibility of dementia.

The conclusion of this project’s analysis can guide and help the public make informed decisions in terms of individual lifestyle choices. It can support the government in implementing better health policies to prevent dementia. However, it is essential to understand the limitations of the ML model and the data. Hence, these conclusions should be taken as insights that can potentially play a part in the prevention of dementia rather than just casual patterns or trends.

On the basis of substantial research employing polynomial regression models to comprehend their intricate interplay, this paper emphasizes the need of attending to mental health, lifestyle choices, and education as critical elements impacting cognitive health and dementia risk. It highlights the potential of these regions to improve brain health and reduce the development of dementia, and proposes specific interventions in them. Medical, behavioral, and lifestyle changes should all be part of a comprehensive strategy to reduce the risk of dementia, according to the results.

Key words: Dementia, Livingston et al, modifiable risk factors, PAF, SHARE dataset, Lancet Commission, lifestyle interventions

These are the available variables given in the dataset easyshare.csv:

- mergeid - person identifier
- wave - wave identifier
- country - country identifier
- country mod - modified country identifier
- female - dummy encoded gender with 0 for male and 1 for female
- age - age at interview

- birth country - country of birth
- citizenship - citizenship of respondent
- isced1997 r - ISCED-97 encoding of education (6 levels - see pg. 11 of data guide)
- eduyears mod - years of education
- eurod - depression scale ranging from 0 “not depressed” to 12 “very depressed”
- bmi - body mass index
- bmi2 - categorized body mass index
- smoking - smoke at present time
- ever smoked - ever smoked daily
- br010 mod - drinking behavior
- br015 - vigorous activities
- casp - CASP-12 score measures quality of life and is based on four subscales on control, autonomy, pleasure and self-realization, ranges from 12 to 48
- chronic mod - number of chronic diseases
- sp008 - gives help to others outside the household
- ch001 - number of children
- cogscore - measure of cognitive function combining results from two numeracy tests, two word recall tests, and an orientation test.

2. Exploratory Data Analysis and Feature Engineering

EDA is a crucial part of this report as it will help us gain some insights into what the relationship between each of of the variables (features) is and the target variable (cogscore). To achieve this task we will be using two techniques:

1. Using boxplots and lineplots
2. Principal Component Analysis (PCA)

In [2]:

```
data = pd.read_csv("easyshare.csv") # Load data from easyshare.csv and viewing first 3 lines
data.head(3)
```

Out[2]:

	mergeid	int_year	wave	country	country_mod	female	age	birth_country	citizenship	isced1997_r	...	bmi2	smoking	ever_smoked	br0
0	AT-000674-01	2011.0	4.0	11.0	40.0	1.0	59.700001	40.0	40.0	5.0	...	2.0	5.0	5.0	
1	AT-001215-01	2011.0	4.0	11.0	40.0	1.0	72.599998	528.0	528.0	5.0	...	3.0	1.0	1.0	
2	AT-001492-01	2011.0	4.0	11.0	40.0	1.0	59.599998	40.0	40.0	3.0	...	2.0	5.0	1.0	

3 rows × 23 columns

> For modelling and further analysis modification to certain variables need to be performed. Manual encoding was done for done for 'smoking' and 'sp008' where the original values of 1 for 'yes' and 5 for 'no' were changed to 1 for 'yes' and 0 for 'no'.

- To deduce the impact of smoking habits, a new variable (smoking_change) was generated using the information from 2 known variables 'smoking' and 'ever_smoked'. Of the 40% of potentially modifiable risk factors related to dementia, the PAF of smoking was estimated by Livingston et al to be 5% which is the third highest risk factor. [4]
- The relationship between chronic diseases and the cogscore variable was explored to assess whether having a specific disease impacts cognitive abilities and thereby affects the severity of dementia. If a particular disease is found to have a significant effect on cognitive function, the findings could lead to recommendations for interventions aimed at preventing or treating that disease in order to enhance cognitive health. [10] Towards this end, a new variable (disease_name) was generated which alters the mapping from numbers to the actual disease name.

In [3]:

```
# Defining a function to categorize smoking changes based on 'smoking' and 'ever_smoked' variables
def categorize_smoking_change(row):
    if pd.isna(row['smoking']) or pd.isna(row['ever_smoked']):
        return 'Unknown'  # Handling missing values
    if row['smoking'] == 1 and row['ever_smoked'] == 1:
        return 'Continued smoking'
    elif row['smoking'] == 5 and row['ever_smoked'] == 1:
        return 'Stopped smoking'
    elif row['smoking'] == 1 and row['ever_smoked'] == 5:
        return 'Picked up smoking'
    elif row['smoking'] == 5 and row['ever_smoked'] == 5:
        return 'Never smoked'
    else:
        return 'Data Issue'

# Creating the new variable - 'smoking_change'
```

```
data['smoking_change'] = data.apply(categorize_smoking_change, axis=1)

disease_mapping = {
    1: 'Heart attack',
    2: 'Hypertension',
    3: 'High cholesterol',
    4: 'Cerebrovascular disease',
    5: 'Diabetes',
    6: 'Chronic lung disease',
    10: 'Cancer',
    11: 'Peptic ulcer',
    12: 'Parkinson disease',
    13: 'Cataracts',
    14: 'Hip fracture'
}

# Mapping to the 'chronic_mod' column
data['disease_name'] = data['chronic_mod'].map(disease_mapping)

# Encoding the 'smoking' column: 1 for yes and 0 for no
data['smoking'] = data['smoking'].apply(lambda x: 1 if x == 1 else 0)
data['sp008_'] = data['sp008_'].apply(lambda x: 1 if x == 1 else 0)
```

It is to be noted that due to the discrepancy of information and its availability lacking in other waves, the mapping for the other diseases was ignored in the disease_name variable. They are:

```
7. Asthma : only available in waves 1, 2
8. Arthritis : only available in waves 1, 2, 4
9. Osteoporosis : only available in waves 1, 2
15. Other fractures : only available in wave 4, 5, 6, 7, 8
16. Alzheimer’s (…) : only available in wave 4, 5, 6, 7, 8
17. Benign tumor (…) : only available in wave 2
18. Other affective (…) : only available in wave 5, 6, 7, 8
19. Rheumatoid Arthritis : only available in wave 5, 6, 7, 8
20. Osteoarthritis (…) : only available in wave 5, 6, 7, 8
21. Chronic kidney disease: only available in wave 6, 7, 8
```

> To exhibit the nature of the variables within the dataframe, info() was used.

In [4]:

```
data_info = data.info() # Data structure
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 98348 entries, 0 to 98347
Data columns (total 25 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   mergeid         97372 non-null  object 
 1   int_year        97372 non-null  float64
 2   wave            97372 non-null  float64
 3   country         97372 non-null  float64
 4   country_mod     97372 non-null  float64
 5   female          97372 non-null  float64
 6   age             97372 non-null  float64
 7   birth_country   97170 non-null  float64
 8   citizenship     97281 non-null  float64
 9   isced1997_r     97372 non-null  float64
 10  eduyears_mod    84532 non-null  float64
 11  eurod           95838 non-null  float64
 12  bmi             94686 non-null  float64
 13  bmi2            94686 non-null  float64
 14  smoking         98348 non-null  int64  
 15  ever_smoked     97116 non-null  float64
 16  br010_mod       79210 non-null  float64
 17  br015_          97113 non-null  float64
 18  casp            84065 non-null  float64
 19  chronic_mod     97283 non-null  float64
 20  sp008_          98348 non-null  int64  
 21  ch001_          96928 non-null  float64
 22  cogscore        97372 non-null  float64
 23  smoking_change  98348 non-null  object 
 24  disease_name    66409 non-null  object 
dtypes: float64(20), int64(2), object(3)
memory usage: 18.8+ MB
```

> The number of missing values were identified for each variable and displayed.

In [5]:

```
# Checking for missing values
missing_values = data.isnull().sum()

# Descriptive statistics for numerical variables can be viewed using the command below
# print(data.describe())
```

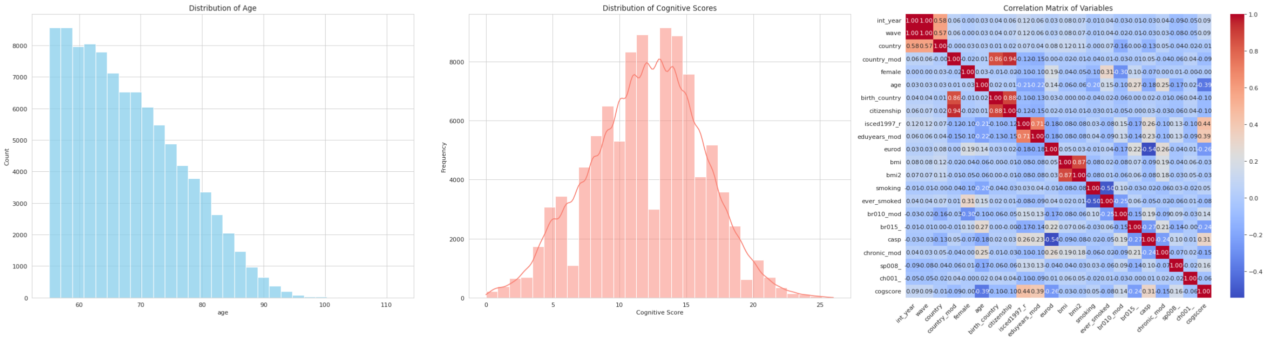
```
print("The missing values in each variable are\n",missing_values)
```

```
The missing values in each variable are
 mergeid              976
int_year             976
wave                 976
country              976
country_mod          976
female               976
age                  976
birth_country       1178
citizenship         1067
isced1997_r          976
eduyears_mod       13816
eurod               2510
bmi                 3662
bmi2                3662
smoking                0
ever_smoked         1232
br010_mod          19138
br015_              1235
casp               14283
chronic_mod         1065
sp008_                 0
ch001_              1420
cogscore             976
smoking_change         0
disease_name       31939
dtype: int64
```

> We will now visualise the distribution of age and cogscore variables using histograms and the correlation between all the variables of the easyshare.csv file for feature selection to look for any discrepancies or skewness in the dataset.

In [6]:

```
# Seting the style of plots
sns.set_style("whitegrid")

# Creating 1x3 grid of subplots
fig, axs = plt.subplots(1, 3, figsize=(30, 8))

# Plotting distribution of Age
sns.histplot(data['age'], bins=30, ax=axs[0], color='skyblue')
axs[0].set_title('Distribution of Age')

# Plotting distribution of Cognitive Scores
sns.histplot(data['cogscore'], kde=True, bins=30, ax=axs[1], color='salmon')
axs[1].set_title('Distribution of Cognitive Scores')
axs[1].set_xlabel('Cognitive Score')
axs[1].set_ylabel('Frequency')

# Excluding non-numeric columns
numeric_data = data.select_dtypes(include=[np.number])

# Calculating the correlation matrix
correlation_matrix = numeric_data.corr()

# Plotting Correlation Matrix
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm', ax=axs[2])
axs[2].set_title('Correlation Matrix of Variables')

# Adjusting x and y labels
plt.setp(axs[2].get_xticklabels(), rotation=45, ha="right", rotation_mode="anchor")
plt.setp(axs[2].get_yticklabels(), rotation=0)
# Adjusting the layout
plt.tight_layout()
plt.show()
```



- The distribution of age is within the range of 55 to 111 with an mean age of 67.
- Cognitive scores tends to follow a normal distribution with a mean score of 12.

- From the correlation matrix the trends can observed amongst the variables and with respect to the cogscore which can help in detecting multicollinearity, if present.

Selection of Useful Modifiable Risk Factors

In order to retrieve useful analyses, certain variables have to be filtered out. This step was taken by deciding between useful and non-useful variables. Hence, at this stage variables were segregated based on domain knowledge and key findings from previous visualizations. The aim is to focus on the modifiable risk factors which could help governmental bodies and other organization to potentially intervene. Additionally, the non-useful variables would be excluded from the Principal Component Analysis (PCA) that is to follow.

The variables that were not used are as follows:

- Irrelevant non-modifiable variables: wave, country_mod, citizenship, female, age, birth_country, ch001.

These variables were relevant but were not used for the modeling:

- isced1997 r - Already accounted for by the variable eduyears_mod, with a very high positive correlation score of 0.71.
- bmi - Already accounted for in the variable bmi2, with a very high correlation score of 0.87.
- ever_smoked - Not used for further analysis due to the ambiguity of the information
- casp - a quality of life metric which has a high correlation score of -0.54 with eurod. Due to the the complexity and limited scope for usage, eurod was preferred as it retrieves information on a broad spectrum making it more suitable for analysis.
- chronic_mod - disease_name replaces the information from this variable which simplified the model by reducing redundant complexity without losing useful insights. However, the representation from people with complicated needs is lacking in clinical trials since the participants for this study are usually healthier young men who are more educated. [11]

These variables can be used to identify interesting relationships with cogscore but won't be directly used in the model:

- smoking_change: smoking_change can be vital for thorough analysis but it needs multinomial regression which currently is beyond the analytical scope of this project.
- disease_name: 31,939 missing values which is too much loss of information.

The following variables have been selected on the basis of support of literature and univariate analysis:

- sp008_: NICE and NIH claim that the social isolation and hearing loss (peripheral) can be risk factors that can contribute to dementia but are factors that can be potentially modified. An analysis centered on social isolation and its relation with incident dementia was done to understand its Population Attribute Fraction (PAF). This research shed more light in three areas: Social contact (this involved phone or face-to-face interactions with family and friends), a social engagement (participation in community activities), and loneliness (discontentment due to isolation from social life). [12] Effective indicator of social engagement: Social isolation can lead to cognitive inactivity, which can be related to accelerated cognitive deterioration and low mood. These are factors that risk the chance of dementia occuring, signifying the necessity of considering social engagement of elderly people along with their health and mental wellbeing. [12]

- br015_: It is observed that exercise can provide neuroprotective benefits and encourage the release of brain-derived neurotrophic factor (BDNF) which can lower cortisol levels and reduce vascular risk. Despite this, exercise alone does not seem to enhance cognitive function in elderly people who are healthier. [13] [14] [15]

- eurod: Studies have shown links of repeated depressive episodes to higher risk of dementia proving that depression is a potential risk factor. It is possible that depression could increase the risk of dementia by affecting stress hormones, growth of neurons and hippocampal size. [16] Additionally, antidepressant prescriptions can impact the rate of dementia while some studies conducted on animals suggest that antidepressants like citalopram could lower amyloid production. [17] [18] [19]

- eduyears_mod: With a global estimation of prevalence around 40%, a lower level of education correlated with dementia relative risk (RR) of 1.59 (95% CI 1.26–2.01). [8] With limited education, this variable ranks as the second largest Population Attributable Fraction (PAF) in the analysis. Although the preventive impact of post secondary education is not known, it is thought that the lower educational levels push the risk of cognitive deterioration up high by decreasing cognitive reserve which enables the preservation of cognitive functions despite brain abnormalities. [20]

- bmi2: Obesity can be linked with pre-diabetes and metabolic syndrome, indicated by insulin resistance levels and higher levels of peripheral insulin. The discrepancies in the peripheral insulin contribute to less insulin in the brain which obstructs clearance of amyloid. Moreover, the higher inflammation and high levels of glucose in blood can contribute to diabetes which can negatively affect the cognitive function. [21] [22]

- smoking/smoking_change: Smoking can be linked to cardiovascular diseases. However, cigarette smoke can produce neurotoxins that can make the cardiovascular health at greater risk. Moreover, the high prevalence of smoking can provide higher Population Attributable Fraction (PAF). [23]

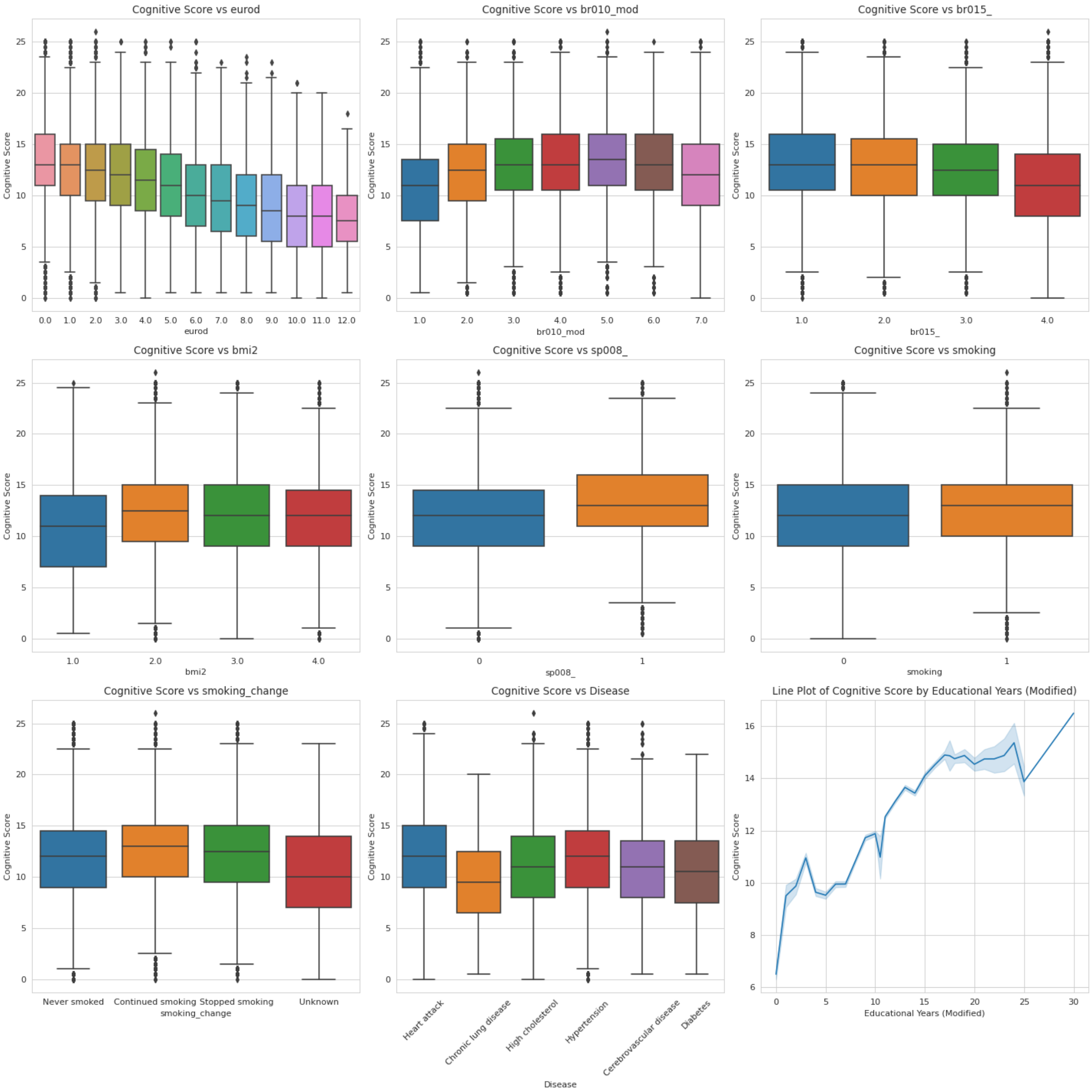
- br010_mod: Heavy drinking is associated with brain changes, cognitive impairment, and dementia, a risk known for centuries. [24] An increasing body of evidence is emerging on alcohol’s complex relationship with cognition and dementia outcomes from a variety of sources including detailed cohorts and large-scale record based studies.

Graphical Analysis

The graphical representation of the useful variables are represented using boxplots and a line plot.

In [7]:

```python
# Defining the variables
variables = ['eurod', 'br010_mod', 'br015_', 'bmi2', 'sp008_', 'smoking', 'smoking_change']

# 3x3 grid subplot
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(18, 18))
axes = axes.flatten()

# Iterating over the variables and creating box plots
for i, var in enumerate(variables):
    sns.boxplot(x=var, y='cogscore', data=data, ax=axes[i])
    axes[i].set_title(f'Cognitive Score vs {var}')
    axes[i].set_xlabel(var)
    axes[i].set_ylabel('Cognitive Score')

# Boxplot for 'disease_name' with 'cogscore'
sns.boxplot(x='disease_name', y='cogscore', data=data, ax=axes[7])
axes[7].set_title('Cognitive Score vs Disease')
axes[7].set_xlabel('Disease')
axes[7].set_ylabel('Cognitive Score')
axes[7].tick_params(axis='x', rotation=45)

# Line plot for 'eduyears_mod' vs 'cogscore'
sns.lineplot(x='eduyears_mod', y='cogscore', data=data, estimator='mean', ax=axes[8])
axes[8].set_title('Line Plot of Cognitive Score by Educational Years (Modified)')
axes[8].set_xlabel('Educational Years (Modified)')
axes[8].set_ylabel('Cognitive Score')

# Adjusting the layout
plt.tight_layout()
plt.show()
```



The observations of the plots are as follows:

- eurod: There is an evident negative correlation between ‘cogscore’ and ‘eurod’, which is consistent with the literature. This implies that it could significantly impact the model and help predict dementia.
- br010_mod: It is visible that the individuals who drink frequently in a week do not exhibit lower cognitive scores. However, excessive drinking results in cognitive decline, which, again, is supported by the literature.
- br015_: It is also observed that the individuals who engage in regular physical activity tend to be healthier, maintain higher ‘cogscore’ and sustain a stronger social connection. These aspects are known for their influence on cogscore and dementia risk, a relationship that has been confirmed by studies.
- bmi2: In this categorization of bmi into 4 levels, underweight and obese individuals exhibit a decline in cognitive score, due to poor lifestyle and dietary habits, impacting cognitive functions. On the other hand, individuals with normal bmi or those who are overweight tend to have higher cogscores, likely due to adherence due to healthier lifestyle practices.
- smoking: The median cognitive scores show a slight difference between smokers and non-smokers, with non-smokers achieving a marginally higher median score. However, the spread of scores and occurrence of outliers appear similar for both groups.
- sp008: Individuals who offer assistance to others beyond their own household tend to show higher cognitive scores compared to the ones who don’t. Although, the distribution of scores and presence of outliers are similar among the two groups.
- smoking_change: Non-smokers exhibit highest median cognitive scores, with the ones who stopped smoking trailing slightly behind. The individuals who continued smoking show lower median scores, whereas the group with unknown smoking status records the lowest scores. The data does not present any discernible trends.
- disease_name: The median cognitive scores vary across different diseases, with individuals suffering from heart attack and chronic lung diseases exhibiting slightly lower median scores compared to those with conditions such as cholesterol and hypertension. Nonetheless, the range and occurrence of outliers remain consistent across various diseases.
- eduyears_mod: There exists a distinct positive pattern, indicating that an increase in years of education correlates with higher cognitive scores. The confidence interval surrounding the line plot expands with rise in education years, suggesting significant variability in cognitive scores among those with higher levels of education.

Principal Component Analysis

Principal component analysis (PCA) is a method used to reduce the number of dimensions in a dataset. It is commonly used in exploratory data analysis, visualization, and data preparation. In this context, it will be used to validate how much the variables contribute to the variablitity in the data with respect to cogscore.

In [8]:

```
# Variables we need to do PCA
variables_for_pca = ['eurod', 'br010_mod', 'br015_', 'bmi2', 'sp008_', 'smoking', 'eduyears_mod', 'cogscore']

# Creating a new dataframe for PCA
data_for_pca = data[variables_for_pca]
```

Since PCA cannot deal with missing/NA values, they are removed from the dataset. The remaining features are scaled.

In [9]:

```
# Removing rows with any missing/NA values
data_clean = data_for_pca.dropna()

# Separating out the features and the target variable
features = data_clean.drop(columns=['cogscore'])
target = data_clean['cogscore']

# Standardizing the features
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)
```

In [10]:

```
# Applying PCA
pca = PCA()
pca.fit(features_scaled)

# Displaying explained variance
np.cumsum(pca.explained_variance_ratio_)[:7]
```

Out[10]:

```
array([0.22667091, 0.37748764, 0.51659066, 0.6508502 , 0.77470261,
       0.89368894, 1.        ])
```

In [11]:

```
# Explained variance ratio for each principal component
explained_variance_ratio = pca.explained_variance_ratio_

# Cumulative explained variance
cumulative_explained_variance = np.cumsum(explained_variance_ratio)

# Printing the cumulative explained variance
```

```
print(f"Cumulative Explained Variance:")
(cumulative_explained_variance)
```

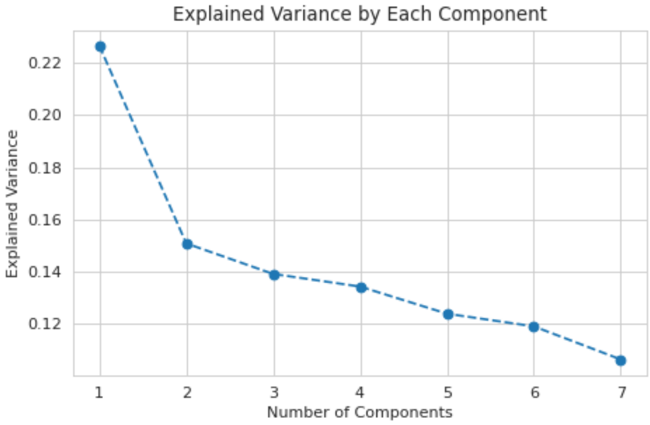
```
Cumulative Explained Variance:
```

Out[11]:

```
array([0.22667091, 0.37748764, 0.51659066, 0.6508502 , 0.77470261,
       0.89368894, 1.        ])
```

In [12]:

```
# Identifying the number of components
n_components = np.where(cumulative_explained_variance >= 0.90)[0][0] + 1

# Using scree plot for the Elbow Point
plt.figure(figsize=(6, 4))
plt.plot(range(1, len(explained_variance_ratio) + 1), explained_variance_ratio, marker='o', linestyle='--')
plt.title('Explained Variance by Each Component')
plt.xlabel('Number of Components')
plt.ylabel('Explained Variance')
plt.grid(True)
plt.tight_layout()
plt.show()

n_components
```



Out[12]: 7

The plot of the cumulative explained variance by the PCA components shows how much variance is captured by the first few components. The first two component explains about 38% of the variance, and with each additional component, more variance is captured.

Typically in health studies, the first two components should capture arounnd 70-90% of the variance in the data. In this context, using just the first two components we are able to capture almost 40% of the variance with just 2 instead of using 5 to reach the typical threshold of 70%. This could be due to the fact that we have utilised a small subset of all the variables of the dataset which are more useful in providing future interventions rather than simply accouting for the variance. Also, cogscore is a complex metric which is affected by many factors.

In [13]:

```
# Rerunning PCA with 2 components
pca_2 = PCA(n_components=2)
pca_2.fit(features_scaled)

# Extracting first 2 components
loadings_2 = pca_2.components_.T

# Creating a DataFrame for the loadings
loadings_2_df = pd.DataFrame(loadings_2, index=features.columns, columns=[f'PC{i+1}' for i in range(2)])
loadings_2_df.head(len(features.columns))
```

Out[13]:

	PC1	PC2
eurod	0.460953	0.291125
br010_mod	-0.436159	0.126958
br015_	0.478798	0.204899
bmi2	0.236334	-0.526422
sp008_	-0.306844	-0.086345
smoking	-0.161572	0.750511
eduyears_mod	-0.438095	-0.096445

- Variables like eurord and br015_ have positive loadings on PC1, suggesting they contribute positively to the variance captured by this component.
- br10_mod, sp008_, smoking, and eduyears_mod have negative loadings on PC1, indicating they contribute inversely to this principal component.
- For PC2, smoking has a notably high positive loading, suggesting a strong unique variance in the direction of this component.
- In contrast, bmi2 has a strong negative loading on PC2.

Overview of Exploratory Data Analysis

The graphical analysis and Principal Component Analysis (PCA) indicate that the selected features exhibit ample characteristics for incorporation into the modelling phases. Notably, the alignment of these variables are consistent with findings documented in the literature, suggesting that the model has the potential to accurately predict the most relevant risk factors linked to the severity of dementia.

The list of features that will be used in the models are as follows:

> eurod, br010_mod, br015_, bmi2, sp008_, smoking, eduyears_mod

Feature Engineering

Here, the observations with missing values in any of the features have been identified and removed.

In [14]:
```
# Features to check NaN values in
features_with_change = ['eurod', 'br010_mod', 'br015_', 'bmi2', 'sp008_', 'smoking', 'eduyears_mod']

# Checking for NaN values in each feature
nan_checking = data[features_with_change].isna().sum()

# Printing the results
print(nan_checking)
```

```
eurod            2510
br010_mod       19138
br015_           1235
bmi2             3662
sp008_              0
smoking             0
eduyears_mod    13816
dtype: int64
```

eurod has 2510 NaN values, br010_mod has around 19138, br015_ has around 1235, bmi2 has around 3662 and the rest of the variables have no NaN values. To understand what could be best approach to deal with non-NaN values, the next step is to check whether dropping those values could be a good approach.

In [15]:
```
# Coutinng non-NaN values present in each feature
non_nan_counts = data[features_with_change].count()

# Printing the results
print(non_nan_counts)
```

```
eurod           95838
br010_mod       79210
br015_          97113
bmi2            94686
sp008_          98348
smoking         98348
eduyears_mod    84532
dtype: int64
```

From above, it is evident that there are many Non-NaN values which are in higher number when compared to the number of NaN values which can be assumed to not affect the data majorly. Since we dont lose out on too much data, NAN values are dropped instead.

In [16]:
```
# Updated feature list for modeling
features_with_change = ['eurod', 'br010_mod', 'br015_', 'bmi2', 'sp008_', 'smoking', 'eduyears_mod']
target_var = ['cogscore']

# Dropping rows with NaN values
data_cleaned = data.dropna(subset=features_with_change)

# Checking for NaN values
nan_check_3 = data_cleaned[features_with_change].isna().sum()

# Printing the results
print(nan_check_3)
```

```
eurod           0
br010_mod       0
br015_          0
bmi2            0
sp008_          0
smoking         0
eduyears_mod    0
dtype: int64
```

The above settings have eliminated the NaN values from the columns of the features.

3. Model Fitting and Tuning

A linear regression model served as the baseline for comparison. Its mean squared error (MSE) and R-squared values were evaluated against those from alternative models, including lasso regression, ridge regression, and polynomial regression.

Splitting Data into Train and Test Sets

In [17]:

```
# Feature list and Target variable
features_with_change = ['eurod', 'br010_mod', 'br015_', 'bmi2', 'sp008_', 'smoking', 'eduyears_mod']
target_var = 'cogscore'

# Splitting the dataset
X = data_cleaned[features_with_change]
y = data_cleaned[target_var]

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

In this model, the dependent variable, cogscore, is the target variable. It can be deduced from test_size that 70% of the data was allocated for training and the remaining 30% was set aside for testing, with a reproducibility seed of 42.

Linear Regression - Baseline Model

In [18]:

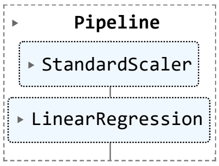
```
# Creating first pipeline for preprocessing & modeling
pipeline = Pipeline([
    ('scaler', StandardScaler()),  # Standardizing the features
    ('regressor', LinearRegression())  # with linear regression model as base model
])

# Model fitting
pipeline.fit(X_train, y_train)

# Predicted values on test
y_pred = pipeline.predict(X_test)

# Model evaluation with MSE and R-squared values
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

#Displaying result
print("Mean Squared Error:%0.3f"% mse)
print("R-squared:%0.3f"% r2)
```

```
Mean Squared Error:13.360
R-squared:0.218
```

In [19]:

```
pipeline
```

Out[19]:



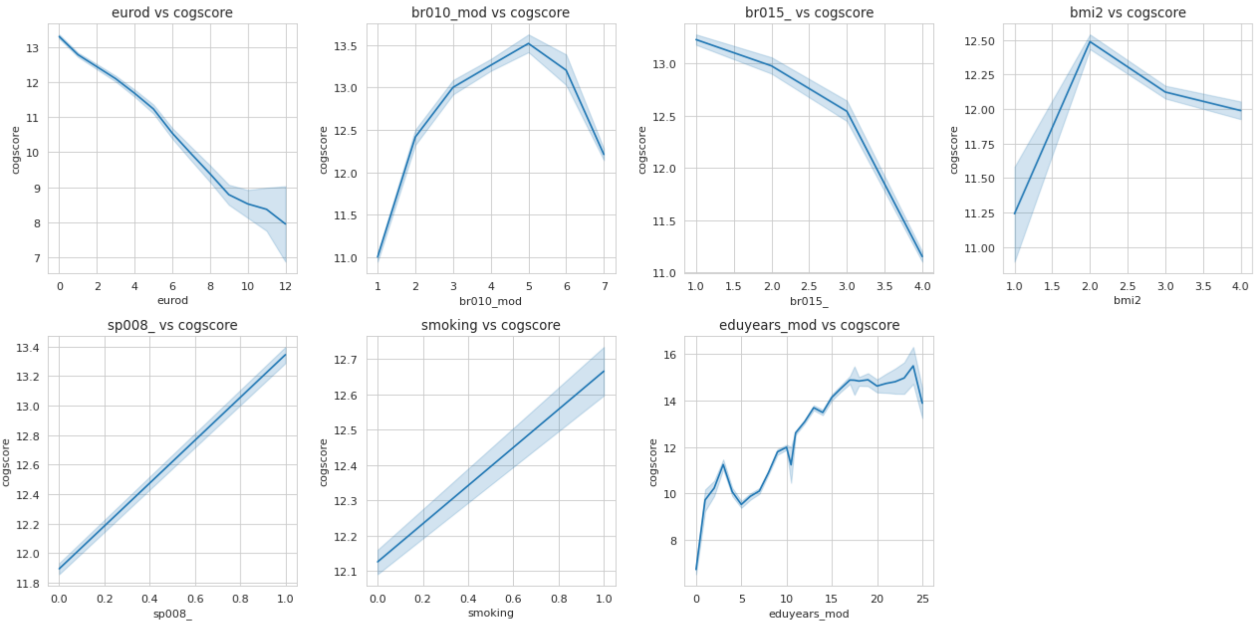
The pipeline referenced in this context represents the linear regression baseline model. It is evident that the linear regression model was fitted and the features were standardized. The predicted test values were subsequently presented alongside the MSE and R-squared values.

According to the MSE value of this fundamental model, the R-squared value indicates that the model explains 21.8% of the variability in the data.

In [20]:

```
# Fitting linear regression model
lm = LinearRegression().fit(X_train, y_train)

# Producing scatterplots for each feature
fig, axes = plt.subplots(nrows=2, ncols=4, figsize=(16, 8))

axes = axes.flatten()

# Generating plots
for i, feature in enumerate(features_with_change):
    sns.lineplot(x=feature, y=target_var, data=data_cleaned, ax=axes[i])

    # Labelling plots
    axes[i].set_title(f'{feature} vs {target_var}')
    axes[i].set_xlabel(feature)
    axes[i].set_ylabel(target_var)

# Removing empty subplots
for j in range(len(features_with_change), len(axes)):
    fig.delaxes(axes[j])
```

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



Each lineplot compares a feature variable against the target variable, ‘cogscore’. Inferences can be drawn from each plot:- eurod vs cogscore: There is a steady declining trend, indicating that "cogscore" value decreases as the value of "eurod" increases.

- br010_mod vs cogscore: There is a steady rise, where the cogscore peaks around 5 on the x-axis which shows that there is an ideal range of 'br010_mod' for the cogscore before it declines again.
- br015_ vs cogscore: Exhibits a decreasing trend, similar to the first plot, where a drop in the ‘cogscore’ is correlated with an increase in ‘br015_’.
- bmi2 vs cogscore: Shows a non-linear relationship as it has two peaks, indicating there exists two distinct ‘bmi2’ ranges where ‘cogscore’ is higher.
- sp008_ vs cogscore: Displays a strong, steady positive linear relationship, implying that as ‘sp008_’ increases, so does ‘cogscore’.
- smoking vs cogscore: Also demonstrates a positive trend, indicating that the ‘cogscore’ rises with an increase in ‘smoking’.

- eduyears_mod vs cogscore: Also displays a positive relationship, but not so steady as the two previous ones.

The confidence intervals, or shaded regions surrounding the blue lines, show how uncertain the mean cogscore estimates are for a given set of independent variable values.

In [21]:

```
# Checking the trained LinearRegression model from the pipeline
linear_regression_model = pipeline.named_steps['regressor']

# Intercept of the linear regression model
intercept = linear_regression_model.intercept_
print("Intercept:%0.3f"% intercept)

# Coefficients of the linear regression model
coefficients = linear_regression_model.coef_
print("Coefficients:")
for feature, coef in zip(features_with_change, coefficients):
    print(f"{feature}: {coef:0.3f}")
```

```
Intercept:12.208
Coefficients:
eurod: -0.649
br010_mod: 0.149
br015_: -0.535
bmi2: 0.035
sp008_: 0.311
smoking: 0.125
eduyears_mod: 1.338
```

The trained regression model printed the associated coefficients and the model's intercept as it traversed each feature. Positive coefficients may indicate that the predictor variable and the target variable are positively correlated.

In [22]:

```
# Concatenating intercept and coefficients into w
# Intercept term of the fitted model
w_0 = intercept
# Coefficient terms of the model
w_1 = coefficients

w = np.concatenate([[w_0], w_1])
print("w:", w)
```

```
w: [12.20763558 -0.64903009  0.14937876 -0.53493066  0.03533171  0.31097844
  0.12507045  1.33840679]
```

The above values show the intercept and coefficients which are part of the estimates in $\boldsymbol{w}$. This was done to show that the exact intercept and other coeffecient values can also be found this way instead of the previous method. This can be a better approach to store and use the entire parameters of the model directly for predictions and other types of analyses.

In [23]:

```
# Fitted values
y_fit = pipeline.predict(X_train)

# Predicted values
y_pred = pipeline.predict(X_test)

# Printing the mean squared error of the training set
print("Training Mean squared error: %.3f" % mean_squared_error(y_train, y_fit))
# Printing R^2 of the training set
print("Training R squared: %.3f" % r2_score(y_train, y_fit))

# Printing mean squared error of the test set
print("Test Mean squared error: %.3f" % mean_squared_error(y_test, y_pred))
# Printing R^2 of the test set
print("Test R squared: %.3f" % r2_score(y_test, y_pred))
```

```
Training Mean squared error: 13.344
Training R squared: 0.209
Test Mean squared error: 13.360
Test R squared: 0.218
```

The MSE and R-squared values for the training and testing datasets provide insights into the linear regression model's performance with regard to data-fitting and on unseen or novel data. The training R-squared value of 0.209 indicates that the model can account for approximately 20.9% of the variability. Conversely, the test R-squared value of 0.218 indicates that the model can explain 21.8% of the variability in relation to unobserved data.

Interpretation Using Predicted Values From the Linear Model

> We are seeing the instances where cogscore in y_pred is less than 5 and comparing it with domain knowledge

In [24]:

```
# Initializing a list to store each sample's prediction, truth, and features
samples = []

# Looping through the samples in y_pred
for i in range(len(y_pred)):
    # Checking if prediction is less than 5
    if y_pred[i] < 5:
        # Retrieving the prediction for the i-th sample
        prediction = y_pred[i]

        # Retrieving the true label for the i-th sample
        truth = y_test.iloc[i]

        # Retrieving the i-th row of features from X_test
        features = X_test.iloc[i]

        # Creating a dictionary for the i-th sample
        sample_data = {'Sample': i+1, 'Prediction': prediction, 'Truth': truth}
        sample_data.update(features.to_dict())

        # Adding to the dictionary to the samples list
        samples.append(sample_data)

        # Break when we have collected 10 samples
        if len(samples) == 10:
            break

# Converting the samples list into a pandas DataFrame
samples_df = pd.DataFrame(samples)

# Displaying the DataFrame
print(samples_df.to_string(index=False))
```

```
Empty DataFrame
Columns: []
Index: []
```

Inferences for model predictions with low cogscore:

The model has demonstrated accurate predictions when compared to the actual values.

- High levels of 'eurod' in samples suggest a greater degree of depression in the individual. The literature demonstrates that it has a substantial impact on cogscore. [16] [17] [18] [19]
- The variable 'eduyears_mod' consistently has a value of 0 for all the samples observed, which aligns with the information found in the literature. [8].
- Samples of the dataset where the value of 'br015_' is high i.e people who hardly exercise are seen to have low cogscore. This is also aligning with our domain knowledge. [13] [14] [15]

- People that don't interact with others often have a low cogscore, as shown by the feature 'sp008_' being 0. One possible explanation is the correlation between social isolation and cognitive decline. [12]
- 'smoking' does not seem to have any effect on cogscore atleast from this subset of values. Although it has been identified to have biological reasons in impacting cogscore. [23]
- No clear pattern can be seen for the other features. This could be due to the fact we are viewing a small subset of the samples.

> We are seeing the instances where cogscore in y_pred is more than 17 and comparing it with domain knowledge

In [25]:

```python
# Initializing a list to store each sample's prediction, truth, and features
samples = []

# Looping through the samples in y_pred
for i in range(len(y_pred)):
    # Checking if the prediction is greater than 17
    if y_pred[i] > 17:
        # Retrieving the prediction for the i-th sample
        prediction = y_pred[i]

        # Retrieving the true label for the i-th sample using
        truth = y_test.iloc[i]

        # Retrieving the i-th row of features from X_test using
        features = X_test.iloc[i]

        # Creating a dictionary for the i-th sample
        sample_data = {'Sample': i+1, 'Prediction': prediction, 'Truth': truth}
        sample_data.update(features.to_dict())

        # Adding to the dictionary to the samples list
        samples.append(sample_data)

        # Break when we have collected 10 samples
        if len(samples) == 10:
            break

# Converting the samples list into a pandas DataFrame
samples_df = pd.DataFrame(samples)

# Displaying the DataFrame
print(samples_df.to_string(index=False))
```

```
 Sample  Prediction  Truth  eurod  br010_mod  br015_  bmi2  sp008_  smoking  eduyears_mod
    125   17.211644   13.5    3.0        7.0     2.0   3.0     0.0      1.0          25.0
    136   17.230618    9.0    0.0        5.0     1.0   4.0     1.0      0.0          20.0
    296   17.295320   12.0    0.0        7.0     1.0   2.0     1.0      1.0          19.0
    645   17.008301   15.0    3.0        3.0     1.0   3.0     0.0      0.0          25.0
    782   18.135474   14.0    0.0        7.0     1.0   3.0     0.0      0.0          25.0
   1020   17.440920   19.5    0.0        2.0     1.0   2.0     0.0      0.0          24.0
   1283   17.420927   23.0    0.0        7.0     2.0   3.0     0.0      0.0          24.0
   1513   17.171085   15.0    2.0        4.0     1.0   3.0     1.0      0.0          22.0
   1719   17.524480   16.5    1.0        5.0     1.0   3.0     1.0      0.0          22.0
   1778   17.980818   18.0    0.0        4.0     1.0   4.0     0.0      0.0          25.0
```

Inferences for model predictions with high cogscore:

The model has demonstrated accurate predictions when compared to the actual values.

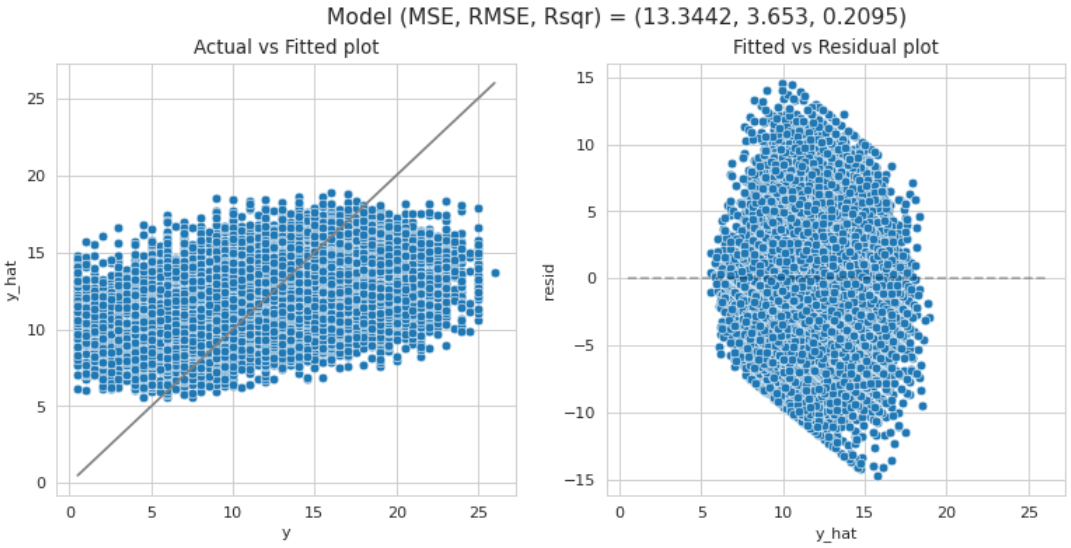
- Low levels of 'eurod' in samples suggest a lesser degree of depression in the individual. The literature demonstrates that it has a substantial impact on cogscore. [16] [17] [18] [19]
- The variable 'eduyears_mod' consistently has a value close to 25 for all the samples observed, which aligns with the information found in the literature. [8].
- Samples of the dataset where the value of 'br015_' is low i.e people who exercise regularly are seen to have high cogscore. This is also aligning with our domain knowledge. [13] [14] [15]
- 'smoking' tends to have a value of 0 here, that is people with high cogscore don't smoke. This is consistent with the literature and often the case with individuals who exercise frequently. [23]
- There is an intriguing high value for 'bmi2' in these samples. These figures may be outliers or there may be some other explanation, even if the research has indicated a negative correlation between them. [21] [22]
- It would appear that br010_mod' with high values indicates that drinking has little effect on people's cognitive abilities. The literature shows that moderate drinking has no effect on the cogscore, whereas high drinking has a noticeable effect. [24]
- No clear pattern can be seen for 'sp008'. This could be due to the fact we are viewing a small subset of the samples.

In [26]:

```python
# The below custom function has been written to generate values of MSE, RMSE and Rsqr and plot Actual vs Fitted
# and Fitted vs Residual plots.
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=(10, 5))

        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("Actual vs Fitted plot")

        plt.subplot(122)
        sns.scatterplot(x='y_hat', y='resid', data=res).set_title("Fitted vs Residual plot")
        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, Rsqr) = " + str(Metrics), fontsize=14)
        plt.show()

    return MSE, RMSE, Rsqr
```

In [27]:

```
# Plotting the model with performance metrics
model_fit(pipeline, X_train, y_train, plot=True)
```



Out[27]:

```
(13.344182041981876, 3.652968935260999, 0.20945047611582923)
```

This Actual vs fitted plot depicts a comparison between the observed values (on the x-axis) and the predicted values or fitted values (on the y-axis) derived from the model. The gray line indicates the line of best fit, when the actual values are exactly equal to the projected values. The dispersion of data points in this graph demonstrates a positive correlation, suggesting that the model possesses a degree of predictive capability. Nevertheless, the data points deviate significantly from the line of best fit, particularly for larger values, indicating that the model's predictions are less precise for higher values within the range.

In the fitted vs. residual plot, the pattern shown here with residuals fanning out (increasing variance with the increase in fitted values) indicates heteroscedasticity, which means the model's predictive performance varies across the range of predicted values, being less reliable at higher values.

The MSE value of the final linear model is around 13.3441 with RMSE being around 3.6529 and around 20.94% of the variance can be explained by this model.

Model Comparison and Selection

After trying out Linear Regression Model, Polynomial Regression Model, Lasso Model, Ridge Model, Lasso Model and Linear Model with Interaction Terms, we were able to reach a conclusion regarding finalising a reasonable model.

After in depth analysis, the model that came out as the most suitable one for our prediction was Polynomial Regression:-

1. **Lower Mean Squared Error (MSE)**: The Polynomial Regression Model showed the lowest MSE values among all the models, both in direct evaluation (13.0694) and cross-validation (13.0476 ± ±0.0991).

2. **Higher R-squared Value**: The Polynomial Regression Model also showed the highest R-squared values (0.2346 in direct evaluation and 0.2346 in cross-validation).

3. **Model Complexity and Fit**: The Polynomial Regression, if we consider interaction terms between features, can identify more complex patterns in the data than the baseline model. This complexity allowed the Polynomial Regression Model to outperform the Baseline Model in terms of both MSE and R-squared.

4. **Risk of Overfitting**: While Polynomial Regression's increased complexity can increase the risk of overfitting, in the present case, the model appears to have successfully captured relevant patterns in the data without simply memorizing the training set, as shown by its better performance metrics (lower MSE and higher R-squared). The cross-validation results further support this, indicating that the model generalizes well to unseen data.

5. **Comparison with Regularized Models**: Despite the regularization in Lasso and Ridge Regression intended to prevent overfitting by penalizing large coefficients, the baseline model did not outperform Polynomial Regression in this context. This implies that the data structure justifies the additional complexity of Polynomial Regression and that it is required to capture the underlying patterns.

The decision to choose the Polynomial Regression Model is supported by its better performance in terms of MSE and R-squared, indicating a better fit to the data while considering the complexity and potential overfitting. The cross-validation results reinforce this decision, showing that the model performs consistently well on different subsets of the data, suggesting good generalization.

Polynomial Regression Model

In [28]:

```
# Defining the degree of the polynomial
degree = 2 # Starting off very simple

# Making pipeline with PolynomialFeatures and LinearRegression
pipeline_poly = make_pipeline(
    PolynomialFeatures(degree),
    LinearRegression()
)

# Fitting the model to the training data
pipeline_poly.fit(X_train, y_train)

# Generating predictions on the testing set
y_pred_poly = pipeline_poly.predict(X_test)

# Model evaluation
mse_poly = mean_squared_error(y_test, y_pred_poly)
r2_poly = r2_score(y_test, y_pred_poly)

# Displaying results
print("Polynomial Regression:")
print("Mean Squared Error:%0.3f"% mse_poly)
print("R-squared:%0.3f"% r2_poly)
```

```
Polynomial Regression:
Mean Squared Error:13.069
R-squared:0.235
```

We have chosen 2 for the order of the polynomial above. However, this is just a base set to understand the polynomial regression from a very simple value like 2. For this simple order 2 polynomial model, the MSE value showed that the squared difference between the actual and predicted values were arounf 13.069 and about 23.46% of the variance can be explained by the model.

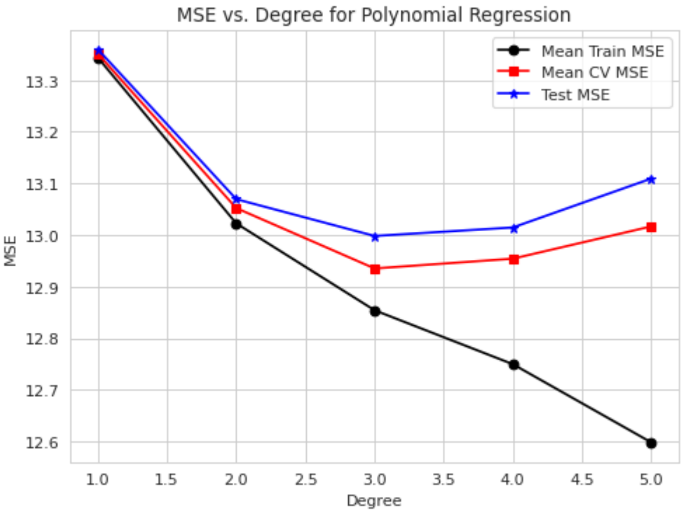
Tuning with GridSearchCV

To make sure that the best model that does not bring higher variance and/or any sort of overfitting is chosen, the following analysis was done where the models of various orders from ranges 1 to 6 are checked over the parameters that we have set using cross validation via GridSearchCV.

In [29]:

```
# Assuming X_train, X_test, y_train, y_test are already defined
from sklearn.model_selection import KFold

# Defining a cross-validation strategy with a fixed random state for reproducibility
cv_strategy = KFold(n_splits=5, shuffle=True, random_state=42)

# Defining the range of degrees for the polynomial
degrees = np.arange(1, 6)

# Initializing the pipeline
pipeline_poly = make_pipeline(PolynomialFeatures(include_bias=False), LinearRegression())

# Defining the parameter grid
parameter_grid = {'polynomialfeatures__degree': degrees}

# Creating and fitting the grid search object
```

```
grid_search = GridSearchCV(
    pipeline_poly,
    parameter_grid,
    cv=cv_strategy,
    scoring='neg_mean_squared_error',
    return_train_score=True
)
grid_search.fit(X_train, y_train)

# Extracting the mean training and cross-validation scores from the grid search
train_mse_values = -grid_search.cv_results_['mean_train_score']
cv_mse_values = -grid_search.cv_results_['mean_test_score']

# Calculating the test MSE for each degree
test_mse_values = []
for degree in degrees:
    # Setting the degree of the polynomial features
    pipeline_poly.set_params(polynomialfeatures__degree=degree)

    # Fitting the model
    pipeline_poly.fit(X_train, y_train)

    # Calculating the MSE on the test set
    y_pred = pipeline_poly.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    test_mse_values.append(mse)

# Plotting
fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(degrees, train_mse_values, 'o-k', label='Mean Train MSE')
ax.plot(degrees, cv_mse_values, 's-r', label='Mean CV MSE')
ax.plot(degrees, test_mse_values, '*-b', label='Test MSE')

# Seting the labels and title
ax.set_xlabel('Degree')
ax.set_ylabel('MSE')
ax.set_title('MSE vs. Degree for Polynomial Regression')

# Legend and grid
ax.legend()
plt.grid(True)
plt.show()

# Printing the best degree from GridSearchCV
best_degree = grid_search.best_params_['polynomialfeatures__degree']
print(f"Best Degree from GridSearchCV: {best_degree}")

# Printing the test MSE for the best degree
best_test_mse = test_mse_values[best_degree - 1]
print(f"Test MSE for the best degree: {best_test_mse:0.3f}")
```



```
Best Degree from GridSearchCV: 3
Test MSE for the best degree: 12.998
```

Making use of a 5 fold cross validation (picked only for the purpose of computational efficiency) and negative mean squared error, the polynomial regression models were generated and the best model order was chosen to be 3 with test MSE value of 12.998

Though 5 has the higest value as seen in the plot between Training and Test MSE vs. Polynomial Degree, it could have been rejected due to the reasons that the variance in that order might have been high with great chances of overfitting. Thats why the model order 3 was chosen to be ideal.

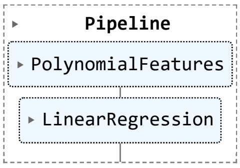
Having the mean train and mean test values close to each other can suggest that the model is not suffering from any overfitting. Hence, having mean train and mean test values very close to each other is a good indication of model performance.

Running Polynomial Regression Model with Degree 3

In [30]:
```
# Updating the pipeline with the best polynomial degree
pipeline_poly = make_pipeline(
    PolynomialFeatures(degree=3, include_bias=False), # Updated degree to 3
    LinearRegression()
)
```

In [31]:
```
# Training the best model
pipeline_poly.fit(X_train, y_train)
```

Out[31]:

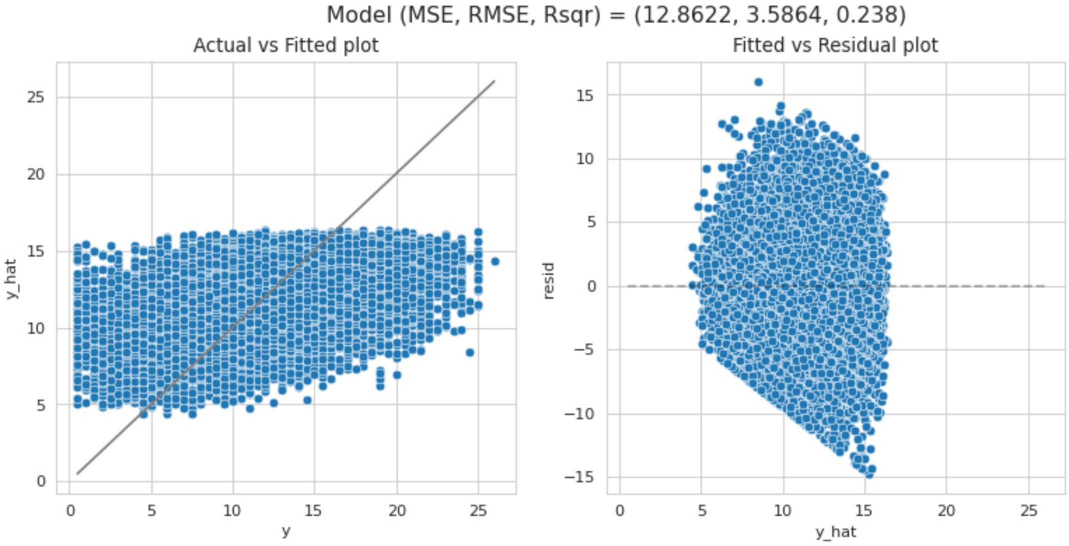


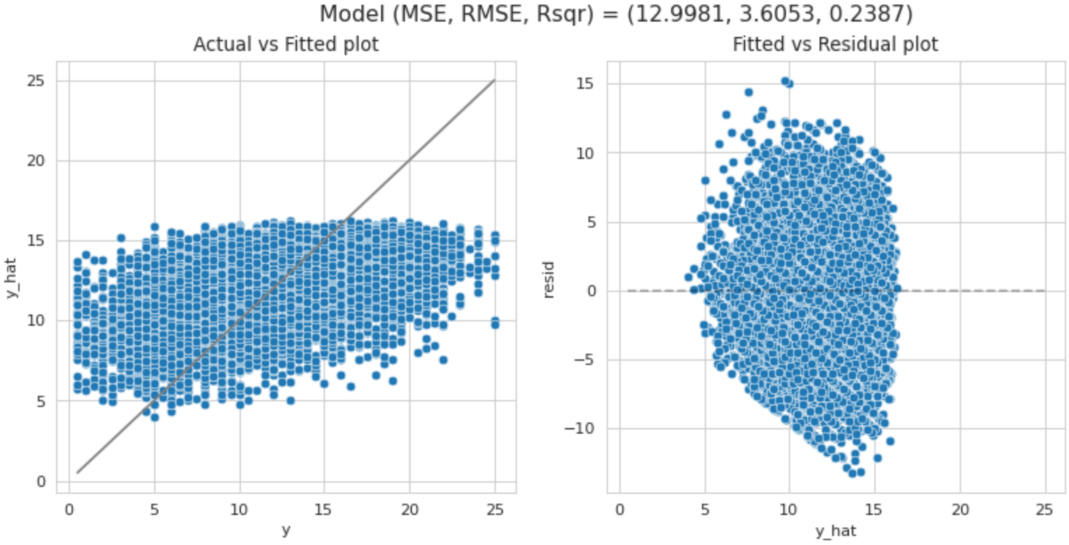
▸ Pipeline

▸ PolynomialFeatures

▸ LinearRegression

In [32]:
```
# Model fitting for Actual vs Fitted and Fitted vs Residual as per custom function model_fit
model_fit(pipeline_poly, X_train, y_train, plot=True)
model_fit(pipeline_poly, X_test, y_test, plot = True)
```





Out[32]:
```
(12.998086287025766, 3.605285881455972, 0.23873335358360603)
```

The above plots show the results of running a regression model on both the training and test sets of data.

There is a positive correlation between the actual and predicted values in the Actual vs. Fitted plots for both the training and test data, but there is a lot of scatter around the line of perfect fit. It appears that the model fits the training data somewhat better in the training plot than in the test plot, as the points in the former plot are somewhat closer to the perfect fit line.

Heteroscedasticity is evident in the training and test data in the Fitted vs. Residual plots; that is, the residuals' variability grows in relation to the size of the predictions. An indication that the model's predictions are less consistent on unknown data is the test data residuals showing a somewhat more prominent dispersion as the expected values grow.

In [33]:

```python
# Printing intercept and coefficients of the best model rounded to 3 decimal points

# Accessing the trained LinearRegression model
poly_regression_model = pipeline_poly.named_steps['linearregression']

# Intercept of the polynomial regression model
poly_intercept = poly_regression_model.intercept_
print("Polynomial Regression Intercept: %0.3f"% poly_intercept)

# Coefficients of the polynomial regression model
poly_coefficients = poly_regression_model.coef_
print("Polynomial Regression Coefficients:")

# Printing coefficients for each feature
for i, feature in enumerate(features_with_change):
    print(f"{feature}: {poly_coefficients[i]:0.3f}")
```

```
Polynomial Regression Intercept: 10.197
Polynomial Regression Coefficients:
eurod: -0.776
br010_mod: 1.988
br015_: -0.001
bmi2: 2.012
sp008_: 0.310
smoking: 0.467
eduyears_mod: -0.650
```

In [34]:

```python
# Assuming best_model
poly_features = pipeline_poly.named_steps['polynomialfeatures']
linear_regression = pipeline_poly.named_steps['linearregression']

# Getting the names of the polynomial features
feature_names = poly_features.get_feature_names_out(input_features=features_with_change)

# Extracting coefficients for features
coefficients_with_names = list(zip(feature_names, linear_regression.coef_))

# Creating a DataFrame
df = pd.DataFrame(coefficients_with_names, columns=['Feature', 'Coefficient'])

# Extracting base feature names for grouping
df['BaseFeature'] = df['Feature'].apply(lambda x: x.split()[0].split('^')[0])

# Grouping by BaseFeature and listing the coefficients in a tabular format
grouped = df.groupby('BaseFeature').apply(lambda x: x[['Feature', 'Coefficient']].to_dict('records'))

# Printing the grouped coefficients in a readable format
for base_feature, group in grouped.items():
    print(f"Base Feature: {base_feature}")
    for item in group:
        print(f"  {item['Feature']}: {item['Coefficient']:.3f}")
    print("\n")
```

```
Base Feature: bmi2
  bmi2: 2.012
  bmi2^2: -1.133
  bmi2 sp008_: -0.222
  bmi2 smoking: -0.070
  bmi2 eduyears_mod: 0.125
  bmi2^3: 0.171
  bmi2^2 sp008_: 0.184
  bmi2^2 smoking: 0.053
  bmi2^2 eduyears_mod: -0.007
  bmi2 sp008_^2: -0.222
  bmi2 sp008_ smoking: 0.110
  bmi2 sp008_ eduyears_mod: -0.040
  bmi2 smoking^2: -0.070
  bmi2 smoking eduyears_mod: -0.008
  bmi2 eduyears_mod^2: -0.003


Base Feature: br010_mod
  br010_mod: 1.988
  br010_mod^2: -0.137
  br010_mod br015_: -0.025
  br010_mod bmi2: -0.163
  br010_mod sp008_: -0.119
  br010_mod smoking: -0.199
  br010_mod eduyears_mod: -0.098
  br010_mod^3: -0.009
  br010_mod^2 br015_: 0.002
  br010_mod^2 bmi2: 0.015
  br010_mod^2 sp008_: 0.020
  br010_mod^2 smoking: 0.037
  br010_mod^2 eduyears_mod: 0.008
  br010_mod br015_^2: -0.003
  br010_mod br015_ bmi2: 0.010
  br010_mod br015_ sp008_: -0.015
  br010_mod br015_ smoking: 0.027
  br010_mod br015_ eduyears_mod: -0.001
  br010_mod bmi2^2: -0.013
  br010_mod bmi2 sp008_: 0.031
  br010_mod bmi2 smoking: -0.002
  br010_mod bmi2 eduyears_mod: 0.004
  br010_mod sp008_^2: -0.119
  br010_mod sp008_ smoking: 0.037
  br010_mod sp008_ eduyears_mod: 0.000
  br010_mod smoking^2: -0.199
  br010_mod smoking eduyears_mod: -0.001
  br010_mod eduyears_mod^2: 0.001


Base Feature: br015_
  br015_: -0.001
  br015_^2: -0.087
  br015_ bmi2: -0.029
  br015_ sp008_: -0.040
  br015_ smoking: 0.050
  br015_ eduyears_mod: 0.029
  br015_^3: -0.038
  br015_^2 bmi2: 0.067
  br015_^2 sp008_: 0.062
  br015_^2 smoking: -0.011
  br015_^2 eduyears_mod: 0.006
  br015_ bmi2^2: -0.046
  br015_ bmi2 sp008_: -0.051
  br015_ bmi2 smoking: -0.007
  br015_ bmi2 eduyears_mod: 0.000
  br015_ sp008_^2: -0.040
  br015_ sp008_ smoking: -0.172
  br015_ sp008_ eduyears_mod: 0.009
  br015_ smoking^2: 0.050
  br015_ smoking eduyears_mod: 0.002
  br015_ eduyears_mod^2: -0.001


Base Feature: eduyears_mod
  eduyears_mod: -0.650
  eduyears_mod^2: 0.075
  eduyears_mod^3: -0.002


Base Feature: eurod
  eurod: -0.776
  eurod^2: 0.023
  eurod br010_mod: -0.000
  eurod br015_: 0.041
  eurod bmi2: 0.008
  eurod sp008_: 0.107
  eurod smoking: -0.099
  eurod eduyears_mod: 0.066
  eurod^3: -0.001
  eurod^2 br010_mod: -0.000
  eurod^2 br015_: 0.001
  eurod^2 bmi2: 0.002
  eurod^2 sp008_: -0.004
  eurod^2 smoking: 0.009
  eurod^2 eduyears_mod: -0.002
  eurod br010_mod^2: -0.003
  eurod br010_mod br015_: 0.002
  eurod br010_mod bmi2: 0.005
  eurod br010_mod sp008_: -0.001
  eurod br010_mod smoking: -0.005
  eurod br010_mod eduyears_mod: 0.001
  eurod br015_^2: -0.010
  eurod br015_ bmi2: 0.000
  eurod br015_ sp008_: 0.037
  eurod br015_ smoking: 0.018
  eurod br015_ eduyears_mod: -0.004
  eurod bmi2^2: 0.003
  eurod bmi2 sp008_: -0.080
  eurod bmi2 smoking: 0.033
  eurod bmi2 eduyears_mod: -0.002
  eurod sp008_^2: 0.107
  eurod sp008_ smoking: 0.005
  eurod sp008_ eduyears_mod: 0.003
  eurod smoking^2: -0.099
  eurod smoking eduyears_mod: 0.003
  eurod eduyears_mod^2: -0.001


Base Feature: smoking
  smoking: 0.467
  smoking^2: 0.467
  smoking eduyears_mod: -0.055
  smoking^3: 0.467
  smoking^2 eduyears_mod: -0.055
  smoking eduyears_mod^2: 0.005


Base Feature: sp008_
  sp008_: 0.310
  sp008_^2: 0.310
  sp008_ smoking: 0.165
  sp008_ eduyears_mod: 0.017
  sp008_^3: 0.310
  sp008_^2 smoking: 0.165
  sp008_^2 eduyears_mod: 0.017
  sp008_ smoking^2: 0.165
  sp008_ smoking eduyears_mod: -0.048
  sp008_ eduyears_mod^2: 0.001
```

- A lower mean squared error (MSE) implies that the model is doing better than the data-driven alternatives.
- An indication of improved performance is a higher R-squared value, which means that the model explains a bigger proportion of the variation in the dependent variable.
- With an MSE of 13.069 and a R^2 of 0.235, the performance metrics for Linear Regression, Lasso Regression, and Ridge Regression are similar.
- Polynomial Regression performs somewhat better, with an MSE of 12.9981 that is lower and a R^2 of 0.238 that is higher.

With the lowest MSE and greatest R-squared value, suggesting it fits the data somewhat better than the other models, Polynomial Regression seems to be the best model among those presented, given these factors. It indicates that the dependent variable's relationship with the independent variables could not be linear and that the polynomial components add complexity to the data.

Interpretation using predicted values from the polynomial model

> We are seeing the instances where cogscore in y_pred_poly is less than 5 and comparing it with domain knowledge

In [35]:

```python
# Initializing a list to store each sample's prediction, truth, and features
samples = []

# Looping through the samples in y_pred
for i in range(len(y_pred_poly)):
    # Checking if the prediction is less than 5
    if y_pred[i] < 5:
        # Retrieving the prediction for the i-th sample
        prediction = y_pred_poly[i]

        # Retrieving the true label for the i-th sample
        truth = y_test.iloc[i]
        # Retrieving the i-th row of features from X_test
        features = X_test.iloc[i]

        # Creating a dictionary for the i-th sample
        sample_data = {'Sample': i+1, 'Prediction': prediction, 'Truth': truth}
        sample_data.update(features.to_dict())

        # Adding to the dictionary to the samples list
        samples.append(sample_data)

        # Break when we have collected 10 samples
        if len(samples) == 10:
            break

# Converting the samples list into a pandas DataFrame
samples_df = pd.DataFrame(samples)

# Displaying the DataFrame
print(samples_df.to_string(index=False))
```

```
 Sample  Prediction  Truth  eurod  br010_mod  br015_  bmi2  sp008_  smoking  eduyears_mod
    556    4.565442    2.5    8.0        1.0     4.0   1.0     0.0      0.0           0.0
    623    7.315654    5.0    7.0        1.0     4.0   4.0     1.0      0.0           0.0
   2831    5.197491    9.5    8.0        1.0     4.0   4.0     0.0      0.0           0.0
   3752    5.813454    4.0    7.0        7.0     4.0   2.0     0.0      0.0           0.0
   6479    5.197491    8.0    8.0        1.0     4.0   4.0     0.0      0.0           0.0
   8898    5.197491    4.5    8.0        1.0     4.0   4.0     0.0      0.0           0.0
  11549    7.283423   11.0   10.0        7.0     1.0   2.0     0.0      1.0           0.0
  11706    4.814605    2.5    9.0        1.0     4.0   4.0     0.0      0.0           0.0
  11782    5.584275    6.5    7.0        1.0     4.0   4.0     0.0      0.0           0.0
  15373    5.197491    5.5    8.0        1.0     4.0   4.0     0.0      0.0           0.0
```

Inferences for model predictions with low cogscore:

The model has demonstrated accurate predictions when compared to the actual values.

- High levels of 'eurod' in samples suggest a greater degree of depression in the individual. The literature demonstrates that it has a substantial impact on cogscore. [16] [17] [18] [19]
- The variable 'eduyears_mod' consistently has a value of 0 for all the samples observed, which aligns with the information found in the literature. [8].
- Samples of the dataset where the value of 'br015_' is high i.e people who hardly exercise are seen to have low cogscore. This is also aligning with our domain knowledge. [13] [14] [15]
- People that don't interact with others often have a low cogscore, as shown by the feature 'sp008_' being 0. One possible explanation is the correlation between social isolation and cognitive decline. [12]
- 'smoking' does not seem to have any effect on cogscore atleast from this subset of values. Although it has been identified to have biological reasons in impacting cogscore. [23]
- 'br010_mod' is seen to be consistently low for this susbet of samples. This implies that not consuming alcohol doesn't help in preventing decline of cognitive ability. [24]
- No clear pattern can be seen for 'bmi2'. This could be due to the fact we are viweing a small subset of the samples.

> We are seeing the instances where cogscore in y_pred_poly is more than 17 and comparing it with domain knowledge

In [36]:

```
# Initializing a list to store each sample's prediction, truth, and features
samples = []

# Looping through the samples in y_pred
for i in range(len(y_pred_poly)):
    # Checking if the prediction is greater than 17
    if y_pred[i] > 17:
        # Retrieving the prediction for the i-th sample
        prediction = y_pred_poly[i]

        # Retrieving the true label for the i-th sample
        truth = y_test.iloc[i]

        # Retrieving the i-th row of features from X_test
        features = X_test.iloc[i]

        # Creating a dictionary for the i-th sample
        sample_data = {'Sample': i+1, 'Prediction': prediction, 'Truth': truth}
        sample_data.update(features.to_dict())

        # Adding to the dictionary to the samples list
        samples.append(sample_data)

        # Break when we have collected 10 samples
        if len(samples) == 10:
            break

# Converting the samples list into a pandas DataFrame
samples_df = pd.DataFrame(samples)
```

```
# Displaying the DataFrame
print(samples_df.to_string(index=False))
```

```
 Sample  Prediction  Truth  eurod  br010_mod  br015_  bmi2  sp008_  smoking  eduyears_mod
    125   16.177238   13.5    3.0        7.0     2.0   3.0     0.0      1.0          25.0
    782   16.241903   14.0    0.0        7.0     1.0   3.0     0.0      0.0          25.0
   2781    8.739710   14.0   11.0        7.0     4.0   4.0     1.0      1.0           3.0
   5792   17.245175   12.0    0.0        4.0     3.0   2.0     0.0      1.0          25.0
   6678   11.006747    4.0    6.0        4.0     1.0   1.0     1.0      0.0           1.0
   7139   16.841551   17.0    1.0        5.0     1.0   4.0     1.0      1.0          25.0
   9966   17.281609   10.0    0.0        2.0     4.0   3.0     1.0      1.0          25.0
  13780   12.869761   15.5   11.0        1.0     2.0   4.0     1.0      1.0          11.0
  14107   12.696166   15.5    7.0        1.0     1.0   1.0     0.0      1.0          15.0
  14649   16.885436   15.0    1.0        5.0     1.0   3.0     1.0      0.0          25.0
```

Inferences for model predictions with high cogscore:

The model has demonstrated accurate predictions when compared to the actual values.

- Low levels of 'eurod' in samples suggest a lesser degree of depression in the individual. The literature demonstrates that it has a substantial impact on cogscore. It must be noted that some of the individuals in this subset with higher levels of depression also retain cognitive score [16] [17] [18] [19]
- The variable 'eduyears_mod' consistently has a value close to 25 for all the samples observed, which aligns with the information found in the literature. Some outliers are also present which could be individuals who did not have the means to access education but still perform well in the cognitive tests. [8].
- Samples of the dataset where the value of 'br015_' is low i.e people who exercise regularly are seen to have high cogscore. This is also aligning with our domain knowledge. [13] [14] [15]
- 'smoking' tends to have a value of 1 here, that is people with high cogscore smoke. This is inconsistent with the literature and often not the case with individuals with high cogscore [23]
- There is an intriguing high value for 'bmi2' in these samples as was seen in the linear regression model. These figures may be outliers or there may be some other explanation, even if the research has indicated a negative correlation between them. It could also be that this is not such an impactful feature for individuals in the EU. [21] [22]
- It would appear that 'br010_mod' with high values indicates that drinking has little effect on people's cognitive abilities. The literature shows that moderate drinking has no effect on the cogscore, whereas chronic severe alcoholism has a noticeable effect. [24]
- No clear pattern can be seen for 'sp008'. This could be due to the fact only a small subset of the samples were viewed.

4. Discussion & Conclusions

Results

Extensive research shows that cognitive scores are affected by a myriad of elements that interact in intricate ways. In particular, the summary highlights how useful polynomial regression is for capturing dynamics including non-linear correlations in data.

It is confirmed that a model that can handle non-linearities and interactions among variables is beneficial for cognitive score prediction, as the polynomial regression model stands out with its lowest Mean Squared Error (MSE) and greatest R-squared value. This realization is especially pertinent in the field of psychology and medicine, where the interplay between factors is often complex and non-additive.

Significant points are highlighted for healthcare professionals and policymakers by insights into variable impacts and non-linear interactions. Education, mental health, and lifestyle characteristics like physical exercise, social engagement, and even seemingly paradoxical elements like the "smoking" variable in the dataset might be the target of interventions to promote cognitive health, according to their findings.

Despite polynomial regression's ability to provide a detailed picture of the data, there is good reason to be aware of the possibility of overfitting. For models to be both dependable and generalizable, it is essential to find the sweet spot between model complexity and predictive performance. The development of machine learning algorithms with enhanced capabilities to manage high-dimensional data and intricate interactions is an area that might be explored in the future.

Finally, the model work appears to be a major step towards a better understanding of cognitive health and dementia risk variables. In particular, when using a degree of 3, the polynomial regression model offers a well-rounded strategy that improves comprehension and prediction accuracy. Furthermore, it suggests that mental health and lifestyle variables may be promising targets for intervention efforts aimed at lowering dementia risk.

Interventions

The results of the model show that these features are suitable for interventions that reduce the risk of dementia. {2020 142,143} They are as follows:

Depressive disorders

- Summary of Intervention: Addressing depression is crucial for reducing the risk of dementia. Treatments can include cognitive behavioral therapy (CBT), other forms of talk therapy, antidepressants, and enhancing social support. The relationship between depression and dementia is complex, involving shared pathways like inflammation and effects on brain health and cognitive reserve.

- Efficacy: Treating depression may not prevent dementia but can improve quality of life, lessen cognitive symptoms of depression, and potentially influence dementia progression risk factors.

Alcohol Use

- Summary of Intervention: Moderation of alcohol consumption is recommended, focusing on reducing excessive drinking. Interventions can include counseling, support groups, and education on the dangers of heavy drinking. The link between alcohol misuse and increased dementia risk may involve impacts on cardiovascular health, direct neurotoxicity, and lifestyle factors.
- Efficacy: Reducing heavy drinking could lower the risk of developing dementias associated with alcohol, such as alcoholic dementia and possibly vascular dementia, though evidence on light to moderate drinking is more mixed.

Physically Active Lifestyle

- Recommendation: Engage in regular physical activities, including aerobics, strength training, and balance exercises. Benefits might come from improved cardiovascular health, increased cerebral blood flow, enhanced neurogenesis, and raised levels of brain-derived neurotrophic factors (BDNF).
- Efficacy: Exercise interventions have shown mixed results in randomized controlled trials (RCTs), but some studies have indicated that exercise can improve executive function, memory, and overall cognitive performance in older individuals. More research is needed to identify the most effective types and intensities of exercise, but evidence suggests a role for exercise in reducing dementia risk.

Obesity

- Summary of Intervention: Interventions for obesity focus on weight loss through diet changes, increased physical activity, and sometimes medication or surgery. Obesity, particularly in midlife, is linked to an increased risk of dementia due to factors like metabolic syndrome, inflammation, and cardiovascular risk factors.
- Efficacy: While direct evidence linking obesity interventions to dementia risk reduction is still emerging, managing weight can potentially lower dementia risk by addressing these underlying factors.

Social Interaction

- Summary of Intervention: This intervention aims to boost social engagement through community involvement, social activities, and relationship building. Social isolation and loneliness are risk factors for cognitive decline and dementia.
- Efficacy: Although longitudinal studies suggest social interaction might offer protective effects, intervention study results are less clear. However, strong social connections are associated with better mental health and could help build cognitive reserve.

Smoking

- Summary of Intervention: Smoking cessation programs, which may include nicotine replacement therapy, counseling, and support groups, target smoking's well-established role as a risk factor for dementia through its effects on vascular health and neuroinflammation.
- Efficacy: The benefits of quitting smoking in reducing the risk of dementia accumulate over smoke-free years, showing its effectiveness.

Education

- Summary of Intervention: Higher education levels are linked to a lower risk of dementia, possibly due to cognitive reserve. Promoting lifelong learning and cognitive stimulation can be achieved through formal education and continuous learning opportunities.
- Efficacy: Engaging in intellectually stimulating activities and lifelong learning can help build cognitive reserve and reduce dementia risk, even when past education levels cannot be altered.

Interventions targeting these risk factors incorporate lifestyle, behavioral, and medical changes. While the efficacy of these interventions varies, they collectively contribute to better brain health and may reduce or delay dementia onset.

Limitations

- **Longitudinal studies and causality.**

The PAF model presupposes a causal relationship between a risk factor and dementia; treatments cannot reduce the incidence of dementia unless there is a causative link. The strongest proof of causation would come from human randomized controlled trials (RCTs). We know that higher levels of education are related with a decrease in the age-specific incidence of dementia, but conducting comparable studies for many other potential risk variables is not feasible. 25 Causality criteria have been suggested in lieu of this human experimental data.135 Potentially modifiable risk factors According to the existing research, removing risk factors would lead to a proportionate decrease in incident dementia cases, and PAF reflects this. It is not possible to eradicate all of these risk factors, and some of them may even be a component of dementia syndrome, therefore this number should be taken with a grain of salt.

- **Reverse causality**

There are cases when the direction of causation is ambiguous or even bidirectional. For instance, our estimates may be exaggerated because cognitive impairment might induce and exacerbate symptoms such as decreased socialization and increased depressive symptoms. It is not always easy to tell which way a causal relationship runs when thinking about risk variables that manifest just before impairment begins to manifest; for example, it is not always clear whether depression raises the chance of dementia or dementia raises the risk of depression.

- **Global estimates of prevalence**

Although we have utilized the biggest populations we could locate to determine the frequency of dangers, it is important to note that these populations are not necessarily worldwide and will vary among regions, cultures, and economic levels.

- **Population Attributable Factor**

Whether we think the real PAF is lower or greater than our estimate, the underlying premise remains that modifiable risk factors account for a significant proportion of dementia cases. There would be far-reaching consequences for healthcare and societal expenditures if modifying risk factors reduced the worldwide prevalence of dementia.

One major drawback is that the majority of the information comes from high-income nations, whereas there is a lack of data from low- and middle-income countries (LMIC). Unfortunately, we do not have data to model hearing loss, despite the fact that it is a significant modifiable risk factor found in midlife patients.

Management of metabolic, psychological, auditory, and cerebrovascular risk factors may postpone the beginning of many instances of dementia by a few years, while public health measures will not halt, prevent, or cure every potentially curable dementia. A five-year delay in the development of dementia would cut the frequency in half. Reducing the prevalence of the seven main health and lifestyle factors by 10% would reduce the number of dementia cases by over a million, according to estimates. Alternatively, an intervention that delays dementia by one year could reduce the number of people living with dementia by 9 million by 2050. Although it may not be expected to have such a large impact from modifying risk factors in practice, any decrease in the likelihood of dementia would be considered an impressive feat.

5. References

1. What is dementia?, April 2019. Available online at CDC.
2. Marco Carone, Masoud Asgharian, and Nicholas P. Jewell, "Estimating the lifetime risk of dementia in the Canadian elderly population using cross-sectional cohort survival data." *Journal of the American Statistical Association*, 109:24–35, January 2014.
3. Tim Stevens, Gill Livingston, Ginnette Kitchen, Monica Manela, Zuzana Walker, and Cornelius Katona, "Islington study of dementia subtypes in the community." *British Journal of Psychiatry*, 180:270–276, March 2002.
4. Gill Livingston, Andrew Sommerlad, Vasiliki Orgeta, Sergi G Costafreda, Jonathan Huntley, David Ames, Clive Ballard, Sube Banerjee, Alistair Burns, Jiska Cohen-Mansfield, Claudia Cooper, Nick Fox, Laura N Gitlin, Robert Howard, Helen C Kales, Eric B Larson, Karen Ritchie, Kenneth Rockwood, Elizabeth L Sampson, Quincy Samus, Lon S Schneider, Geir Selbæk, Linda Teri, and Naaheed Mukadam, "Dementia prevention, intervention, and care." *The Lancet*, 390:2673–2734, December 2017.
5. C. Patterson, "World Alzheimer Report 2018." Alzheimer's Disease International, London, 2018.
6. Martin Prince, Anders Wimo, Maëlenn Guerchet, Gemma-Claire Ali, Yu-Tzu Wu, et al., "World Alzheimer Report 2015: The global impact of dementia: An analysis of prevalence, incidence, cost and trends." Alzheimer's Disease International, 2015. Research Report.
7. Bengt Winblad, Philippe Amouyel, Sandrine Andrieu, Clive Ballard, Carol Brayne, Henry Brodaty, Angel Cedazo-Minguez, Bruno Dubois, David Edvardsson, Howard Feldman, Laura Fratiglioni, Giovanni B Frisoni, Serge Gauthier, Jean Georges, Caroline Graff, Khalid Iqbal, Frank Jessen, Gunilla Johansson, Linus Jönsson, Miia Kivipelto, Martin Knapp, Francesca Mangialasche, René Melis, Agneta Nordberg, Marcel Olde Rikkert, Chengxuan Qiu, Thomas P Sakmar, Philip Scheltens, Lon S Schneider, Reisa Sperling, Lars O Tjernberg, Gunhild Waldemar, Anders Wimo, and Henrik Zetterberg, "Defeating Alzheimer's disease and other dementias: a priority for European science and society." *The Lancet Neurology*, 15:455–532, April 2016.
8. Michael J. Valenzuela and Perminder Sachdev, "Brain reserve and dementia: a systematic review." *Psychological Medicine*, 36:441–454, April 2006.
9. Latha Velayudhan, Seung-Ho Ryu, Malgorzata Raczek, Michael Philpot, James Lindesay, Matthew Critchfield, and Gill Livingston, "Review of brief cognitive tests for patients with suspected dementia." *International Psychogeriatrics*, 26:1247–1262, August 2014.
10. Gill Livingston, Jonathan Huntley, Andrew Sommerlad, David Ames, Clive Ballard, Sube Banerjee, Carol Brayne, Alistair Burns, Jiska Cohen-Mansfield, Claudia Cooper, Sergi G Costafreda, Amit Dias, Nick Fox, Laura N Gitlin, Robert Howard, Helen C Kales, Mika Kivimäki, Eric B Larson, Adesola Ogunniyi, Vasiliki Orgeta, Karen Ritchie, Kenneth Rockwood, Elizabeth L Sampson, Quincy Samus, Lon S Schneider, Geir Selbæk, Linda Teri, and Naaheed Mukadam, "Dementia prevention, intervention, and care: 2020 report of the Lancet commission." *The Lancet*, 396:413–446, August 2020.
11. Prince, Martin, Bryce, Renata, Albanese, Emiliano, Wimo, Anders, Ribeiro, Wagner, & Ferri, Cleusa P. (2013). "The global prevalence of dementia: A systematic review and metaanalysis." Alzheimer's & Dementia, 9(1):63.
12. Kuiper, Jisca S., Zuidersma, Marij, Oude Voshaar, Richard C., Zuidema, Sytse U., van den Heuvel, Edwin R., Stolk, Ronald P., & Smidt, Nynke. (2015, July). "Social relationships and risk of dementia: A systematic review and meta-analysis of longitudinal cohort studies." *Ageing Research Reviews*, 22:39–57.

13. Young, Jeremy, Angevaren, Maaike, Rusted, Jennifer, & Tabet, Naji. (2015, April). "Aerobic exercise to improve cognitive function in older people without known cognitive impairment." *Cochrane Database of Systematic Reviews*, 2015(4).

14. Leckie, Regina L., Oberlin, Lauren E., Voss, Michelle W., Prakash, Ruchika S., Szabo-Reed, Amanda, Chaddock-Heyman, Laura, Phillips, Siobhan M., Gothe, Neha P., Mailey, Emily, Vieira-Potter, Victoria J., Martin, Stephen A., Pence, Brandt D., Lin, Mingkuan, Parasuraman, Raja, Greenwood, Pamela M., Fryxell, Karl J., Woods, Jeffrey A., McAuley, Edward, Kramer, Arthur F., & Erickson, Kirk I. (2014, December). "Bdnf mediates improvements in executive function following a 1-year exercise intervention." *Frontiers in Human Neuroscience*, 8.

15. Vaughan, Sue, Wallis, Marianne, Polit, Denise, Steele, Mike, Shum, David, & Morris, Norman. (2014, September). "The effects of multimodal exercise on cognitive and physical functioning and brain-derived neurotrophic factor in older women: a randomised controlled trial." *Age and Ageing*, 43:623–629.

16. George S. Alexopoulos. Vascular disease, depression, and dementia. Journal of the American Geriatrics Society, 51:1178–1180, 8 2003

17. Morkem, Rachael, Barber, David, Williamson, Tyler, & Patten, Scott B. (2015, December). "A Canadian primary care sentinel surveillance network study evaluating antidepressant prescribing in Canada from 2006 to 2012." *The Canadian Journal of Psychiatry*, 60:564–570.

18. Olfson, Mark & Marcus, Steven C. (2009, August). "National patterns in antidepressant medication treatment." *Archives of General Psychiatry*, 66:848.

19. Sheline, Yvette I., West, Tim, Yarasheski, Kevin, Swarm, Robert, Jasielec, Mateusz S., Fisher, Jonathan R., Ficker, Whitney D., Yan, Ping, Xiong, Chengjie, Frederiksen, Christine, Grzelak, Monica V., Chott, Robert, Bateman, Randall J., Morris, John C., Mintun, Mark A., Lee, Jin-Moo, & Cirrito, John R. (2014, May). "An antidepressant decreases CSF Aβ production in healthy individuals and in transgenic AD mice." *Science Translational Medicine*, 6.

20. Valenzuela, Michael J. (2008, May). "Brain reserve and the prevention of dementia." *Current Opinion in Psychiatry*, 21:296–302.

21. Yaffe, Kristine. (2007, April). "Metabolic syndrome and cognitive disorders." *Alzheimer Disease Associated Disorders*, 21:167–171.

22. Luchsinger, José A. & Gustafson, Deborah R. (2009, January). "Adiposity and Alzheimer's disease." *Current Opinion in Clinical Nutrition Metabolic Care*, 12:15–21.

23. Swan, Gary E. & Lessov-Schlaggar, Christina N. (2007, September). "The effects of tobacco smoke and nicotine on cognition and the brain." *Neuropsychology Review*, 17:259–273.

24. Rehm, Jürgen, Hasan, Omer S. M., Black, Sandra E., Shield, Kevin D., & Schwarzinger, Michaël. (2019, December). "Alcohol use and dementia: a systematic scoping review." *Alzheimer's Research & Therapy*, 11:1.