Project 1: Predicting Dementia Severity

Student 1, Student 2

March 11, 2024

1 Machine Learning in Python - Project 1

Due Friday, March 8th by 4 pm.

Oliver Webb, Will Henshon, Ezra Muratoglu, James Zoryk

1.1 Setup

Install any packages here and load data

[1]:
```
# 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
sns.set(color_codes=True)

# Plotting defaults
plt.rcParams['figure.figsize'] = (8,5)
plt.rcParams['figure.dpi'] = 80

# sklearn modules
import sklearn
from scipy import stats
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import GridSearchCV, KFold

from sklearn.model_selection import train_test_split

from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.linear_model import LinearRegression
from sklearn.impute import SimpleImputer
```

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import PolynomialFeatures
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import Lasso
from sklearn.linear_model import LassoCV

from itertools import combinations
from sklearn.linear_model import LinearRegression
from itertools import combinations_with_replacement



import warnings
# Suppress specific seaborn warning
warnings.filterwarnings("ignore", message="Vertical orientation ignored with␣
 ↪only `x` specified.", category=UserWarning)
```

[12]:
```
def model_plot(m, X, y, test=True):
    """Creates residual plots for model fits.

    Args:
        m: sklearn model object
        X: model matrix to use for prediction
        y: outcome vector to use to calculating rmse and residuals
        test: boolean value for whether the input dataset is a test or train
    """

    y_hat = m.predict(X)


    res = pd.DataFrame(
        data = {'Cogscore': y, 'Estimated cogscore': y_hat, 'Residual': y -␣
 ↪y_hat}
    )

    plt.figure(figsize=(12, 6))

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

    plt.subplot(122)
    sns.scatterplot(x='Estimated cogscore', y='Residual', 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)
    if(test==True):
        plt.suptitle("Model test plots for the test dataset", fontsize=14)
    else:
        plt.suptitle("Model test plots for the train dataset", fontsize=14)
    plt.show()
```

[13]:
```
def model_fit(m, X, y):
    """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
    """

    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))

    return Metrics
```

[14]:
```
def get_coefs(m):
    """Returns the model coefficients from a Scikit-learn model object as an␣
 ↪array.
    """

    # If pipeline, use the last step as the model
    if (isinstance(m, sklearn.pipeline.Pipeline)):
        m = m.steps[-1][1]
    if m.intercept_ is None:
        return m.coef_
    return np.concatenate([m.coef_])
```

2 Introduction

Dementia is a syndrome associated with a continuous decline in one's general cognitive ability severe enough to interfere with daily activities [4]. It most commonly presents itself in the form of

Alzheimer's disease, vascular dementia, and Lewy body dementia [8, 9]. While the precise causes of dementia are not fully understood, there are several possible risk factors associated with its onset, with age being one of the largest [5, 6]. Other potential risk factors include lack of exercise, depression, low education levels, midlife hearing loss, obesity, hypertension, smoking, physical inactivity, diabetes, and social isolation [7].

This report aims to improve the understanding of factors associated with dementia risk by developing a predictive model for cognitive function using observational survey data from the Survey of Health, Ageing and Retirement in Europe (SHARE) [1, 2, 3]. The key objectives are to highlight the factors that play the largest role in predicting dementia risk, and to develop an accurate model for assessing an individual's risk based on their health and lifestyle information. While some factors, such as chronic diseases, age, or gender, are immutable, we particularly seek to identify modifiable risk factors that may be amenable to lifestyle interventions and public health measures.

The data used in this study comes from the easySHARE dataset [3], a simplified version of the SHARE survey conducted in multiple European countries across 8 waves between 2004 and 2020 (with wave 3 omitted). The dataset contains key features related to participants' health and lifestyle, including country of origin and birth, years of education, body mass index (BMI), smoking and drinking behavior, physical activity levels, quality of life, and the presence of chronic diseases or depression. The primary measure of dementia risk is a variable called 'cogscore,' a number between 0 and 26 representing the cognitive function of participants, with lower values corresponding to lower cognitive ability. The cogscore is determined by combining results from two numeracy tests, two word recall tests, and an orientation test. It is worthwhile to note that while we assume a strong relationship between low cogscore and dementia severity or risk, this metric is not a direct measure for dementia in the medical sense.

In developing the model, we explored the key features and their relationships with the cogscore variable both quantitatively and graphically, removing less relevant features as necessary. To highlight the key factors associated with dementia risk while maintaining interpretability, we used a regression model to predict the cogscore (our target variable) given the input features after data processing and feature engineering. Several models were considered, such as lasso, ridge, polynomial, and linear regression, allowing further exploration of factors associated with dementia risk.

After this testing, we decided to present a modified polynomial model, where the majority of terms are linear, and a small subset of quadratic terms were added. This model improved substantially on a baseline linear model, while keeping the model interpretable with a suitably low-dimensional feature space. While we were able to achieve marginal accuracy improvements with more complex models, we ultimately discarded them due to their lack of interpretability and the relatively small magnitude of improvements in accuracy.

From this model, we found that in addition to age, education, quality of life, and level of vigorous activity were three of the most significant factors in predicting cognitive score. Therefore, any plan of action should focus on improving these three factors, with the latter two applying to the at-risk population. Education will require a more long-term outlook for prevention.

3 Exploratory Data Analysis and Feature Engineering

3.1 1: Data cleaning and separation

Firstly, duplicate rows and those with NaNs are dropped. We check the datatypes to ensure representations are appropriate. We also drop the following categories: mergeid, int_year, wave, country_mod, birth_country, citizenship, bmi2, ever_smoked, ch001_.

Mergeid, int_year, and wave have been dropped since our dataset is modified to include only 1 wave per participant. Country and citizenship data have been dropped as these features are not modifiable through lifestyle interventions. Additionally, the use of country and citizenship data in the model would likely be in order to compare cognitive score to income data in different countries, but factors related to income are already covered by several other variables, such as quality of life score and whether people make ends meet. bmi2 has been dropped as bmi provides more information, and ever_smoked has been dropped due to low correlation to cogscore in preliminary EDA. Finally, ch001_ (number of children) was dropped as this feature is not modifiable via lifestyle interventions.

In our preliminary exploration of easyshare_all.csv, we identified two additional features that seemed to be correlated with cogscore: these were 'euro5' and 'co007_' which correspond respectively to sleep and the extent to which one's household makes ends meet. These features are included in our model by merging on wave and mergeid.

Finally some of the more confusingly named features are renamed to improve the readability of the dataset.

[15]:
```
# Load data from easyshare_all.csv
d = pd.read_csv("../mlp_project_one/project1_materials/easyshare.csv")
dall = pd.read_csv("../mlp_project_one//project1_materials/easyshare_all.csv")

# drop nans in wave column which would otherwise cause problems for merging
d = d.dropna(subset=['wave'])

# Merge the two DataFrames based on the mergeid and wave columns
# This equates to merging by person, and by the same questionaire / time they␣
 ↪answered the survey
d = pd.merge(d, dall[['mergeid','wave', 'euro5', 'co007_']],␣
 ↪on=['mergeid','wave'], how='left')
```

[16]:
```
# Drop irrelevant columns
d = d.drop(['mergeid', 'int_year', 'wave','country', 'country_mod',␣
 ↪'birth_country',
              'citizenship', 'bmi2', 'ever_smoked','ch001_'], axis=1)
# Drop rows containing NAs
d = d.dropna()
# Drop duplicates:
d = d.drop_duplicates()
# Rename confusing columns
d = d.rename(columns={
    'isced1997_r': 'Education_Level',
```

```
    'eduyears_mod': 'Years_of_Education',
    'eurod': 'Depression_Scale',
    'bmi': 'Body_Mass_Index',
    'smoking': 'Smoking_Habit',
    'br010_mod': 'Alcohol_Consumption_Per_Week',
    'br015_': 'Vigorous_Activities',
    'casp': 'Quality_of_Life_Index',
    'sp008_': 'Help_Given_Outside_Household',
    'cogscore': 'Cognitive_Score',
    'euro5': 'Sleep',
    'co007_':'Ends_Meet'
})

# Cast the binary data as ints:
d['Smoking_Habit'] = d['Smoking_Habit'].astype(int)
d['female'] = d['female'].astype(int)
d['Help_Given_Outside_Household'] = d['Help_Given_Outside_Household'].
 ↪astype(int)
```

[17]:
```
#### Testing
# List columns with NAs
columns_with_nas = d.columns[d.isnull().any()].tolist()
print("Columns with NAs:", columns_with_nas)
```

```
Columns with NAs: []
```

[18]:
```
from sklearn.model_selection import train_test_split

# Separate features from target variable:
X = d.drop('Cognitive_Score', axis = 1) # Set of features
y = d['Cognitive_Score']

# Withhold 10% of data for testing
rng = np.random.seed(0)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1,␣
 ↪random_state = rng)

# For the sake of EDA, recombine the training data in a single df
df = X_train.join(y_train)
df_test = X_test.join(y_test)
```

3.2 2: Detecting outliers and visualizing distributions

An outlier is a point or set of points that differs dramatically from the majority of points in the dataset. Oftentimes, outliers can skew the predictive accuracy of a model, so it is reasonable to remove some of the more extreme datapoints to ensure that our model most accurately depicts the typical demographic range. It is helpful to first visualize the data ranges using box plots to identify

which features might have outliers that are reasonable candidates for removal.

[19]:
```
num_rows = 5  # Adjust the number of rows as needed based on the number of␣
 ↪columns
num_cols = 2

# Create subplots
fig, axes = plt.subplots(num_rows, num_cols, figsize=(18, 14))  # Adjust the␣
 ↪figsize as needed

# Flatten the axes for easier iteration
axes = axes.flatten()

# Iterate over each column and create a box plot
for i, column in enumerate(df.select_dtypes(exclude=['int64']).columns): #i,␣
 ↪column in enumerate(df.columns):
    # Check if the column data type is integer

    sns.boxplot(x=df[column], ax=axes[i], orient="v", width=0.5)  # Create␣
 ↪boxplot for the column

    axes[i].set_title(f"Box plot for {column}")
    axes[i].set_ylabel("Values")
    axes[i].tick_params(axis='x', rotation=45)  # Rotate x-axis labels for␣
 ↪better readability
    i += 1


# Remove any extra empty subplots if the number of columns is not divisible by␣
 ↪the number of rows
if len(df.columns) < num_rows * num_cols:
    for j in range(len(df.columns), num_rows * num_cols):
        fig.delaxes(axes[j])

plt.tight_layout()  # Adjust layout to prevent overlap of plots
plt.show()
```

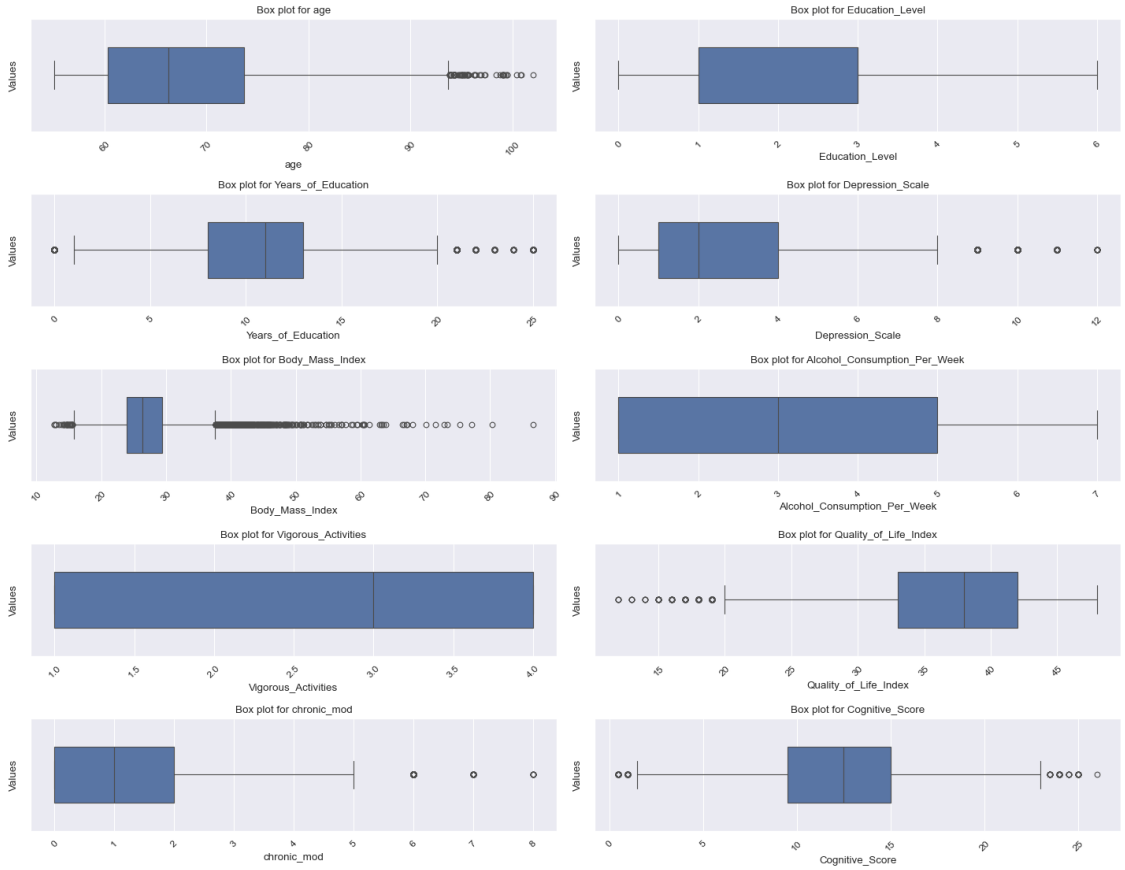

Fig. 1. Box plots for each considered predictor in the dataset. We see a relative abundance of outliers in the age and body mass index plots.

As previously discussed, significant outliers can make the model less applicable to the general population; however, it is inadvisable to remove them without good reason. Keeping the goals of this study in mind - chiefly to identify modifiable features that could be targeted to lower the public health burden of dementia - we decided that removing outliers at the upper end of the 'age' and 'bmi' ranges would be a reasonable choice. As one might expect, age is the feature most strongly correlated with lower cognitive score, and realistically any interventions gleaned from this study can only serve to delay the onset of dementia. As such, we argue that including data corresponding to the oldest study subjects (who are also most likely to have reduced cognitive scores) might introduce a survivor bias that could skew our models representation of relationships between other features and cognitive score. For example, were we to include older 'age' outliers, health marking features related to longevity might appear to contribute to lower cognitive scores.

Likewise, we chose to remove participants with a bmi larger than 60 (bmi over 40 corresponds to class iii obesity - the most severe rating for the illness), because our model should capture feature interactions that are indicative of patterns in the general population. Moreover, it is likely that individuals with very large bmi might bring forth other health concerns and complications before the onset of dementia, which may cause unrelated declines in their cognitive scores.

We also made the decision to drop rows containing NAs rather than impute these values in our

pipeline, as we found that imputing resulted in poorly distributed data and due to concerns that the imputed data might corrupt some of the feature interactions captured within the model. While we considered a sophisticated imputer that would estimate the missing values based on the other inputs from the same row, this approach was abandoned because it would falsely imply certainty in the model that does not necessarily exist.

We remove outliers from the train data, but as is best practice we choose to leave the test data unmodified.

Moreover, as all of our data are meant to be positive numbers, rows with any negative data are removed from both the testing and training sets.

[20]:
```
prev_shape = df.shape
print(prev_shape)
# Calculate the IQR for age
def remove_outliers(dataframe):
    Q1_age = dataframe['age'].quantile(0.25)
    Q3_age = dataframe['age'].quantile(0.75)
    IQR_age = Q3_age - Q1_age

    # Remove outliers in age
    result = dataframe[~((dataframe['age'] > (Q3_age + 1.5 * IQR_age)))]

    # Remove values above 60 in Body_Mass_Index
    result = result[result['Body_Mass_Index'] <= 60]

    print("Shape after removing negative rows and outliers with large age or␣
 ↪values above 60 in Body_Mass_Index:", result.shape)

    return result

def remove_negs(dataframe):
    mask = (dataframe < 0).any(axis=1)

    # Filter the DataFrame to keep only rows without negative values
    result = dataframe[~mask]

    return result

df = remove_outliers(df)
df = remove_negs(df)

df_test = remove_negs(df_test)
```

```
(42933, 15)
Shape after removing negative rows and outliers with large age or values above
60 in Body_Mass_Index: (42837, 15)
```

Now that outliers have been removed, we want to visualize our data distributions and consider data

transformations that may be amenable to model training and performance. To do this, a histogram is created for each of the above predictors.

[21]:
```
# Define the number of rows and columns for subplots
num_rows = 14  # Adjust the number of rows as needed based on the number of␣
 ↪columns
num_cols = 1

# Create subplots
fig, axes = plt.subplots(num_rows, num_cols, figsize=(18, 24))  # Adjust the␣
 ↪figsize as needed

# Flatten the axes for easier iteration
axes = axes.flatten()

# Iterate over each column and create a box plot and histogram
for i, column in enumerate(df.columns[:-1]):

    # Plot histogram
    sns.histplot(data=df, x=column, ax=axes[i])
    axes[i].set_title(f"Histogram for {column}")
    axes[i].set_ylabel("Frequency")
    axes[i].set_xlabel("Values")

# Remove any extra empty subplots if the number of columns is not divisible by␣
 ↪the number of rows
if len(df.columns) < num_rows * 2*num_cols:
    for j in range(len(df.columns)*2, num_rows * 2*num_cols):
        fig.delaxes(axes[j])

plt.tight_layout()  # Adjust layout to prevent overlap of plots
plt.show()
```

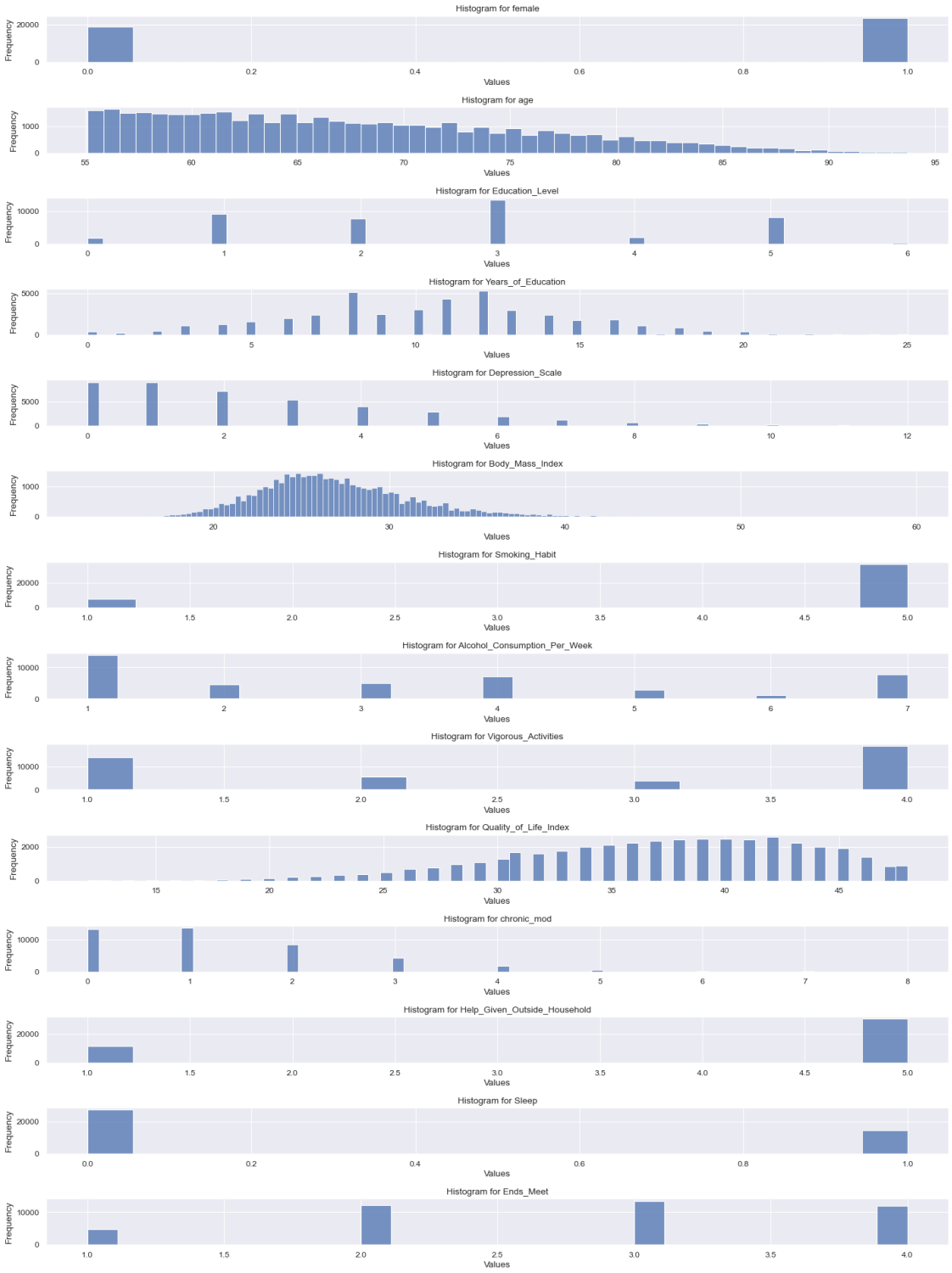

Fig. 2. Histograms for each of the model predictors.

The histograms for our data appear generally well distributed, and many of the finer-resolution

data (such as bmi or quality of life) appear near-gaussian. While transforming the data to give it a more symmetrical or gaussian shape can sometimes improve model fit, this is avoided as non-linear transformations such as box-cox can make weight coefficients less interpretable. Later on in this script, our data is standardized in the pipeline using standard scaler, so that linear relationships between cogscore and other features can be interpreted without having to consider the ranges of the independant variables.

We also considered one-hot encoding our ordinal data, and while doing so in preliminary baseline modeling resulted in a more accurate model, we found that the model became far less interpretable under one-hot encoding as the dimensionality of the feature space ballooned dramatically. For example, the ends_meet and vigorous activity predictors are discretized in such a way that they could be almost considered categorical. But in order to do this, in the model, we would need to split these predictors into 4 separate categories, and keep track of all of them together. While this is feasible for interpretation still, once polynomial terms or more complex weighting systems are used, it becomes impossible to determine whether the predictors remain important.

Regarding the histogram for cognitive score values, generally, in a linear regression, the output should take the shape of a normal distribution, since we assume that the error in each individual estimate is normally distributed. If this is not the case, then we consider transformations of the data, such as a Box-Cox, which would scale the data such that it mets this assumption.

[22]:
```
fig, axes = plt.subplots(1,1, figsize=(6,6)) #Create plot window
axes = sns.histplot(data=df, x='Cognitive_Score', bins=26) #Create histogram
plt.show()
```

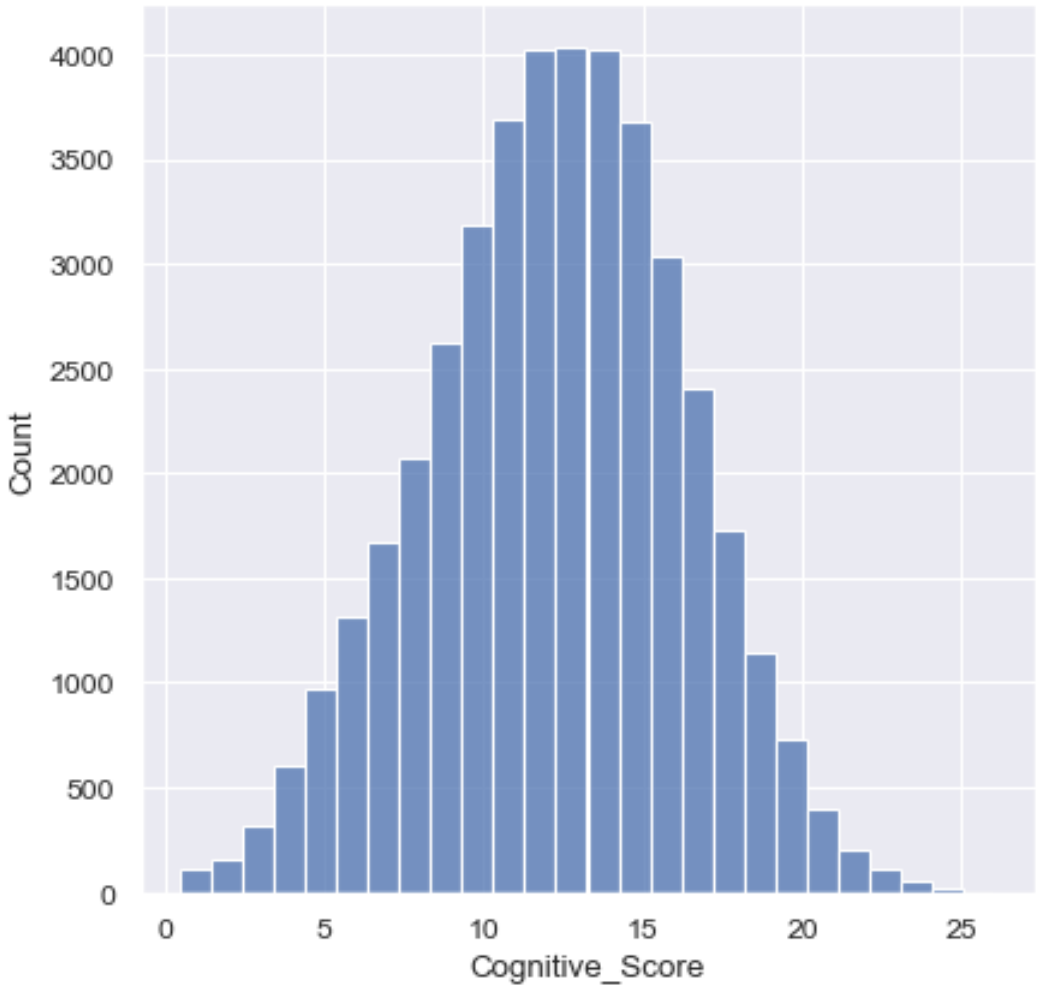

Fig. 3. Histogram of the cognitive score counts. Bins are defined so that each unit of cognitive score receives its own bin (note that half points are possible, so the bins still act as a filter).

Fortunately, the cognitive score is already normally distributed without any sort of scaling. Therefore, no transofmration of the congnitive scores is required in order to build the model.

3.3 3: Correlation heatmap on training data

Finally, a correlation heatmap is generated using the training data to visualize how our selected features correlate to each other.

[23]:
```
# Generate and plot heatmap
plt.figure(figsize=(20,10))
c= df.corr()
sns.heatmap(c,cmap="BrBG",annot=True)
```

[23]: <Axes: >

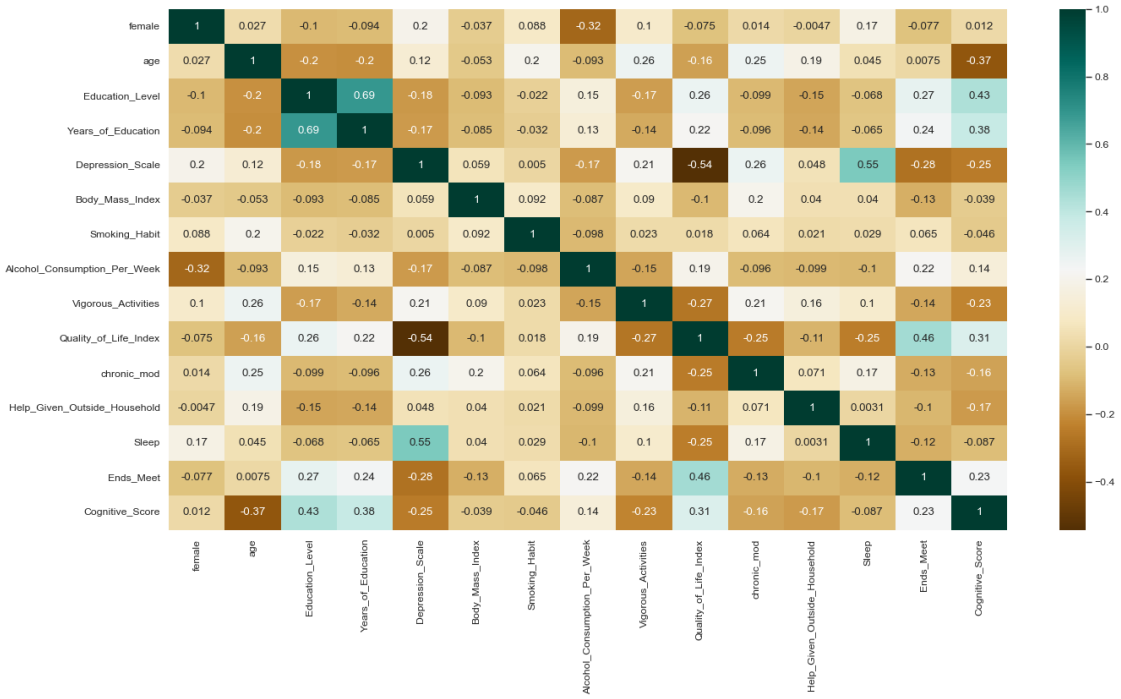

Fig. 4. Correlation heatmap for the selected predictors. Note that cognitive score is the bottom row (and rightmost column).

Heatmaps are a useful tool to visualize multiple correlations. Focusing on the bottom row, showing the correlations between each feature and the cognitive score. In particular, we notice that age shows a moderately negative correlation with cognitive score (r = -0.37), suggesting that as age increases, there may be a tendency for cognitive abilities to decline. However, this correlation must be contextualized within the broader spectrum of individual variability and the potential for confounding variables, such as the presence of chronic conditions or socioeconomic factors (such as 'Ends_Meet').

The correlation between the Depression Scale and cognitive score (r = -0.25) is another area of interest. This relationship may indicate that higher levels of depressive symptoms are associated with lower cognitive performance.

Likewise, vigorous activities exhibit a negative correlation with cognitive score (r = -0.23). Recalling that Vigorous_Activity takes values in {1, 2, 3, 4}, with 1 being the most active and 4 being the least active, this negative correlation seems intuitive.

Curiously, Body_Mass_index has near-zero correlations with almost all other health-related features. Bearing in mind that this observation comes after the removal of outliers, it is still particularly surprising to see such weak correlations. However, when considering that bmi, which is calculated by dividing an individuals weight in kilograms by the square of their height in meters, does not directly measure bone density, body fat percentage, or muscle mass, it seems reasonable that this metric might not actually be a strong indicator of overall health (or cognitive health for that matter) — at least not for individuals within a standard/average bmi range.

'Education_Level' and 'Years_of_Education' show positive correlations with cognitive score (r = 0.43 and r = 0.38, respectively), which could imply that higher educational attainment is associated with better cognitive function. Nonetheless, the interaction between Education Level and 'Ends_Meet' (r = 0.27) highlights potential collinearity issues that could inflate or deflate the apparent effects of each individual variable on cognitive scores when modeling. Other instances of high collinearity that ought to be kept in mind include a strong negative correlation between 'Depression_Scale' and 'Sleep' (r = .55), a strong positive correlation between 'Quality_of_Life_index' and 'Ends_Meet' (r = .46), a strong negative correlation between 'Quality_of_life_index' and 'Depression_Scale' (r = -.54), and of course a strong positive correlation between 'Education_Level' and 'Years_of_education' (r = .69).

Lastly, it's essential to discuss the 'Ends_Meet' variable, which shows a positive correlation with cognitive score (r = 0.23). This relationship might reflect the influence of economic stability on cognitive health, a consideration that intersects with social determinants of health.

While the heatmap provides a valuable overview of pairwise relationships, the interpretation of these associations must be done in the context of an interpretable, multivariate framework that can provide a more nuanced picture of how these features collectively interact with cognitive scores.

4 Model Fitting and Tuning

Once we determined this reasonable set of parameters as a starting point for the model, we turned towards fitting a model. To begin, a baseline linear model is developed. This model only included the linear terms for each of the variables, with no interactions or other modifications. This, then, is a multiple linear regression. Because of this sharp cutoff, a better approach is required for the final model. We used normalization here because it allowed us to directly compare the weights between variables without scaling. It is difficult to quantify a one-year difference in age versus a one-unit change in BMI, but by scaling the values in terms of their variance, the weights can easily be compared. Additionally, we do not necessarily care about the specific effects on cognitive score from each of the variables because cognitive score in itself is a proxy for our desire to track dementia, so not being able to compare predictor change to cognitive score change in the units of the predictor is not a significant downside.

Several variables that were included in the EDA section were dropped whilst tuning the model. In most of these cases, this was because of multicollinearity with other variables. As introduced above, this occurs when two variables have similar effects on the cognitive score, so the model struggles to tell how each of them affects the cognitive score on its own, so the weights have no meaningful interpretation in the model. For example, alcohol consumption per week, when tracked in a single linear regression against cognitive score, shows a negative correlation. However, in a model containing each of these other variables, the weight associated with alcohol consumption when accounting for each of the other predictors was very positive. Again, this is not because of any positive effects of alcohol on cognitive score, but because the optimal model fit is to assign even more negative weights to many of the other predictors correlated with depression score, and then a positive weight to depression score to offset them. For this reason, we dropped years of education, body mass index, smoking habits, alcohol consumption per week, and sleep as predictors: they caused issues with the interpretation of the model, even though the R^2 increased slightly with them included.

To reiterate, this does not mean that these factors are not significant in predicting dementia.

Depression, BMI, and smoking habits all have negative correlation with cognitive score when taken individually, or in many subsets of the predictors; sleep and years of education had a positive correlation. Therefore, we would recommend that all of these factors be improved in order to minimize the risks of dementia. But these seemed to be less significant than many of the other predictors that we use in the model, so we choose to drop them in order to make a model with more consistent interpretations of the more significant predictors.

[24]:
```
#Fix up datasets for modelling
X_train = df.drop('Cognitive_Score', axis = 1).iloc[:,[0,1,2,4,8,9,10,11,13]]
y_train = df['Cognitive_Score']
X_test = df_test.drop('Cognitive_Score', axis = 1).iloc[:
 ↪,[0,1,2,4,8,9,10,11,13]]
y_test = df_test['Cognitive_Score']
dl = len(X_train.columns)

#One hot encoding for the categorical preprocessing
cat_preprocessing = Pipeline([
    ('onehot',OneHotEncoder(drop='first'))
])

#Standard scaler for numerical preprocessing
num_preprocessing = Pipeline([
    ('scaler',StandardScaler())
])

# Overall ML pipeline
reg_pipe = Pipeline([
     ("pre_processing", ColumnTransformer([
         ("cat_pre", cat_preprocessing, [0]),
         ("num_pre", 'passthrough', [i for i in range(1,dl-1)])])),
    ("model", LinearRegression())
])
#Fit the model
fit = reg_pipe.fit(X_train, y_train)
#Check the fit for both the test and train datasets
model_plot(fit, X_train, y_train, test=False)
```

Fig. 5. Model test plots for the baseline linear model, using the train dataset. The left plot contains the observed cognitive score values on the x-axis, and the predicted cognitive score values on the y-axis, along with a y=x line, indicating the ideal scenario where the observed value and predicted values are equivalent. The right plot contains residuals for the model plotted against the fitted cognitive scores.

From the Actual vs. Fitted plot in the left half of Figure 5, the linear model very generally underestimates cogscore for the people with predictors associated with dementia, and overestimates cogscore for the people with predictors that suggest they are less likely to be at risk. This is likely simply due to a relative lack of structure in these values.

On the right hand side, we see the residuals for this fit. These values are slightly concerning: usually, a residual plot should have even spread on both axes. Linear regression assumes that the predicted values are spread normally about the regression line, and a lack of pattern in residuals as a function of estimate corresponds with this assumption. While most of the residual plot looks reasonable for this assumption, there is an extended line serving as a lower bound for residuals for cogscore estimates between 5 and 15. This suggests that there may be some sort of structure within our data that the model fails to capture. In particular, our model seems to be biased towards the mean cogscores and consequently seems to over or underestimate the more extreme values.

[25]:
```
lms_metrics_test = model_plot(fit, X_test, y_test)
```

Fig. 6. Model test plots for the baseline linear model, using the test dataset. The plots are oriented the same as in Fig. 5.

Fig. 6 shows the same plots, but for the test dataset. The residual plot here looks better than in the train case, but the actual vs. fitted plot shows the same general issues, although the spread looks slightly better here. In fact, upon close inspection, this may only be due to a lower volume of points. Finally, we compute metrics for each of the models.

[26]:
```
print(f'Train metrics:{model_fit(fit, X_train, y_train)}')
print(f'Test metrics:{model_fit(fit,X_test,y_test)}')
```

```
Train metrics:(11.1501, 3.3392, 0.3217)
Test metrics:(11.5237, 3.3947, 0.3264)
```

To evaluate each of our models, we compute the mean squares error, root mean squares error, and R^2, which are displayed for the train dataset and test dataset above. The mean squares error terms are useful for comparing models, but are difficult to interpret on their own. The R^2, value, on the other hand, represents how much of the variance in the cogscore values can be explained by the model. We see that in the training case, about 32.2% of the variance in the data can be explained by the model, and this number rises to 32.6% in the test case. These numbers are certainly not high, but this can be expected for something like dementia: there are many other genetic factors involved that we do not have access to, and cogscores themselves are predictors of dementia, not diagnoses of it. Following the use of the linear model, we then considered several extensions to attempt to find a better fit. The first of these was the addition of polynomial terms.

In a polynomial fit, predictor columns are multiplied together into quadratic terms, in which two of the same columns are multiplied together, and interaction terms, in which two separate columns are multiplied together. We only seriously tested polynomial models up to degree 2. Although it is plausible that higher order models produce better fits, we quickly end up with a blackbox model. A degree 3 model, for example, resulted in a slightly better R^2 than the baseline linear model, but

used well over 200 predictors. Because many of these predictors imply opposite conclusions about the same variables (one interaction may indicate a positive correlation with cogscore, while another may indicate a negative correlation), and many of them display strong multicollinearity, it is near impossible to make any conclusions outside of the scope of the model, which we care about.

Therefore, we tested the degree 2 expansion of the predictors. Since we wanted to avoid including every single degree 2 term (since there are so many that it would make the model impossible to interpret, as in the degree 3 case), we used a process to select the included terms. We generated a series of individual simple linear regressions, each predicting cognitive score as a function of just the interaction term. Then, we took the 6 interactions/quadratic terms with the highest R^2 values for their individual regressions, and added these to the baseline linear model. This number of terms was selected after trial and error: fewer than this reduced the R^2 value of the model significantly, while more than this quickly had diminishing returns. Therefore, in order to optimize the performance without adding too much complexity, we used this value of 6 additional terms.

The following code adds these terms as columns in the dataset:

[27]:
```
# Generate a list of all possible combinations of factors
columns_list = X_train.columns
interactions = list(combinations_with_replacement(columns_list, 2))

# Dictionary of interactions and R^2 scores
interaction_dict = {}

# Generate the interaction column for each combination, and compute the␣
 ↪correlation with cognitive score
for interaction in interactions:
    X_train['int'] = X_train[interaction[0]] * X_train[interaction[1]]
    lr3 = LinearRegression()
    lr3.fit(X_train, y_train)
    interaction_dict[lr3.score(X_train, y_train)] = interaction
    X_train = X_train.drop('int', axis=1)  # Remove the 'int' column after each␣
 ↪iteration

# Store the number of terms as desired. +1 term accounts for indexing; this␣
 ↪includes 6 values
n = 7 +1
top_n = sorted(interaction_dict.keys(), reverse = True)[:n]

# Add new column for each of the top n interactions
for score in top_n:
    col1, col2 = interaction_dict[score]
    new_col_name = f"{col1}_{col2}"
    X_train.loc[:, new_col_name] = (X_train[col1] * X_train[col2])
    X_test.loc[:, new_col_name] = (X_test[col1] * X_test[col2])
```

Then, using these additional features, we can perform the same tests as for the baseline linear model. Fig. 3 shows the same plots as above for the training dataset.

[28]:
```
# Overall ML pipeline inlcuding all
reg_pipe_2 = Pipeline([
     ("pre_processing", ColumnTransformer([
          ("cat_pre", cat_preprocessing, [0]),
          ("num_pre", num_preprocessing,  list(range(1,n+dl-5)))
     ],
          remainder = 'passthrough')),
     ("model", LinearRegression())
])

fit2 = reg_pipe_2.fit(X_train, y_train)

model_plot(fit2, X_train, y_train, test=False)
```

Fig. 7. Model test plots for the train dataset for the model including interactions.

These plots are not more informative than the original baseline ones, and are therefore not reproduced for the test dataset. More crucially, we can also produce the evaluative measures for this model, both for the train and test datasets.

[29]:
```
print(f'Train metrics:{model_fit(fit2, X_train, y_train)}')
print(f'Test metrics:{model_fit(fit2, X_test, y_test)}')
```

```
Train metrics:(10.9589, 3.3104, 0.3333)
Test metrics:(11.2903, 3.3601, 0.34)
```

We see that the mean squares errors and root mean squares drop in both cases for this version of the model, and the R^2 values rise to 33.2% and 33.1% for the training and test case, respectively. This is an increase of under 1% in the explanation of the variance in the model, which certainly does not appear to be very large at all. However, in the context of the accuracy of the model, this

is a reasonably large increase. Even in the case of the degree 3 polynomial model briefly discussed above, in which hundreds of terms are included (including for predictors we chose to drop because of multicollinearity) and the model is essentially uninterpretable, the R^2 values peaked around 34%. Being able to bridge nearly half of the gap between the baseline model and the degree 3 model with only 5 additional terms means that the interaction terms are worth including as an extension to the model, and will make the interpretations of the model more meaningful, even if they do complicate it to a degree.

As a next step, we then continued onto testing non-renormalizable regression techniques. For all three of these approaches, we used cross-validation to determine optimal values for the model penalty parameters. For testing larger groups of variables (such as all of the interaction terms at once), lasso was preferred, while for the smaller subsets, ridge was preferred. However, for all of these cases, the optimal parameters were extremely small: for any assortment of tested alpha values for the ridge and lasso regressions, the smallest alpha value came out to have the best mean squared error. This implies that the models are not different in any significant way from the linear baseline: indeed the mean squared errors were nearly exactly the same for the lasso and ridge regressions as for the baseline model. Therefore, the linear model was preferred, since it incorporates far less complexity.

It is also possible that lasso and ridge regression with a less intuitive set of parameters may result in a better fit. For example, we discussed one-hot encoding all of the discrete variables, which would split them into several indicators, all taken separately, which would allow their relationships with cognitive score to take whatever shape is optimal, regardless of an analytical form. At this point, lasso and/or ridge regression could be used to optimize the fit and select for variables, and the higher dimensionality of the data would likely result in more success than the predictors we tried. However, the intuition of the model becomes a problem in this case. With so many predictors, it becomes very difficult to tell exactly which ones are important, and we would not necessarily get consistent results within each predictor in our original dataset: there is no reason why more extreme values, for example, would need to have as extreme weight estimates as less extreme values.

5 Discussion & Conclusions

Based on this extensive testing of possible models and sets of predictors to include, we used the linear model including 6 quadratic terms as our final model. As discussed above, the performance of the model is good compared to the baseline linear model and the other tested models, especially adjusted for complexity. However, the R^2 value of 0.335 for the test dataset still leaves a lot of unexplained variance in the dataset. Even though the model does do a reasonably good job predicting cognitive score from the set of predictors it includes, we still have unexplained variance, perhaps from missing information (such as genetics) or from random effects in the data.

We can see below a bar graph showing the predictors alongside their weights. When two predictors are listed together, this means that the term in question is an interaction term, and if the same term is listed twice, this means that the term is multiplied by itself.

[30]:
```
# Columns of training data
cols = X_train.columns.tolist()

# Produce table of coefficients
coeff_table = pd.DataFrame({
```

```
    'Feature': cols,
    'Coefficient': get_coefs(fit2)
})

# Sort by size descending
coeff_table = coeff_table.sort_values(by='Coefficient', ascending=False)

plt.figure(figsize=(10,6))
sns.barplot(y=coeff_table.Feature, x=coeff_table.Coefficient)

plt.xlabel('Weight')
plt.ylabel('Feature')

plt.show()
```

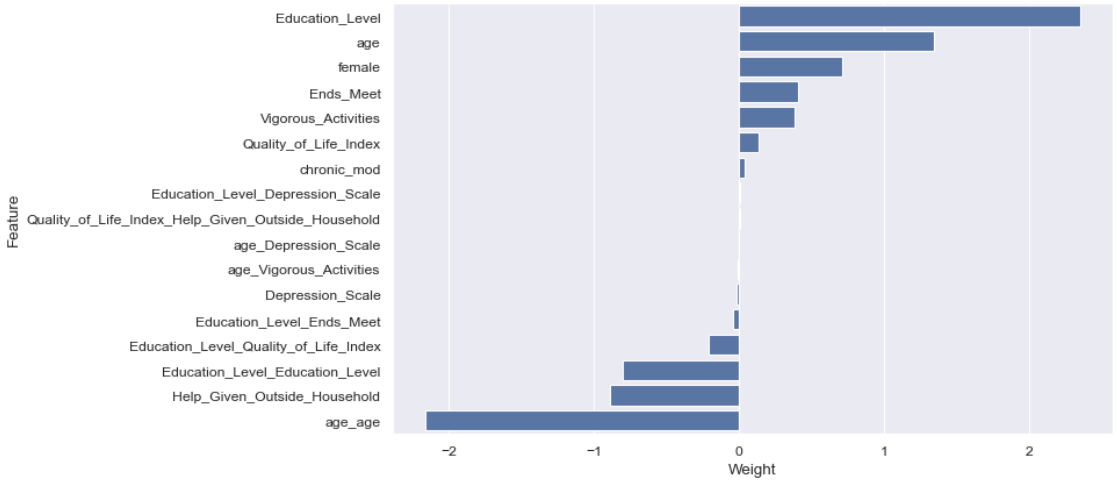

Fig. 8. Predictor vs. Model Weight for each of the final predictors used in the model. The predictors consist of the 10 chosen values from the dataset as well as 6 quadratic terms, which consist of 4 interactions and two squared predictors.

Note again that all of the predictors are standardized, so the weights can be thought of as being on the same scale. The age weight, for example, does not reflect the change in cognitive score associated with a one year age increase, but with a one standard deviation age increase: the same is true for the rest of the variables. From the graph, we can see that strongest positive features are education level, age, sex, the ability to make ends meet, and vigorous activities; the strongest negative features are the age squared term, help given outside the household, and the education level squared term.

The other terms in the model, including chronic diseases, quality of life index, depression, and the other interaction terms, are in the middle: because the model is designed such that the stronger weights are more important, we then omit these quantities from further discussion. At a glance, these values may also be victims of multicollinearity (we see in the literature, for example, that depression has a significant association with dementia) [7], but because their weights are compar-

atively small, we can consider the other predictors instead. We can then focus individually on the high-weight predictors.

Education level has by far the strongest positive coefficient. This is the measurement of education level, coded between 1 and 6, as defined in the dataset. There are also several negative interaction terms associated with education level. Because the magnitude of these terms are much smaller than the positive linear term, and because we are working with a standardized dataset (so the quadratic term is not as much of a concern as it may be otherwise), we can still conclude that education level is extremely important for predicting dementia. (We can also see from the box plots above that there are no outliers and therefore few high-variance terms for education level, minimizing the quadratic term.) The interaction terms make a clear interpretation difficult. Still, the high magnitude of the term, combined with the univariate observations of education level and the conclusions in other studies, such as [7], make this a high confidence assessment.

Age also appears in many of the terms in the most important terms in the model. Here, we can make the opposite assessment of the education level terms. The highest coefficient for age is the negative quadratic one: this means that as age gets further from the mean value (67 years), we see an increasingly large drop in cognitive score. This offsets the linear term very quickly: the quadratic term becomes more important in under one standard deviation above the mean. Though this term is clearly important, and therefore must be accounted for in the model, in terms of a plan of action, this is nearly impossible to address, since we cannot do anything about the aging of the population. Sex is also important, but similar: although we do see that women score higher on the cognitive score assessment than men, we do not see any interactions with sex in the most important terms, meaning that our plan of action should be the same for men and women. There is no reason to emphasize different priorities for each.

Vigorous activity and ends meeting are the other two important positive predictors. Vigorous activity here actually demonstrates the opposite relationship we would expect in the model: based on the single variable correlation, and the conclusions in other studies, we would expect to see a negative weight. This would imply that the term is the victim of multicollinearity; however, the model drops significantly in accuracy if we drop the term, so it is included, though we do not recommend acting on this portion of the model.

Ends meeting, on the other hand, does align with our other model portions and expectations. This is a measure of stress on the people in the study: we do not see this in any interaction terms, so the coefficient can be interpreted quite clearly: the ability to make ends meet has a positive effet on cognitive score.

Help given outside the household is the remaining negative weight. Since the measurement is defined such that giving help outside the household is measured as a lower value than not giving the help, this measurement is also as expected, given the conclusions in the literature.

It should again be noted that the predictive power of the model is relatively low. The recommendations are based off associations between variables when accounting for other ones, but we often do not have the underlying causes for dementia, and some of the predictors may not be causal for dementia. Still, taken individually, all of the highlighted predictors have the same relationship with cognitive score; we therefore believe the model demonstrates substantial linkage in the data despite the 33.1% R^2 value.

Based on these results, we recommend a three-part plan for action to minimize dementia in the coming years. These parts can be broken into two different categories: one with a focus on minimizing

its impact in the near future, and one on prevention in the far future.

The first portion of the plan is to improve the lives of people over the age of 65 in ways that will minimize the risk of dementia. We see above that two of the most important factors in our model are the ability to make ends meet, and consistent helping of those outside the household. The ability to make ends meet is clearly relatively difficult to change at the government level: raising the quality of life for a large portion of the population in a way that will allow them to make ends meet more easily will require a huge investment of time, money, and human resources. Still, for preventing debilitating effects of dementia on the economy in the near future, this is the single best solution. People who do not have stress about paying for food or rent each month, are less likely to have their cognitive scores degrade. Helping outside the home is a much easier way to offset dementia. Although the effects are much weaker in the model, this type of community activity is much easier and cheaper to use as a strategy. We therefore recommend that in addition to improving their ability to make ends meet with social programs, the government should promote and fund opportunities for senior citizens to participate in productive community activities, whether alongside peers or otherwise.

It is crucial to emphasize the importance of early intervention and prevention strategies, as dementia risk factors can accumulate over the life course [10]. While some interventions may be more challenging to implement, they are essential for reducing the long-term burden of dementia on individuals, families, and society. Implementing these recommendations may require significant resources, but the long-term benefits of reduced dementia prevalence could lead to substantial cost savings in healthcare and social care systems [11].

Finally, as a long-term solution for the problem, we see that education is extremely important to maintaining a high cognitive score. Investing in programs to make sure that a higher percentage of the population receives as many years of education as possible will prevent dementia costs from rising out of control when that portion of the population becomes old enough that dementia is a concern. While this has certainly been a priority in recent years for governments in most of Europe, it is important that this emphasis continues as life spans increase and dementia becomes a concern for an increasing percentage of the population.

Though we do see a relative lack of model accuracy, we can be confident about these recommendations because of the preponderance of evidence in a variety of sources in support of these types of measures. In [7], for example, the authors also point to education as the single most important factor in long-term prevention of dementia. That we see the same result in our model points to its importance, and a higher degree of confidence in this fact than taking either model on its own.

However, it is important to acknowledge the limitations of the study, such as the lack of genetic data and other potential risk factors not captured in the dataset. While the model's predictive power may be limited, the identified risk factors align with existing literature, strengthening the validity of the recommendations [12].

Future research should investigate the effectiveness of specific interventions targeting the identified risk factors. Collaboration between researchers, policymakers, and healthcare providers is essential to develop evidence-based strategies for dementia prevention and management [13]. Additionally, raising public awareness about dementia risk factors and the potential for prevention is crucial. Developing public health campaigns and educational initiatives to promote healthy lifestyles and social engagement among older adults should be a priority [14].

6 References

[1] Axel Börsch-Supan. Survey of Health, Ageing and Retirement in Europe (SHARE) Wave 4. en. 2022. doi: 10.6103/SHARE.W4.800. url: http://www.share-project.org/data-documentation/waves-overview/wave-4.html.

[2] Axel Börsch-Supan. Survey of Health, Ageing and Retirement in Europe (SHARE) Wave 6. en. 2022. doi: 10.6103/SHARE.W6.800. url: http://www.share-project.org/data-documentation/waves-overview/wave-6.html.

[3] Axel Börsch-Supan and Stefan Gruber. easySHARE. 2022. doi: 10.6103/SHARE.EASY.800. url: http://www.share-project.org/special-data-sets/easyshare.html.

[4] Mayo Clinic. Dementia. url: https://www.mayoclinic.org/diseases-conditions/dementia/symptoms-causes/syc-20352013.

[5] Centers for Disease Control and Prevention. What is Dementia? Apr. 2019. url: https://www.cdc.gov/aging/dementia/index.html#:~:text=Dementia%5C%20is%5C%20not%5C%20a%5C%20sp

[6] Céline Ben Hassen et al. "Association between age at onset of multimorbidity and incidence of dementia: 30 year follow-up in Whitehall II prospective cohort study". In: BMJ 376 (Feb. 2022), e068005. doi: 10.1136/bmj-2021-068005. url: https://www.bmj.com/content/376/bmj-2021-068005.

[7] Gill Livingston et al. "Dementia prevention, intervention, and care: 2020 report of the Lancet Commission". In: The Lancet 396.10248 (2020), pp. 413–446. issn: 0140-6736. doi: https://doi.org/10.1016/S0140-6736(20)30367-6. url: https://www.sciencedirect.com/science/article/pii/S0140673620303676.

[8] NHS. Dementia guide. Dec. 2017. url: https://www.nhs.uk/conditions/dementia/.

[9] Alzheimer's Society. What Is dementia? — Alzheimer's Society. 2021. url: https://www.alzheimers.org.uk/about-dementia/types-dementia/what-is-dementia.

[10] Livingston, G., Huntley, J., Sommerlad, A., Ames, D., Ballard, C., Banerjee, S., … & Mukadam, N. (2020). Dementia prevention, intervention, and care: 2020 report of the Lancet Commission. The Lancet, 396(10248), 413-446. https://doi.org/10.1016/S0140-6736(20)30367-6

[11] Wimo, A., Guerchet, M., Ali, G. C., Wu, Y. T., Prina, A. M., Winblad, B., … & Prince, M. (2017). The worldwide costs of dementia 2015 and comparisons with 2010. Alzheimer's & Dementia, 13(1), 1-7. https://doi.org/10.1016/j.jalz.2016.07.150

[12] Brayne, C., & Miller, B. (2017). Dementia and aging populations—A global priority for contextualized research and health policy. PLoS Medicine, 14(3), e1002275. https://doi.org/10.1371/journal.pmed.1002275

[13] Kivipelto, M., Mangialasche, F., & Ngandu, T. (2018). Lifestyle interventions to prevent cognitive impairment, dementia and Alzheimer disease. Nature Reviews Neurology, 14(11), 653-666. https://doi.org/10.1038/s41582-018-0070-3

[14] Cations, M., Radisic, G., Crotty, M., & Laver, K. E. (2018). What does the general public understand about prevention and treatment of dementia? A systematic review of population-based surveys. PLoS One, 13(4), e0196085. https://doi.org/10.1371/journal.pone.0196085

[31]:
```
# Run the following to render to PDF
!jupyter nbconvert --to pdf main_v3.ipynb
```

```
usage: jupyter [-h] [--version] [--config-dir] [--data-dir] [--runtime-dir]
               [--paths] [--json] [--debug]
               [subcommand]

Jupyter: Interactive Computing

positional arguments:
  subcommand     the subcommand to launch

options:
  -h, --help     show this help message and exit
  --version      show the versions of core jupyter packages and exit
  --config-dir   show Jupyter config dir
  --data-dir     show Jupyter data dir
  --runtime-dir  show Jupyter runtime dir
  --paths        show all Jupyter paths. Add --json for machine-readable
                 format.
  --json         output paths as machine-readable json
  --debug        output debug information about paths

Available subcommands: kernel kernelspec migrate run troubleshoot

Jupyter command `jupyter-nbconvert` not found.
```

[ ]:
```

```