CS123A

# Alzheimer's Disease

By George Wilfert and Nick Tran

# Motivation & Objectives

[ MRI & Clinical Data ] ⟶ [ Machine Learning Model ] ⟶ [ Dementia Prediction ]

Why it matters

- Early detection of dementia can lead to timely interventions and better patient outcomes.
- MRI-derived brain volume metrics and simple clinical tests are non-invasive and widely available.

Our objective

- Develop a reproducible pipeline that combines MRI measurements and demographic/clinical data to predict dementia status.

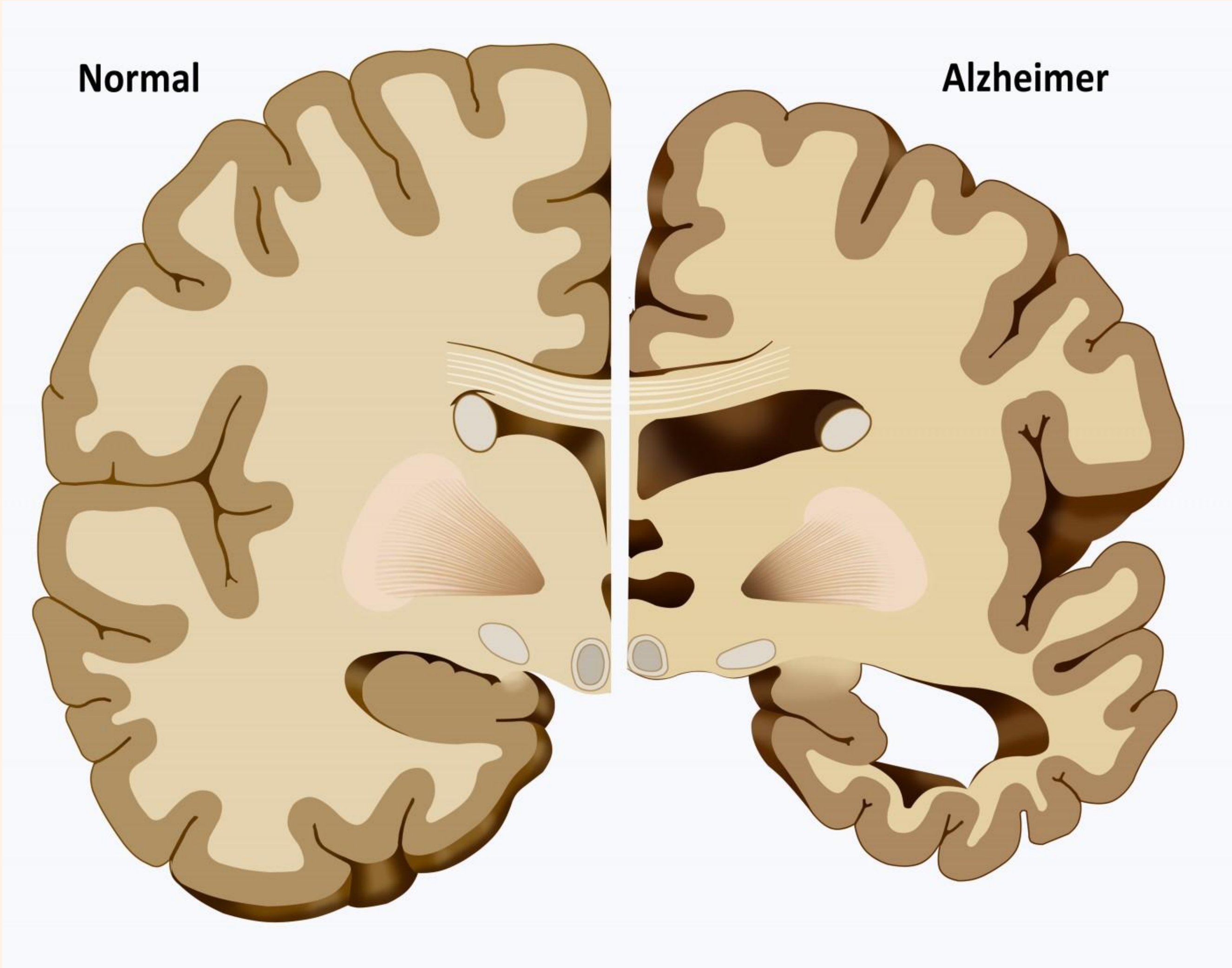
Main points:

- Can demographics + MRI features reliably separate demented vs. non-demented adults?
- Which variables drive the most predictive power?

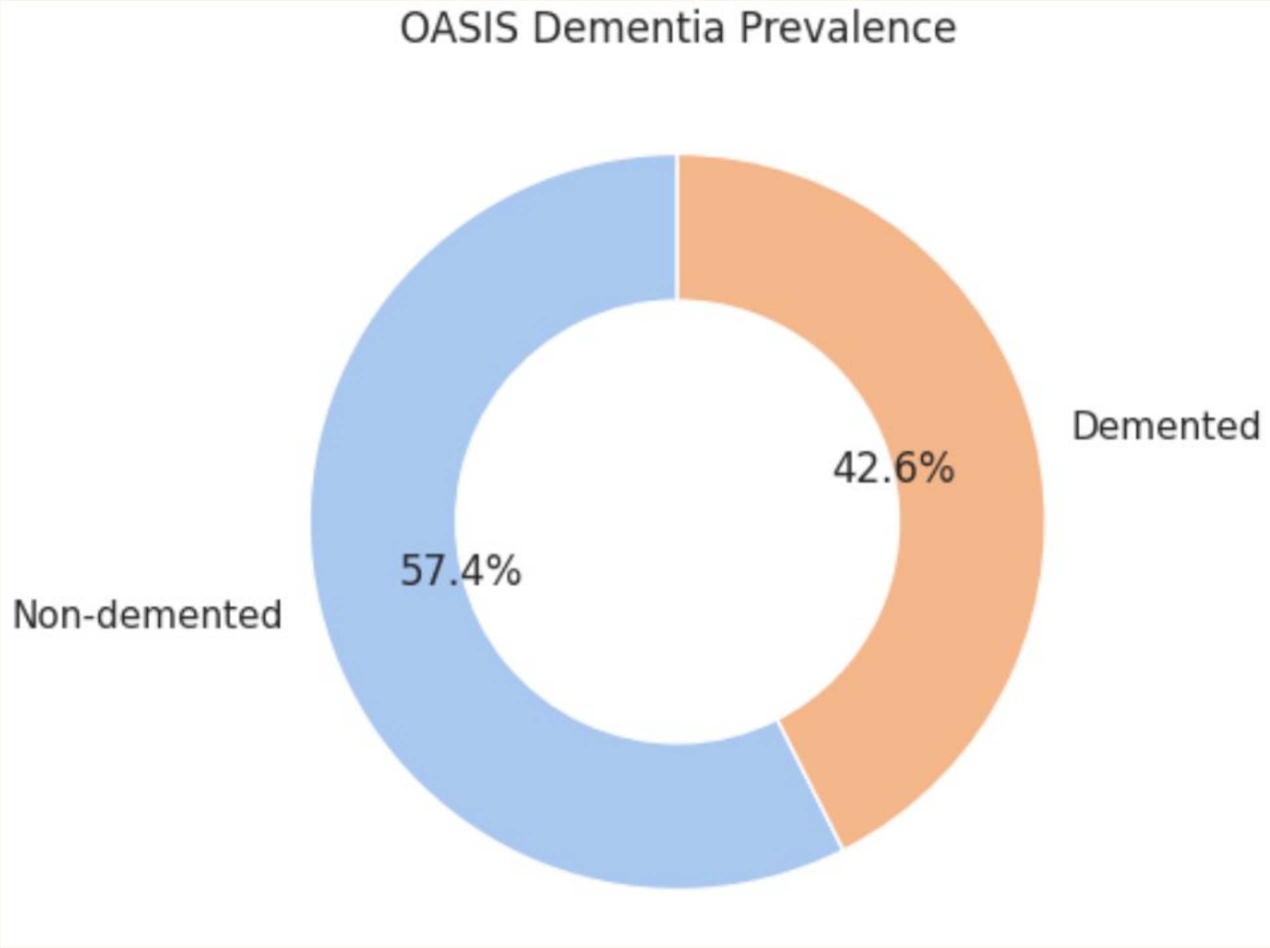
# Underlying Mechanisms

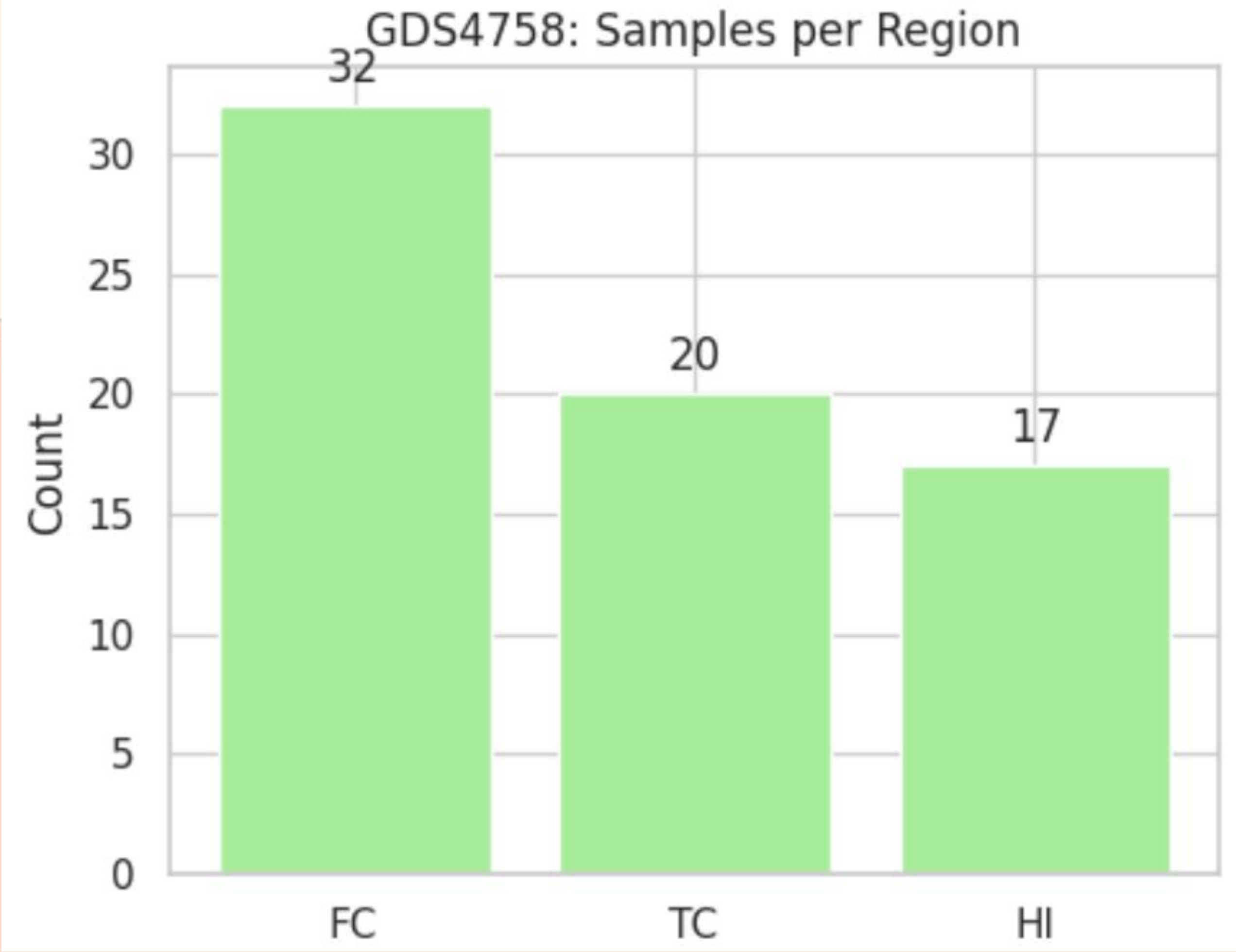
Alzheimer's disease has been found to be genetically driven through certain genes and pathologies such as APOE gene which has been linked to the buildup of plaque like proteins called Amyloid Beta which is a hallmark of the disease. There is also dysfunctional proteins such as Tau which degrades the brain's ability to function. But there is also many epigenetically environmental driven factors that play a role. We will use MRI scans and a simple memory test to predict who is actually demnted, and then connect those prediction back to genes like APOE/Tau. And things like diet, stress, and toxins. Our project will look to isolate both aspects and look deeper into both.

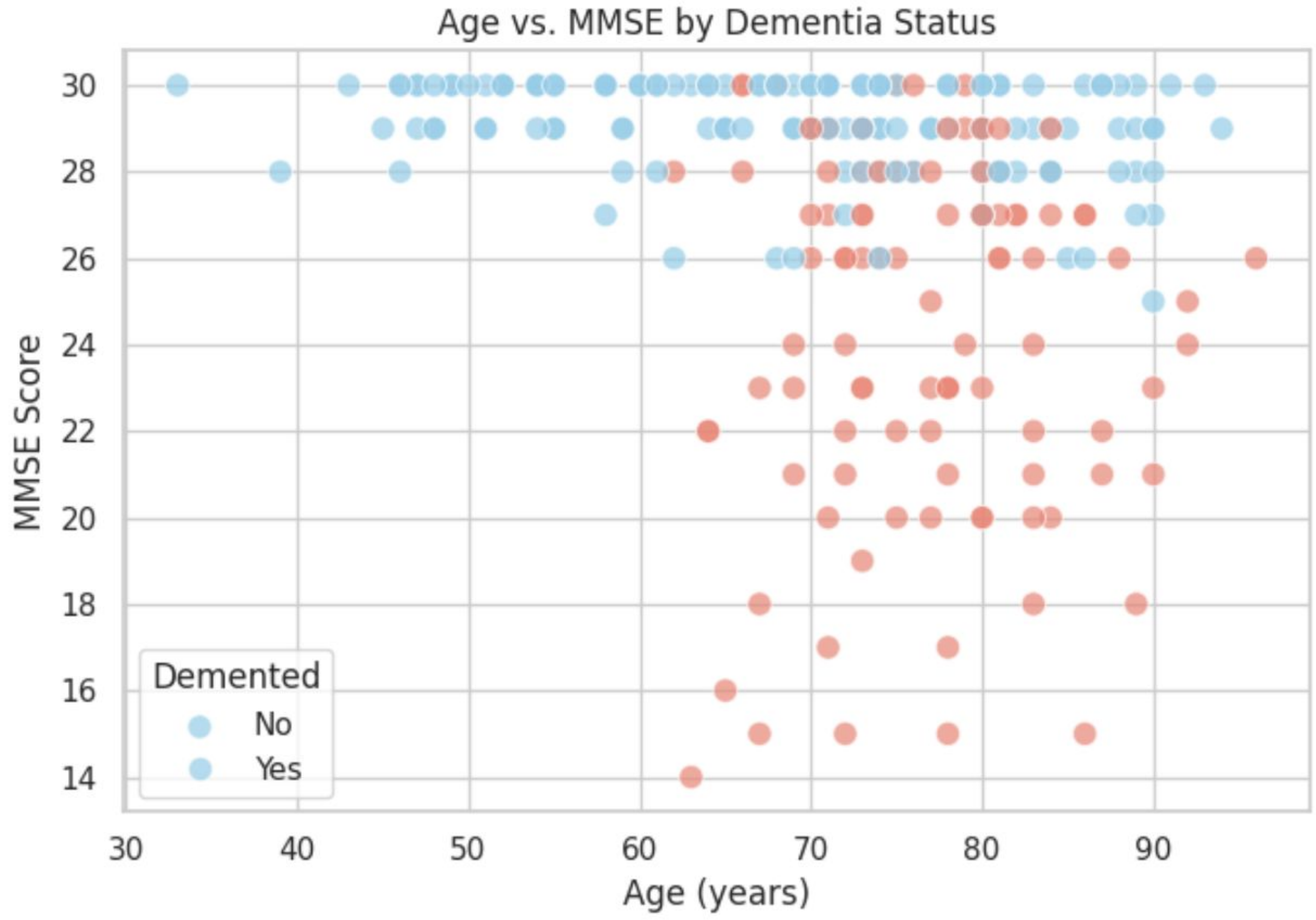
# 1. Phenotype Aspect of Project



# 1. Phenotype Distribution

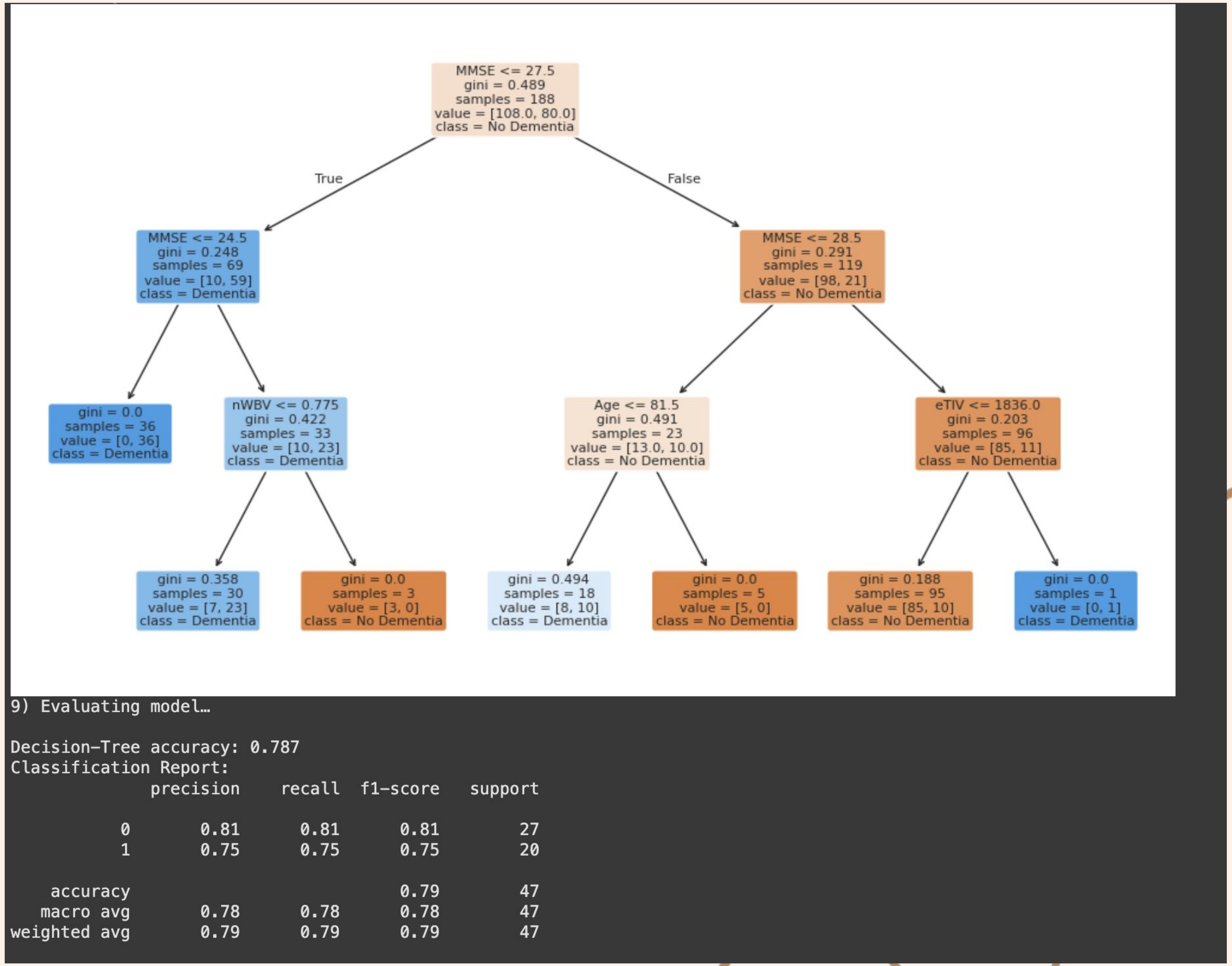






### Key Splits

- **Root:** MMSE ≤ 27.5 → high dementia risk
  - If **True**, next split on MMSE ≤ 24.5: almost all below 24.5 are demented
  - If **False**, splits on:
    - **Age** ≤ 81.5 → mostly non-demented
    - **eTIV** ≤ 1836 cc → almost all non-demented



```
9) Evaluating model…

Decision-Tree accuracy: 0.787
Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.81      0.81        27
           1       0.75      0.75      0.75        20

    accuracy                           0.79        47
   macro avg       0.78      0.78      0.78        47
weighted avg       0.79      0.79      0.79        47
```

**Performance on Test Set (n=47)**

- Accuracy: 0.79
- Non-demented (0): Precision 0.81 / Recall 0.81
- Demented (1): Precision 0.75 / Recall 0.75

# Dataset Overview

Dataset: OASIS cross-sectional MRI, 436 adult samples (~21 KB CSV from Kaggle)
Demographics: ID, gender (F/M), handedness, age range 18–96, education level, SES

Clinical: MMSE cognitive score (median 29, range 18–30) and CDR dementia rating

MRI metrics:

- eTIV (1123 – 1992 cc)
- nWBV (mean 0.78)
- ASF (scaling factor)

Labeling:

- Only 235 subjects have CDR → drop the rest
- Define non-demented (CDR = 0) vs. demented (CDR > 0) (~80% vs. 20%)
- Imputation: Fill missing values in Educ, SES, and MMSE with their cohort medians

# 2. Gene Sequencing Data

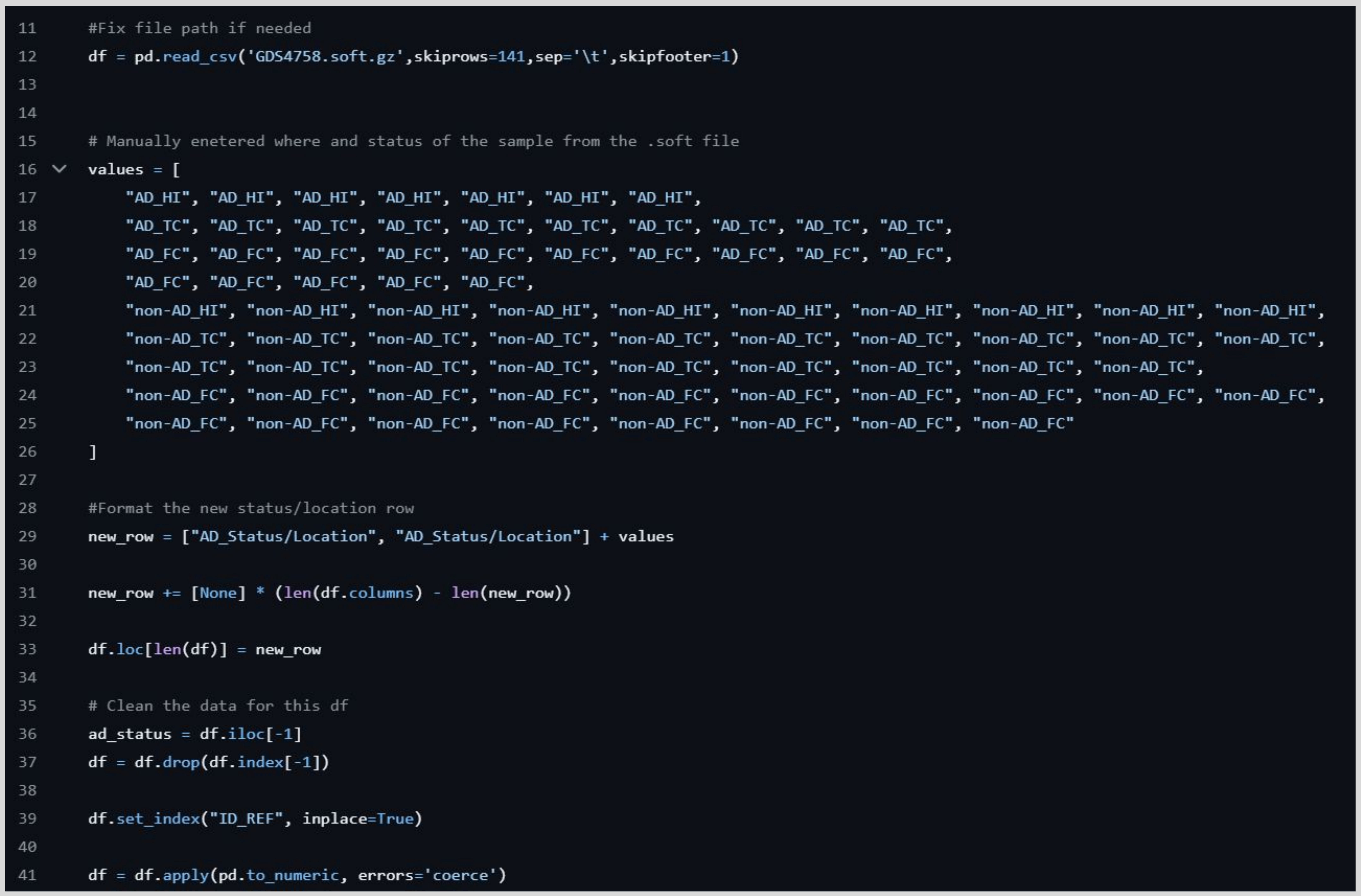
# Dataset Overview

The GDS4758 dataset from the NCBI Gene Expression Omnibus (GEO) provides gene expression profiles from postmortem brain tissues of individuals from the Hisayama study that focuses on Alzheimer's disease.

Comprising 79 samples, the dataset serves as a valuable resource for researchers investigating gene expression alterations associated with Alzheimer's disease It is a .soft file which requires extra intervention compared to strictly tabular data.

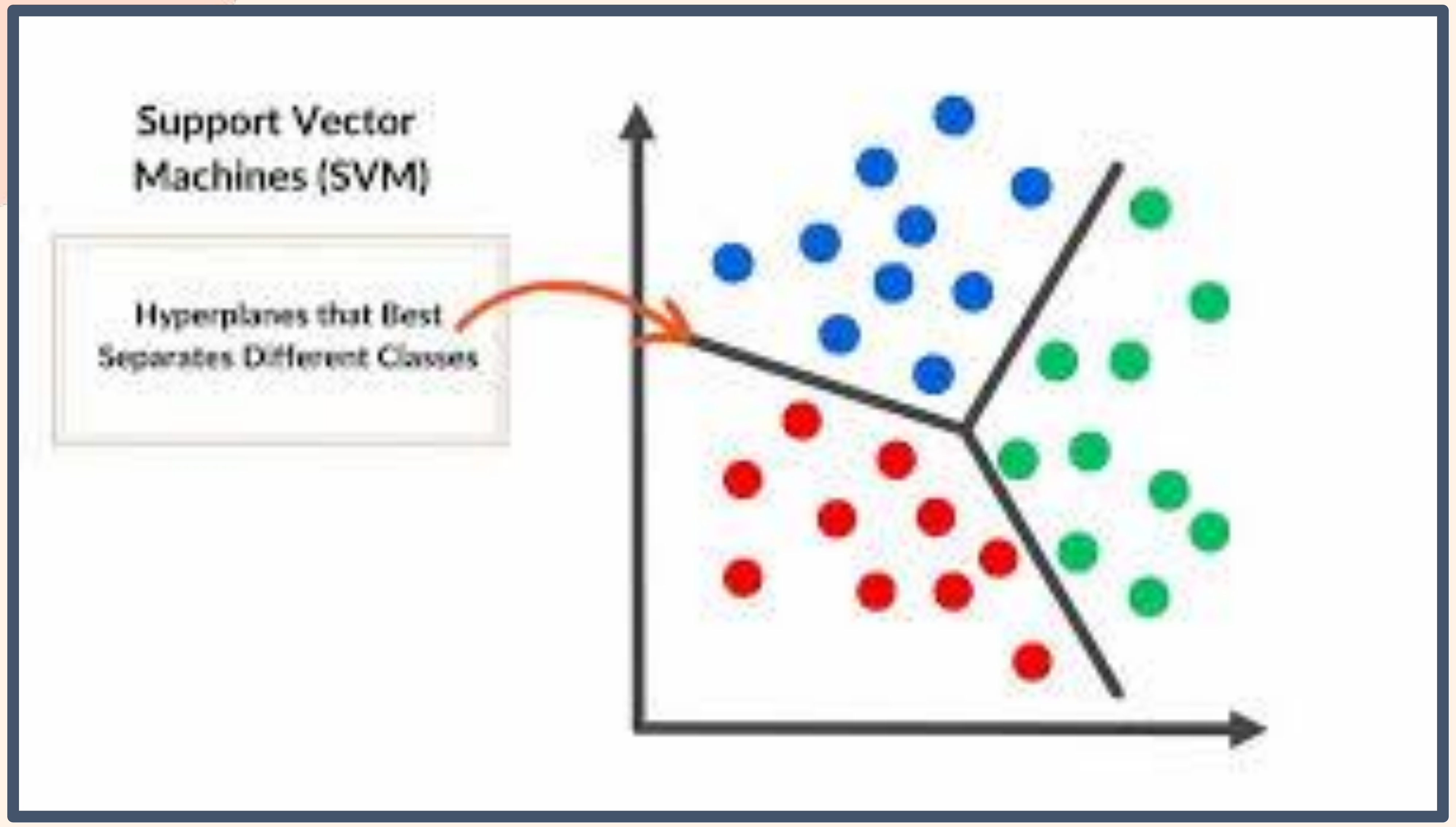
Cons: There is a relatively small sample size for machine learning applications.

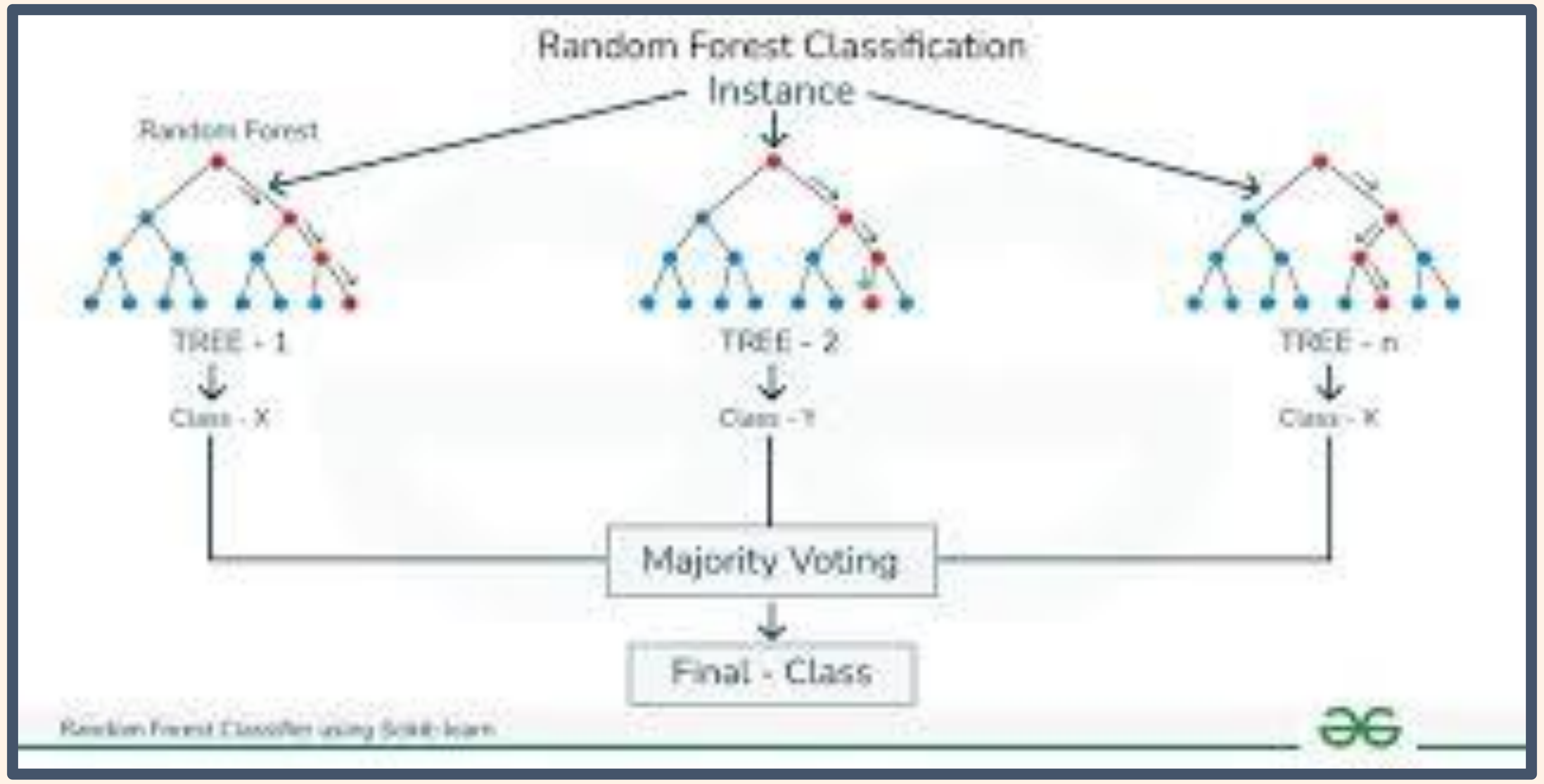
# Data Cleaning



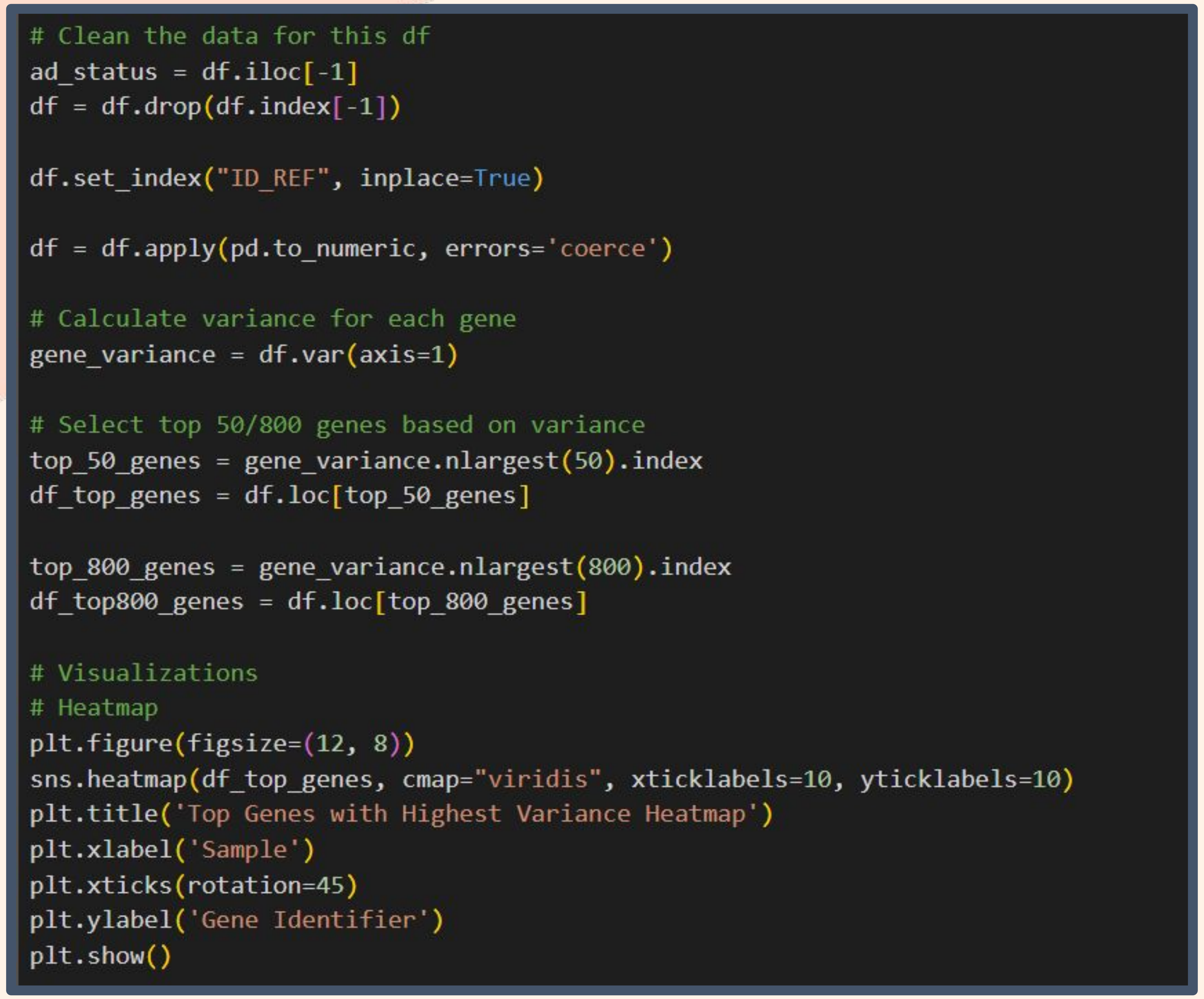
```python
#Fix file path if needed
df = pd.read_csv('GDS4758.soft.gz',skiprows=141,sep='\t',skipfooter=1)


# Manually enetered where and status of the sample from the .soft file
values = [
    "AD_HI", "AD_HI", "AD_HI", "AD_HI", "AD_HI", "AD_HI", "AD_HI",
    "AD_TC", "AD_TC", "AD_TC", "AD_TC", "AD_TC", "AD_TC", "AD_TC", "AD_TC", "AD_TC", "AD_TC",
    "AD_FC", "AD_FC", "AD_FC", "AD_FC", "AD_FC", "AD_FC", "AD_FC", "AD_FC", "AD_FC", "AD_FC",
    "AD_FC", "AD_FC", "AD_FC", "AD_FC", "AD_FC",
    "non-AD_HI", "non-AD_HI", "non-AD_HI", "non-AD_HI", "non-AD_HI", "non-AD_HI", "non-AD_HI", "non-AD_HI", "non-AD_HI", "non-AD_HI",
    "non-AD_TC", "non-AD_TC", "non-AD_TC", "non-AD_TC", "non-AD_TC", "non-AD_TC", "non-AD_TC", "non-AD_TC", "non-AD_TC", "non-AD_TC",
    "non-AD_TC", "non-AD_TC", "non-AD_TC", "non-AD_TC", "non-AD_TC", "non-AD_TC", "non-AD_TC", "non-AD_TC", "non-AD_TC",
    "non-AD_FC", "non-AD_FC", "non-AD_FC", "non-AD_FC", "non-AD_FC", "non-AD_FC", "non-AD_FC", "non-AD_FC", "non-AD_FC", "non-AD_FC",
    "non-AD_FC", "non-AD_FC", "non-AD_FC", "non-AD_FC", "non-AD_FC", "non-AD_FC", "non-AD_FC", "non-AD_FC"
]

#Format the new status/location row
new_row = ["AD_Status/Location", "AD_Status/Location"] + values

new_row += [None] * (len(df.columns) - len(new_row))

df.loc[len(df)] = new_row

# Clean the data for this df
ad_status = df.iloc[-1]
df = df.drop(df.index[-1])

df.set_index("ID_REF", inplace=True)

df = df.apply(pd.to_numeric, errors='coerce')
```

# Choice of Models

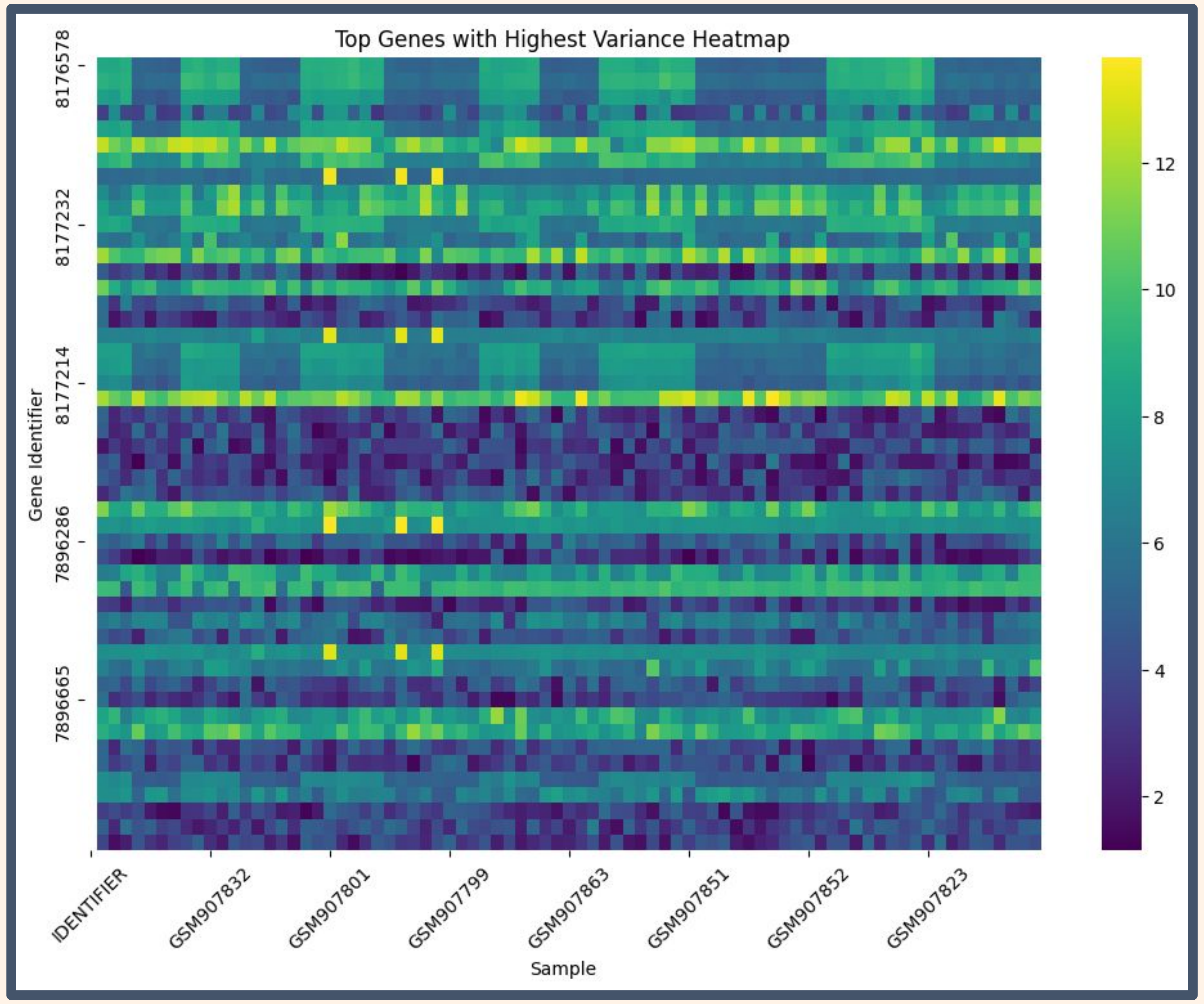




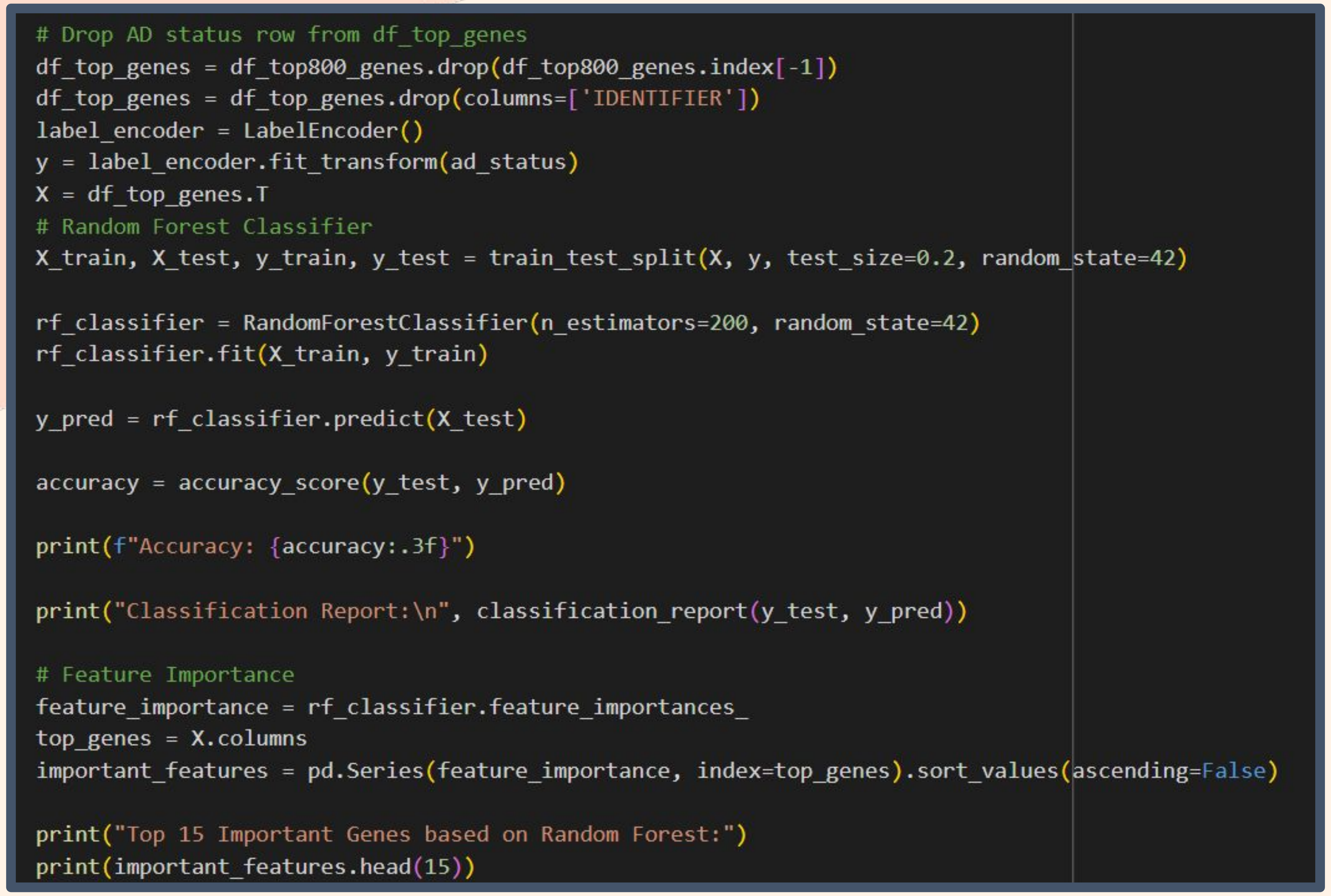
# Main Code



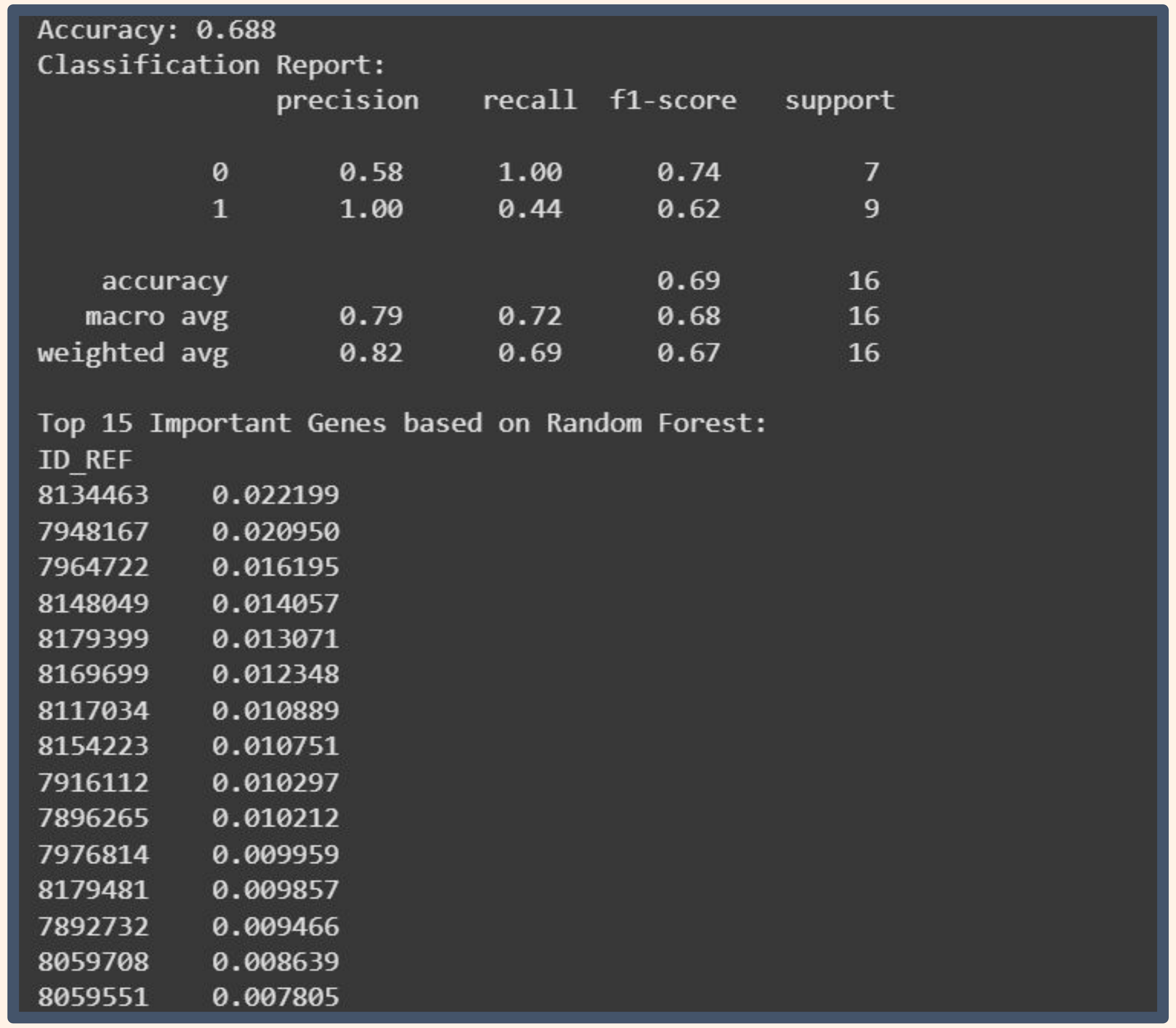
```
# Clean the data for this df
ad_status = df.iloc[-1]
df = df.drop(df.index[-1])

df.set_index("ID_REF", inplace=True)

df = df.apply(pd.to_numeric, errors='coerce')

# Calculate variance for each gene
gene_variance = df.var(axis=1)

# Select top 50/800 genes based on variance
top_50_genes = gene_variance.nlargest(50).index
df_top_genes = df.loc[top_50_genes]

top_800_genes = gene_variance.nlargest(800).index
df_top800_genes = df.loc[top_800_genes]

# Visualizations
# Heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(df_top_genes, cmap="viridis", xticklabels=10, yticklabels=10)
plt.title('Top Genes with Highest Variance Heatmap')
plt.xlabel('Sample')
plt.xticks(rotation=45)
plt.ylabel('Gene Identifier')
plt.show()
```



# Final Results



```
# Drop AD status row from df_top_genes
df_top_genes = df_top800_genes.drop(df_top800_genes.index[-1])
df_top_genes = df_top_genes.drop(columns=['IDENTIFIER'])
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(ad_status)
X = df_top_genes.T
# Random Forest Classifier
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

rf_classifier = RandomForestClassifier(n_estimators=200, random_state=42)
rf_classifier.fit(X_train, y_train)

y_pred = rf_classifier.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(f"Accuracy: {accuracy:.3f}")

print("Classification Report:\n", classification_report(y_test, y_pred))

# Feature Importance
feature_importance = rf_classifier.feature_importances_
top_genes = X.columns
important_features = pd.Series(feature_importance, index=top_genes).sort_values(ascending=False)

print("Top 15 Important Genes based on Random Forest:")
print(important_features.head(15))
```



```
Accuracy: 0.688
Classification Report:
               precision    recall  f1-score   support

           0       0.58      1.00      0.74         7
           1       1.00      0.44      0.62         9

    accuracy                           0.69        16
   macro avg       0.79      0.72      0.68        16
weighted avg       0.82      0.69      0.67        16

Top 15 Important Genes based on Random Forest:
ID_REF
8134463    0.022199
7948167    0.020950
7964722    0.016195
8148049    0.014057
8179399    0.013071
8169699    0.012348
8117034    0.010889
8154223    0.010751
7916112    0.010297
7896265    0.010212
7976814    0.009959
8179481    0.009857
7892732    0.009466
8059708    0.008639
8059551    0.007805
```

# Bio Meets Bioinformatics

Correlating those genes of high relevance to the model the highest three are Neuronal Pentraxin 2, Apelin Receptor, Wnt Inhibitory Factor 1.

These genes do quite a few things that include synaptic plasticity which is definitely important to look at because of the effect of neurons during the stages of the disease. But there is also Blood vessel development and growth of cells which are all clues for biologist to look for and how these expression of genes may change over time and progression of the disease.

# Conclusion

### 1. What We Learned

- We built a system that looks at brain scans and a few simple tests (like a memory quiz) and guesses if someone has dementia about **80–83%** of the time.

- In the brain data, the most important things were how people scored on the memory quiz (MMSE), their brain size measurements, and age.

- In the gene data, a handful of genes (like Neuronal Pentraxin 2 and Apelin Receptor) changed the most between Alzheimer's and healthy brains.

### 2. What's Next for the future

- **Test on more people:** Try our methods on new MRI scans and gene data to see if they still work well.

- **Try different computers tricks:** Use other machine-learning tools (like XGBoost) or include extra health info to boost accuracy.

- **Learn about the genes:** Study the top genes we found to understand how they might cause or protect against dementia.

# Thank you!