Predicting Alzheimer's Disease

Development and Validation of Machine Learning Models

Jay Fu

Project Number - BFXX

Alzheimer's is *prevalent*.

5.5 Million

People diagnosed of Alzheimer's per year in US

50 million → 152 million

Amount of Alzheimer's patients from now to 2050

4th

Leading cause of death in the US

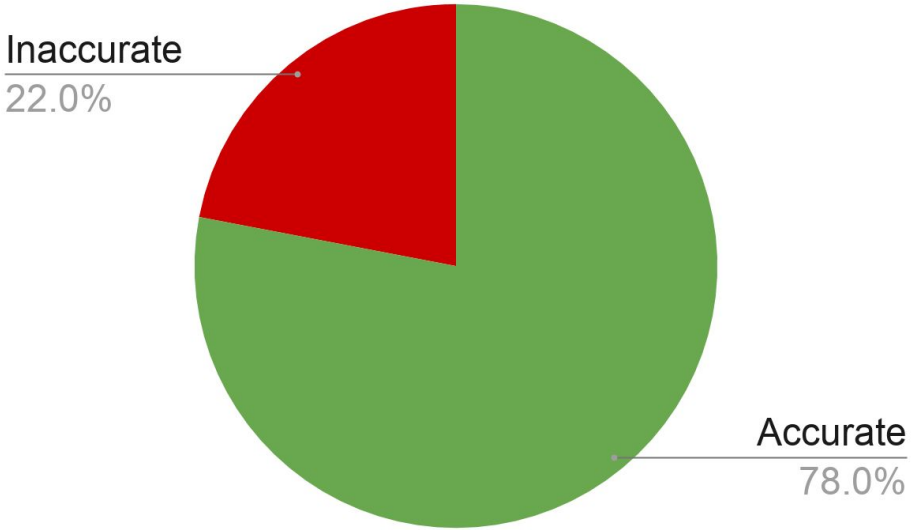
Why is clinical diagnosis accuracy for Alzheimer's important?

An **early, accurate diagnosis** will:

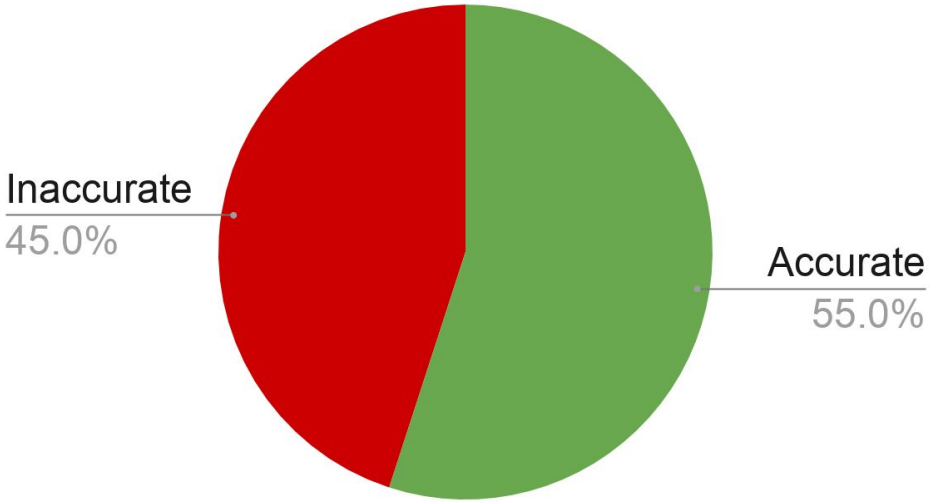
- Help doctors devise effective strategies to manage symptoms and long term care
- Allow patient to get treated earlier and experience slower memory deterioration
- Lower cost of treatment
- **Prolong the patient's life**

Problem: Poor clinical diagnosis accuracy of Alzheimer's.

Average Clinical Diagnosis Accuracy For **All American Doctors**



Average Clinical Diagnosis Accuracy For **Rural American Doctors**



Hypothesis

If **machine learning models** are trained with **easily obtainable clinical data** including MRI, cognitive assessment, and demographic data, then doctors can diagnose Alzheimer's in patients **more swiftly and accurately** than conventional methods.

The Models

- **Logistic Regression Model**
- **K-Nearest Neighbor Algorithm**
- **Support Vector Machine Model**
- **Random Forest Algorithm**
- **Feedforward Neural Network**

Creating multiple different machine learning models allows us to see which has **highest accuracy** and will be the **best to use** in a clinical setting.

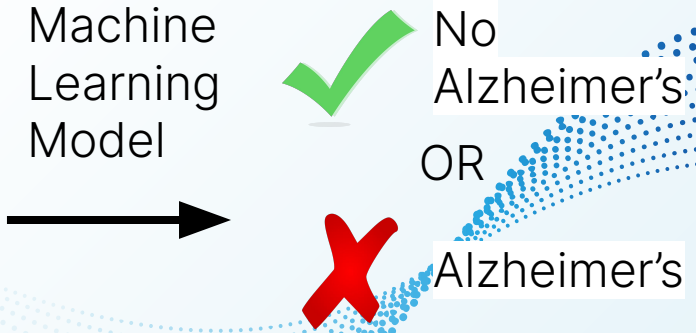
The Data

- Open Access Series of Imaging Studies (OASIS)
- 373 MRI imaging sessions

Model Input

Sex	1: male 2: female
Age	60 to 96 years old
EDUC (years of education)	6 to 23 years
eTIV (estimated total intracranial volume)	1106 to 2004 mm^3
nWBV (normalized whole-brain volume)	Proportion of tissue in brain volume, value from 0 to 1
MMSE (Mini Mental State Examination)	Score between 1 to 30

Model Output



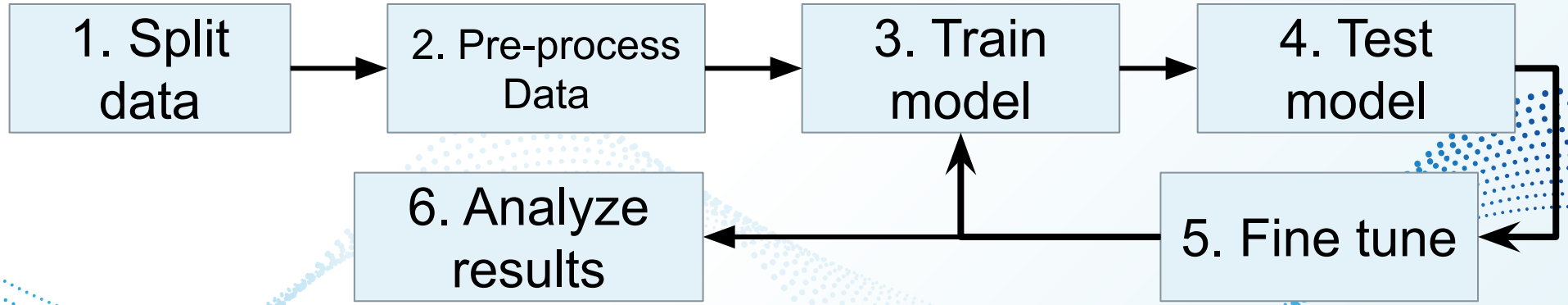
Procedure

The data is randomly split into training set (60%) and testing set (40%)

The data is normalized and optimized for the model training.

Each model is trained with the training set data.

The accuracy of prediction for each model is assessed by the testing set data.



Key risk factors are identified. Models are compared to each other. Concordance between models are computed.

The model's hyperparameters are adjusted to improve model accuracy.

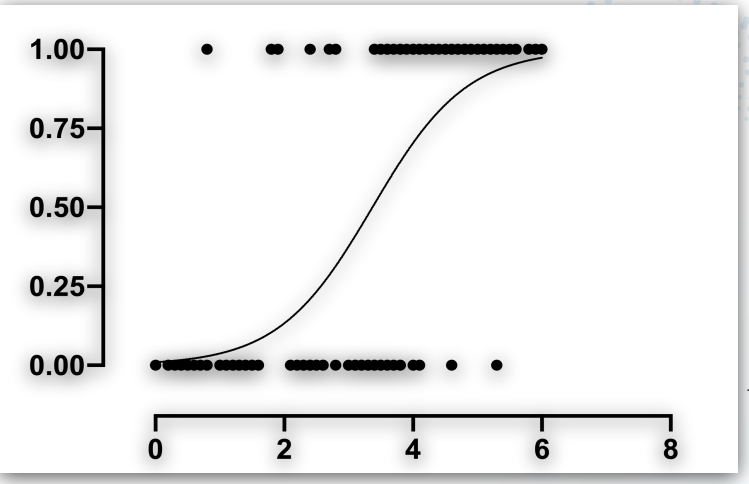
What is logistic regression?

- Logistic regression is a machine learning algorithm typically used for binary classification problems
- It is modeled by: $\ln(\frac{P}{1-P}) = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$

β_0 is the intercept

The Logit (chance that the patient has Alzheimer's)

β_1 through β_n are **regression coefficients**.
x_1 through x_n are the **independent variables**.



This plot demonstrates a Logistic Regression model with the independent variable in y axis and probability of a condition in x axis.

Logistic Regression Results

Predicted logit of Alzheimer's = 80.1293 − 31.1353 × nWBV − 0.0038 × eTIV − 0.1200 × Age − 1.3874 × MMSE − 0.2594 × Years of Education + 1.4993 × Gender

Refers to coefficient $\boldsymbol{\beta}$

Typical distance that the data points fall from regression line

Since *p* values under 0.05 are statistically significant, all variables here are significant.

The odds ratio is the natural exponential of $\boldsymbol{\beta}$. The greater the odds ratio is above 1, the greater positive correlation. The greater it is below 1, the greater negative correlation.

Predictor	Est.	Std. Error	*P*	Odds Ratio	Risk increase
Intercept	80.129	15.246	<0.001	NA	NA
nWBV	-31.135	9.858	0.0016	3e-14	-100%
eTIV	-0.0038	0.0018	0.0341	0.996	-0.38%
Age	-0.12	0.045	0.0079	0.887	-11.31%
MMSE	-1.387	0.239	<0.001	0.25	-75.03%
EDUC	-0.259	0.096	0.0068	0.772	-22.85%
Gender	1.499	0.604	0.013	4.479	347.85%

Risk increase indicates the percent risk increase of Alzheimer's if a variable is increased by 1 unit.

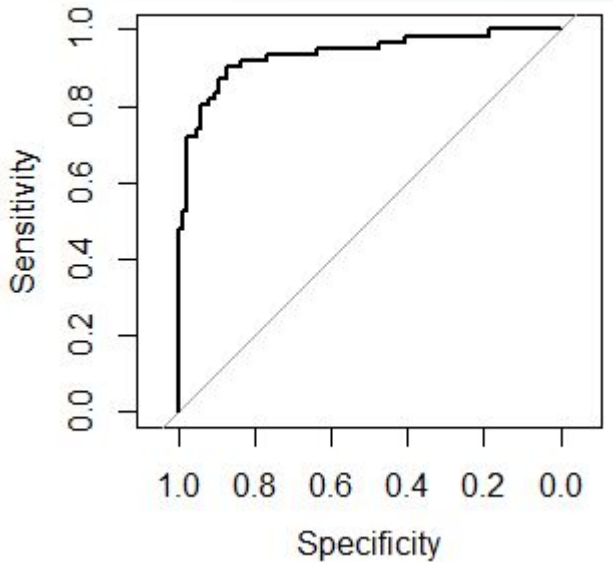
Logistic Regression Results

Confusion Matrix of Logistic Regression Model Test

Actual Condition	Predicted Alzheimer's	Predicted No Alzheimer's	% Correct
Alzheimer's	**43**	**11**	**79.63%**
No Alzheimer's	**10**	**94**	**90.38%**

Overall Accuracy: **86.71%**

ROC Plot of Logistic Regression

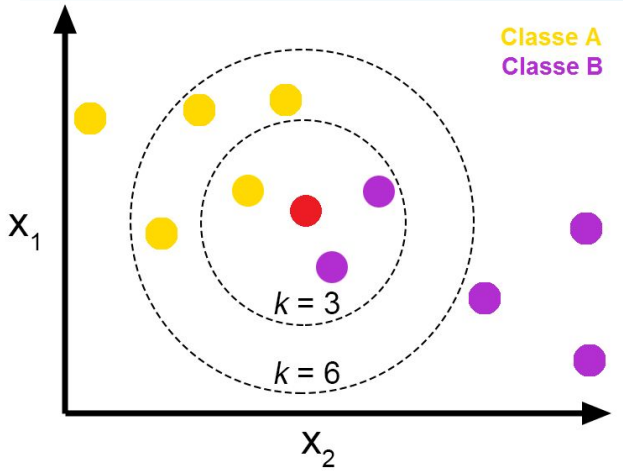


The closer the ROC curve gets to the top-left corner, the better the model.

What is K-Nearest-Neighbor (KNN) Algorithm?

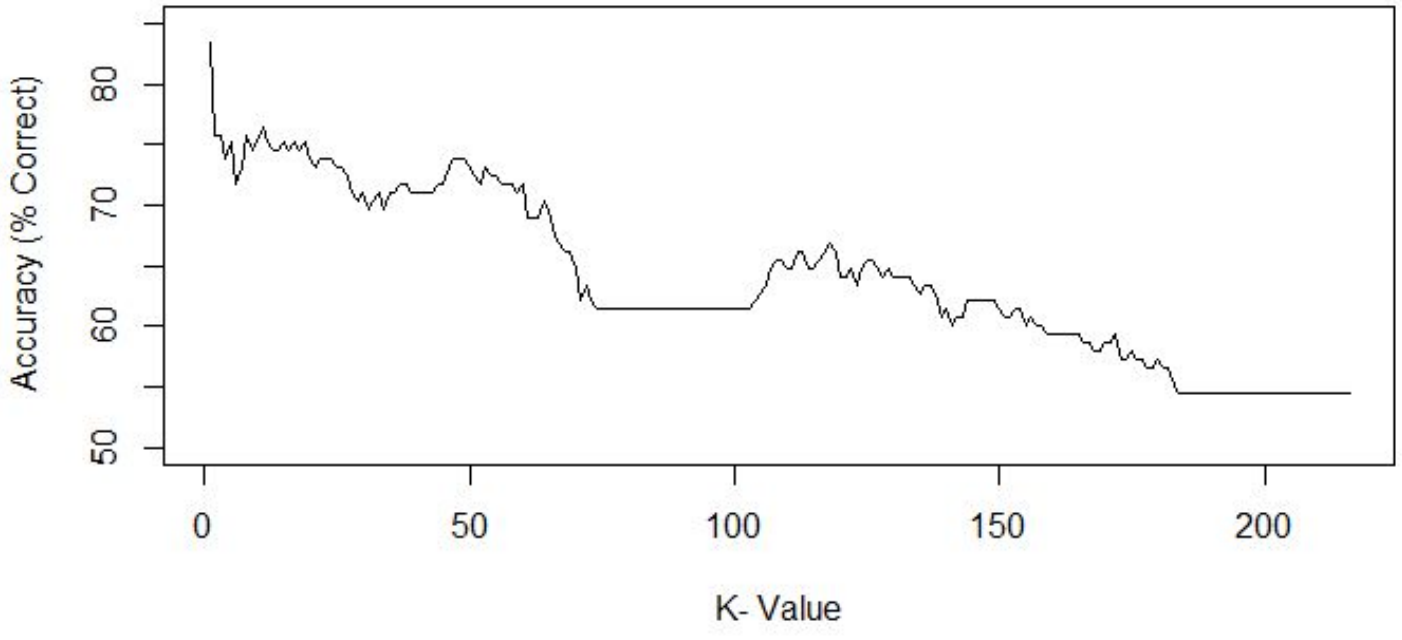
- KNN is a supervised machine learning algorithm.
- Each MRI imaging session in the data set is considered a data point and is plotted in a *n* dimensional space (where *n* is the # of independent variables)
- The algorithm determines what class a new data point is by finding what class the majority of the *K* nearest data points are.
- The hyperparameter *K* value is adjusted when fine tuning the KNN model.

An example demonstrating KNN algorithm with two variables x_1 and x_2 and *k* values 3 and 6.



KNN Algorithm Results

KNN Algorithm Accuracy Plot



As shown above, all K values from 1 to 216 were test. The K value with the highest accuracy is 1.

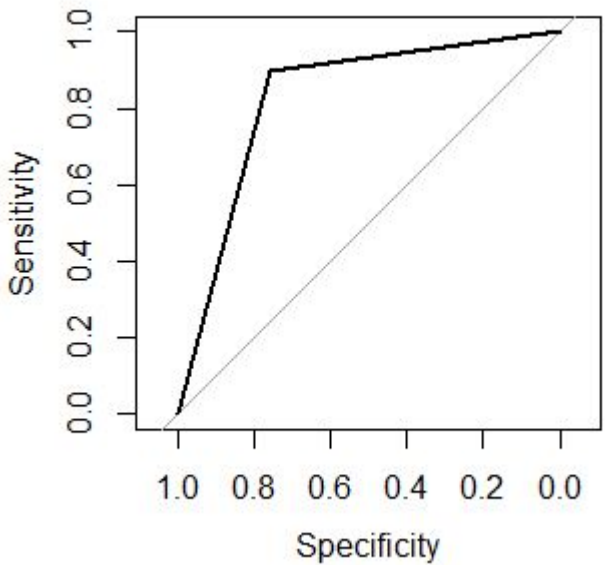
KNN Algorithm Results

Confusion Matrix of KNN Algorithm Test

Actual Condition	Predicted Alzheimer's	Predicted No Alzheimer's	% Correct
Alzheimer's	**50**	**8**	**86.21%**
No Alzheimer's	**16**	**71**	**81.61%**

Overall Accuracy: **83.45%**

ROC Plot of Logistic Regression

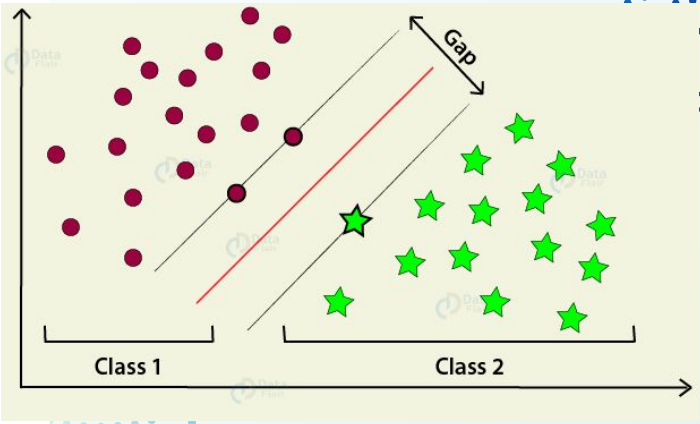


The closer the ROC curve gets to the top-left corner, the better the model.

What is Support Vector Machine (SVM)?

- SVM is a **supervised machine learning algorithm**.
- Like the KNN algorithm, all the **data points are plotted in an *n* dimensional field** where *n* is the number of independent variables
- Classification is performed by finding the hyperplane (an *n*-1 dimensional subspace) that **segregates the classes so that the margin** (distance between the two nearest data points of different classes) **is maximized**
- The hyperparameter ***c* value is used to fine tune** the model. The *c* value refers to the penalty of having data points in the margin. Higher *c* values allow for less data points inside the margin than lower *c* values.

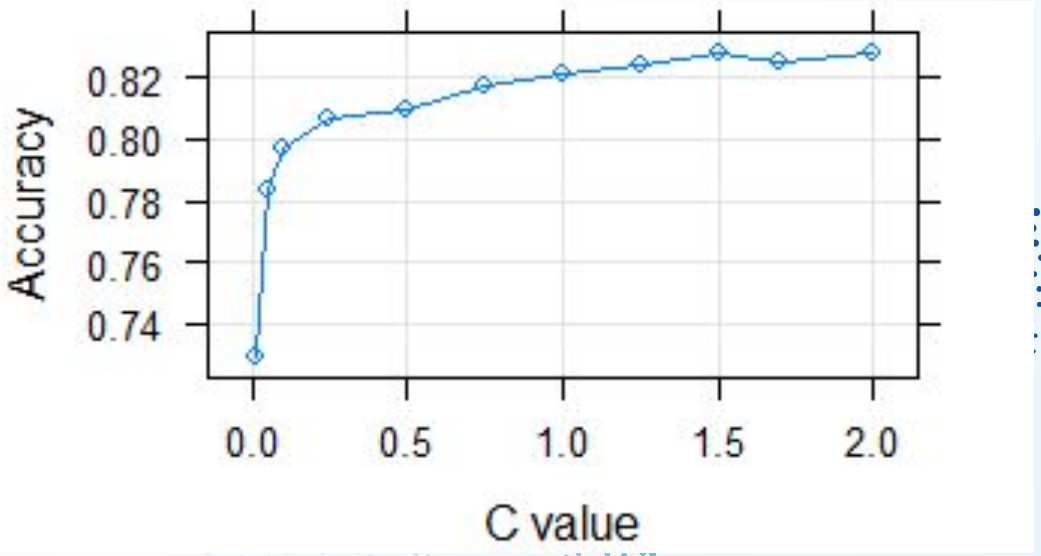
Below is an example of a plot demonstrating the hyperplane as the red line and the margin (gap).



SVM Results

- Multiple SVM models were created and the *c* value was continually adjusted in the fine tuning process.
- *C* values 0, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.7, and 2.0 were used
- The accuracy of each *c* value was evaluated using 10 fold validation
- The optimal *c* value is 1.5

Accuracy of SVM Based on C Value



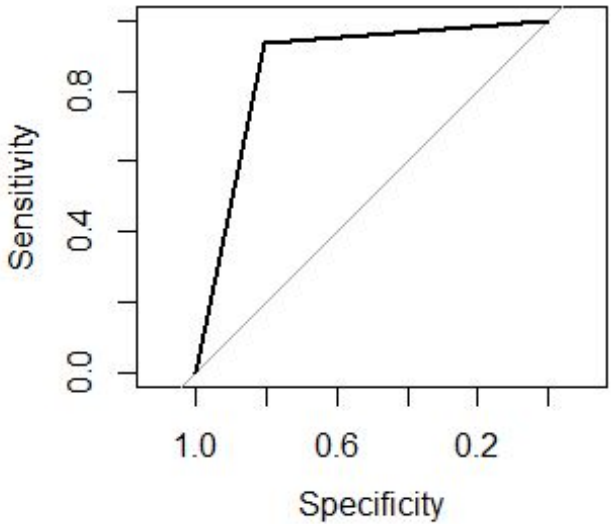
SVM Results

Confusion Matrix of SVM Model Test

Actual Condition	Predicted Alzheimer's	Predicted No Alzheimer's	% Correct
Alzheimer's	**43**	**11**	**80.60%**
No Alzheimer's	**10**	**94**	**93.59%**

Overall Accuracy: **87.59%**

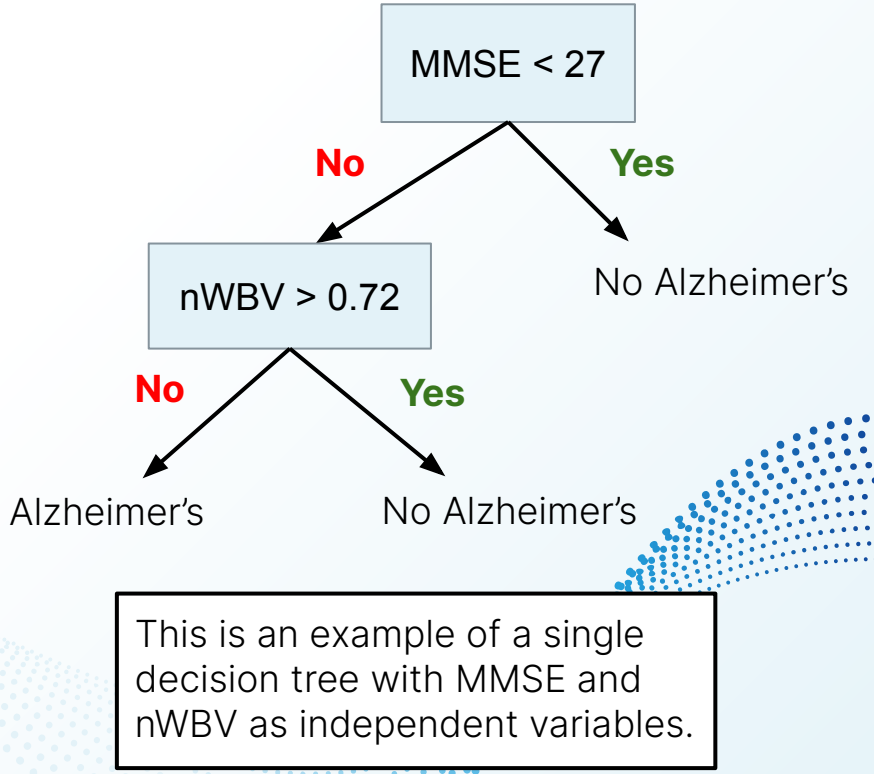
ROC Plot of Support Vector Machine



The closer the ROC curve gets to the top-left corner, the better the model.

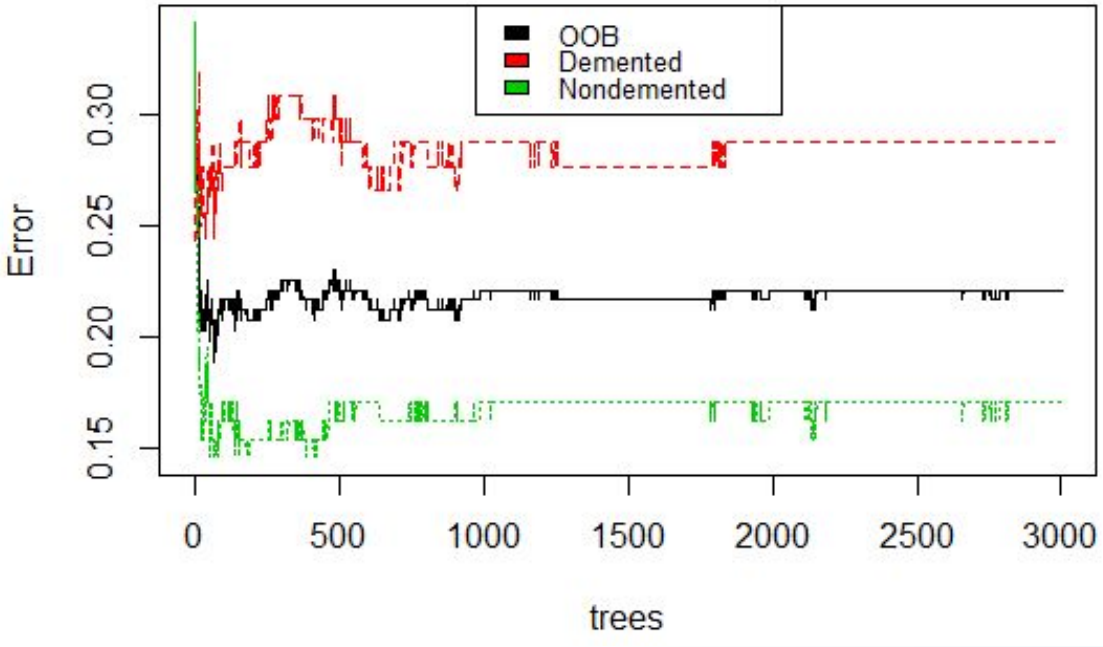
What is Random Forest?

- Random Forest is a **supervised machine learning** algorithm.
- A **decision tree** is a tree-like model of decisions and their possible consequences. Each decision tree outputs its own class prediction.
- A **random forest** contains a large number of decision trees that operate as an ensemble. The most popular class prediction among the decision trees is used as the final prediction for the random forest.
- The **number of trees** in the random forest and **number of independent variables** in each decision tree are used as a hyperparameter for fine tuning the model.
- For each decision tree, a **bootstrapped data set** is made. Then, a decision tree is built based off that bootstrapped data set.

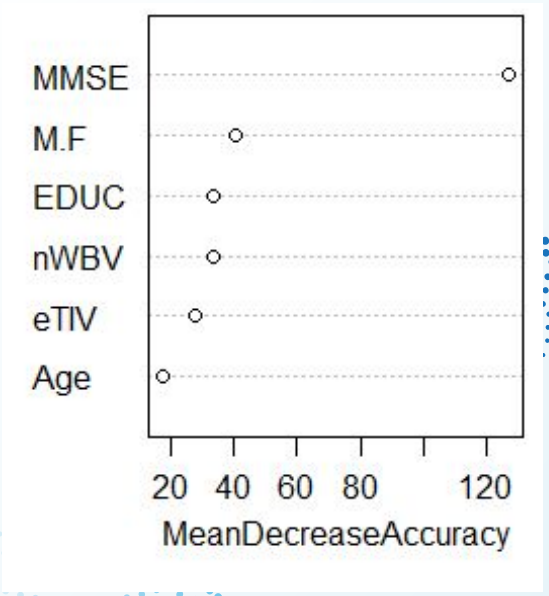


Random Forest Results

OOB and Misclassification Error Based on Number of Trees in Random Forest



Mean Decrease Accuracy of Independent Variables in Random Forest Model



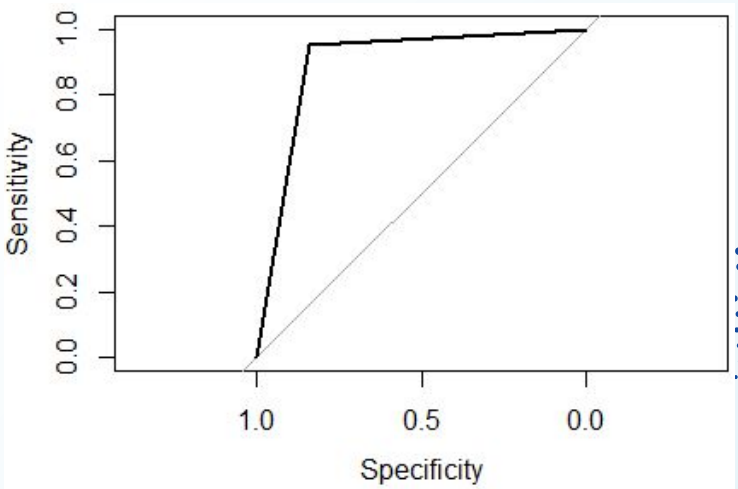
Random Forest Results

Confusion Matrix of Random Forest Test

Actual Condition	Predicted Alzheimer's	Predicted No Alzheimer's	% Correct
Alzheimer's	**54**	**10**	**84.38%**
No Alzheimer's	**4**	**77**	**95.06%**

Overall Accuracy: **90.34%**

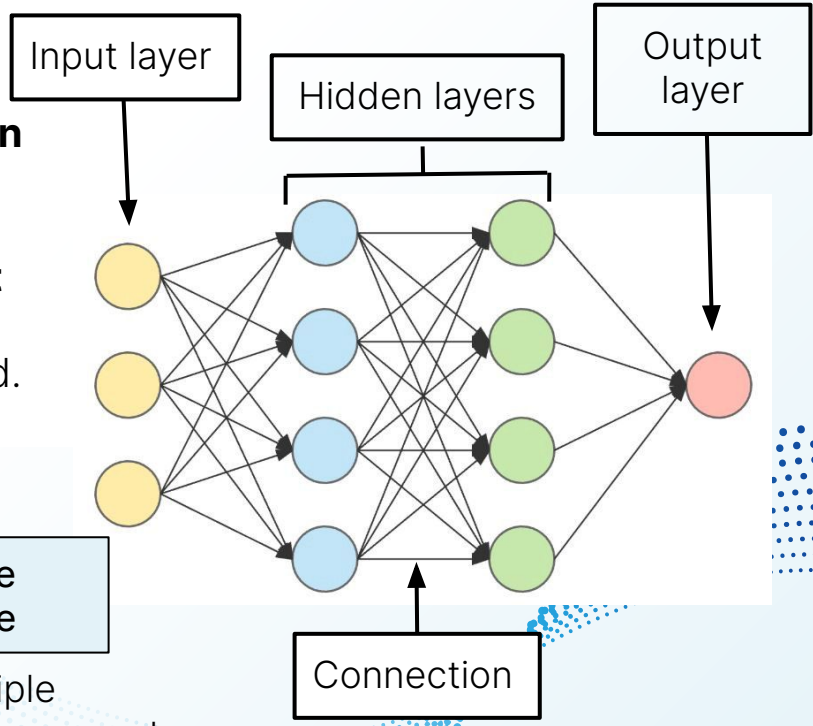
ROC Plot of Random Forest



The closer the ROC curve gets to the top-left corner, the better the model.

What is Feedforward Neural Network?

- Neural network is a machine learning technique that mimics a **biological network of neurons** to **analyze data** and **perform classification**.
- An **input layer** takes in input, the neurons in the **hidden layers** perform some mathematical computation, and the **output layer** node contains the final prediction
- Each connection (edge between neuron) has a **weight** and each neuron has a **bias** and **activation function**, which determines whether the neuron will be activated.



Process of Training:

Forward propagation → Backward propagation → Fine tune

Initially all **weights are assigned random values** and the final output is calculated in the **forward propagation** process.

In **backward propagation**, the weights and biases are modified to **minimize the cost function** for each neuron and create the optimal neural network.

Multiple hyperparameters are modified to **optimize the accuracy** of the neural network and **training is repeated**.

Feedforward Neural Network

Parameters like **type of cost function**, **type of backpropagation algorithm**, **hidden layer layout,** and **type of activation function** were fine tuned in the process.

After fine tuning the model:

- Best cost function: **Cross entropy**
- Best backpropagation algorithm: **Resilient backpropagation**
- Best activation function: **Sigmoid function**
- Best hidden layer layout: **One layer with two hidden nodes**

Feedforward Neural Network

Activation and cost function

To determine whether each neuron is **activated** as well as the **final prediction** of the output neuron:

Z result

Bias of current neuron

$$z = \sum_{i=1}^{m} w_i x_i + b$$

Weights of each connection coming from an activated neuron in the previous layer

The output for each activated neuron in the previous layer.

$$f(z) = \frac{1}{1+e^{-z}}$$

Sigmoid function

Activation function: if f(z) ≥ 0.5, the neuron is activated (if it's the output neuron, the output would be "Alzheimers" if f(z) ≥ 0.5 or "No Alzheimer's" if f(z) < 0.5)

To determine the ***cost*** of each neuron, the **cross entropy** function is used:

$$C = -\frac{1}{n} X[y \ln a + (1-y) \ln(1-a)]$$

Cost value

Actual final output of the neuron

Desired output of the neuron

n is the number of independent variables
X is the sum of all input attributes

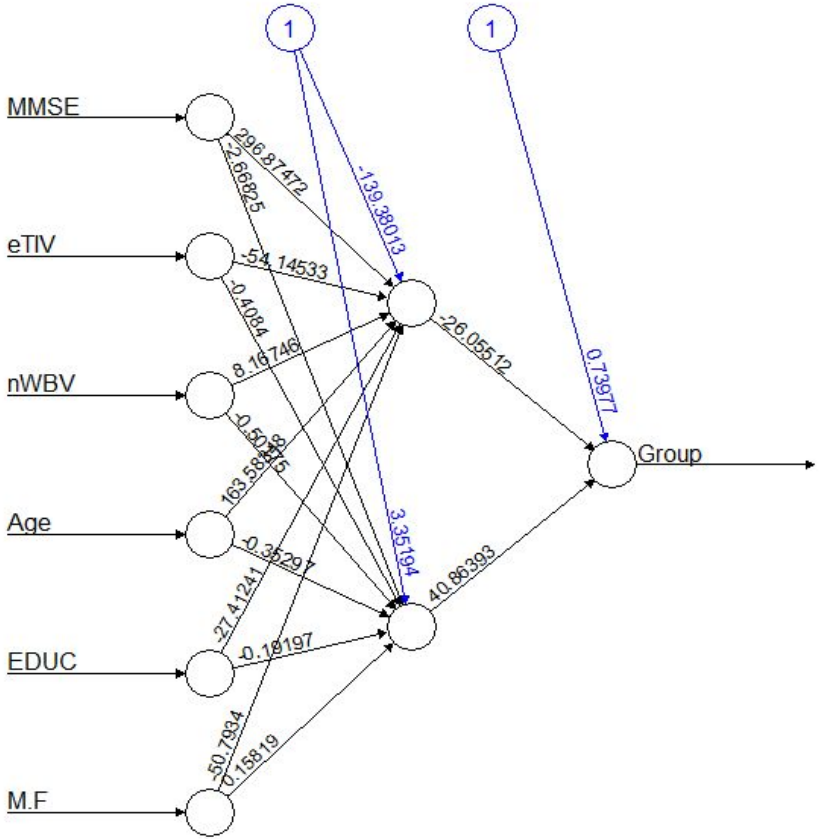
Neural Network Results

Confusion Matrix of Neural Network Test

Actual Condition	Predicted Alzheimer's	Predicted No Alzheimer's	% Correct
Alzheimer's	**52**	**14**	**78.79%**
No Alzheimer's	**6**	**73**	**92.41%**

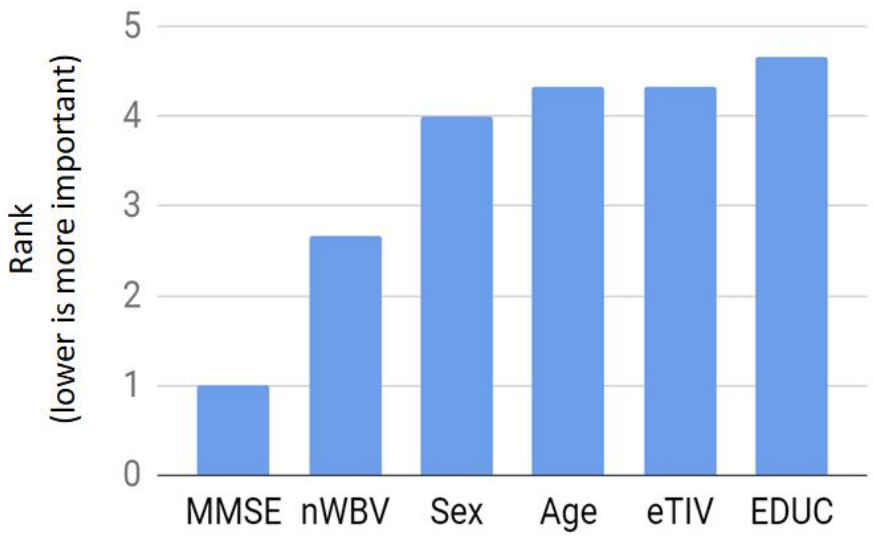
Overall Accuracy: **86.21%**

Plot of Feedforward Neural Network

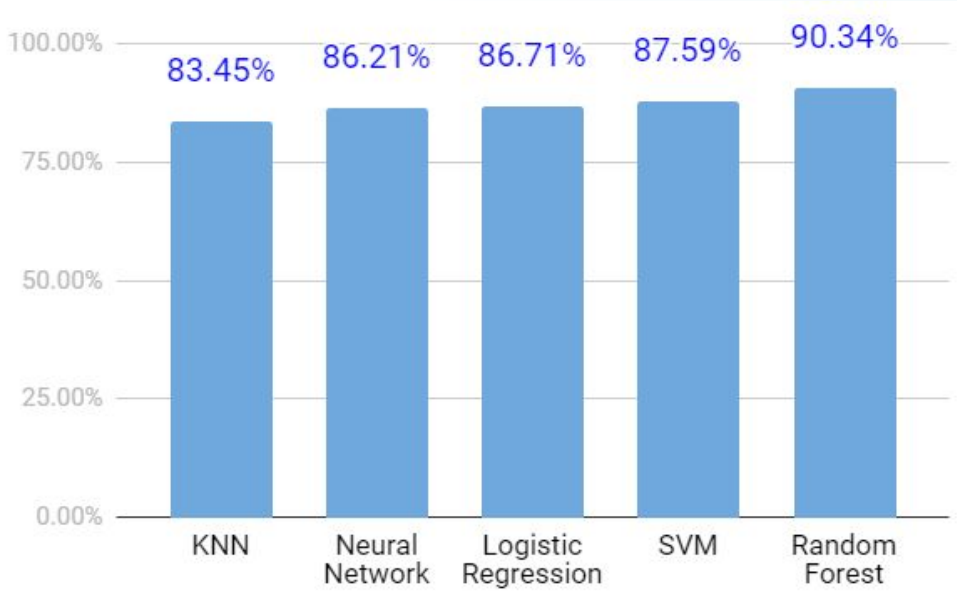


Overall Results

Comparison of Key Risk Factors



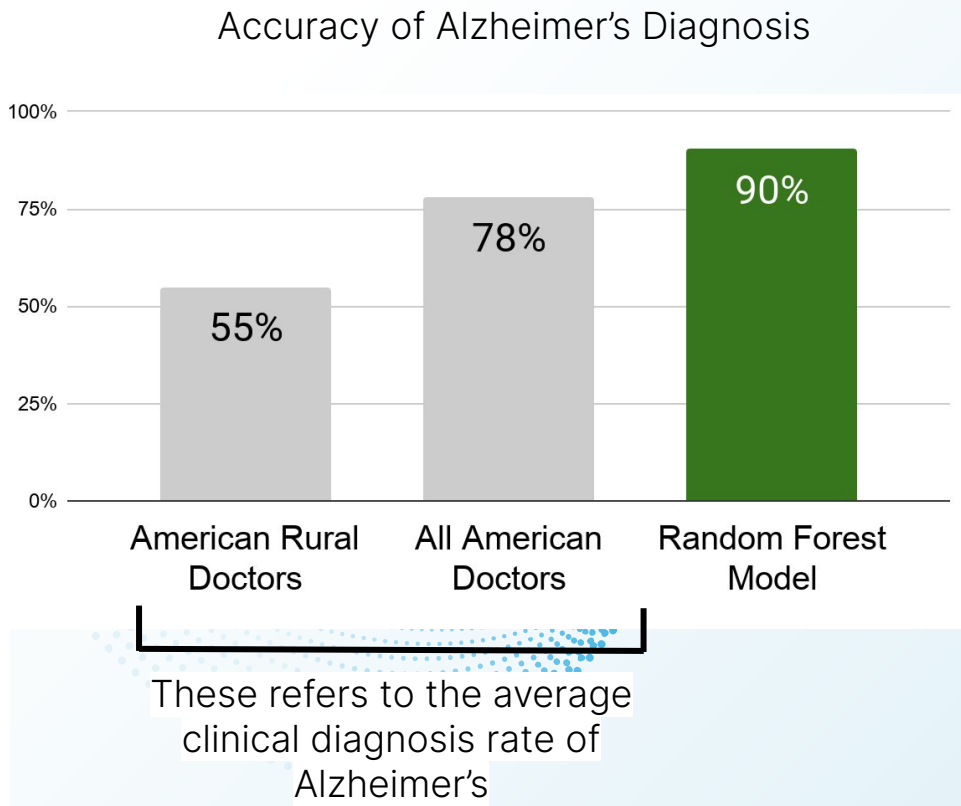
Comparison of Model Accuracies



- **MMSE** is the most important factor in diagnosing Alzheimer's, followed by nWBV and Sex.
- All models had an accuracy between **83% to 91%**.
- **Random forest** had the best accuracy at 90.34%.
- There was **good pairwise concordance**, ranging from **88% to 97%**. The percent in which at least 4 of the 5 models shared the same diagnosis for a testing input was 90.42%.

Conclusion

- **Much better than conventional clinical diagnosis**: The Random Forest model detects Alzheimer's, on average, at an accuracy **35% higher** than American rural doctors and **12% higher** than all American doctors.
- The input needed for the model is **easily obtainable** in a clinical setting, making it **easy and practical** to use in a real setting.
- This improvement in accuracy means that:
  - Patients get **treated earlier**
  - Treatment cost is **lower**
  - Deterioration of Alzheimer's is **slower**
  - Patient's life is **prolonged**.



Future Work

- In the future, I plan to create a web application to make it easy for doctors to use the machine learning models I made to clinically diagnose Alzheimer's

Doctor or nurse inputs patient's demographic and clinical data in user interface.

Model takes in the patient's data and creates a prediction.

The user interface displays the model's prediction and confidence.