Identifying Age-Related Conditions

August 9, 2023

Akhila Ganti
Megan Nguyen
Leslie Nie
Kevin Stallone
Tim Tung

Berkeley
UNIVERSITY OF CALIFORNIA

Introduction

- The goal of our project is to predict if a person has any of three age-related medical conditions. In order to assess, we created models trained on measurements of health characteristics.

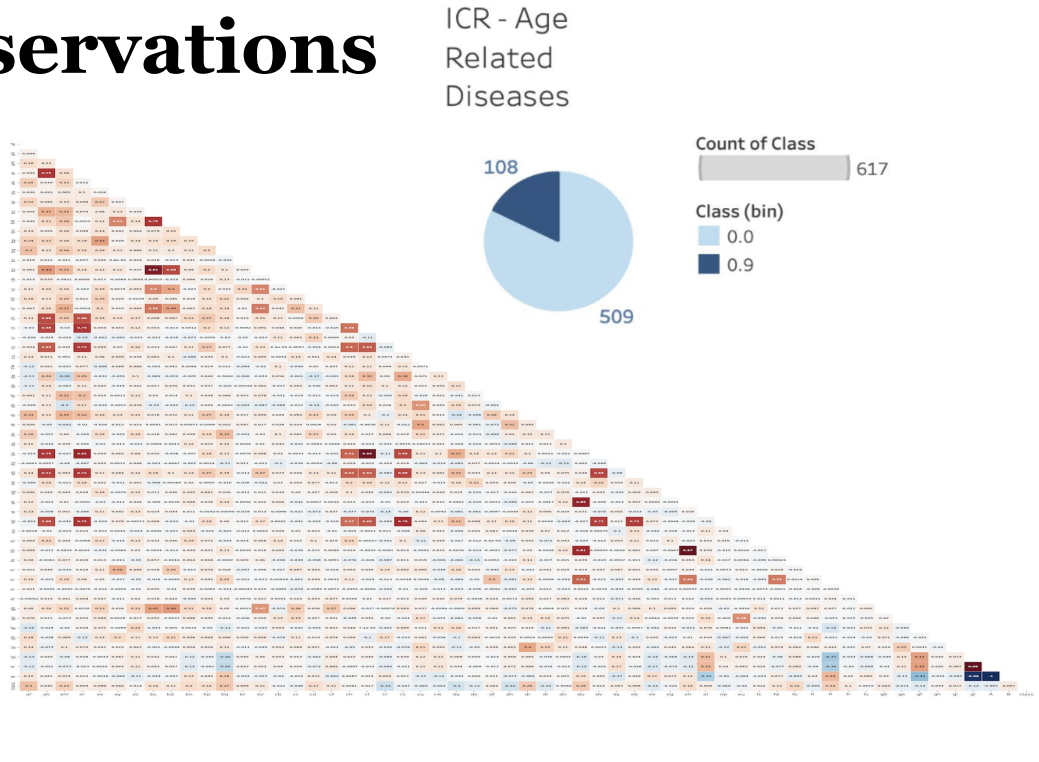
- Aging is a risk factor for numerous diseases and complications. The growing field of bioinformatics includes research into interventions that can help slow and possibly reverse biological aging and prevent major age-related ailments.
    - *Biological Aging Is No Longer an Unsolved Problem*
    - *Machine learning for predicting lifespan-extending chemical compounds*

- Data science has a role to play in developing new methods to solve problems with diverse data, even if the number of samples is relatively small.

- We established a baseline model and then explored decision trees, random forests, logistic regression, and feedforward neural networks for potential performance improvement.

EDA – Initial Observations

- Data is sourced from an active (June 2023) Kaggle competition
- Small dataset (617 observations)
- Imbalanced target class:
  - 82.5% of class 0
  - 17.5% of class 1
- 56 health characteristic features
- Possibly multicollinearity from the correlation matrix
- Feature selection will be considered



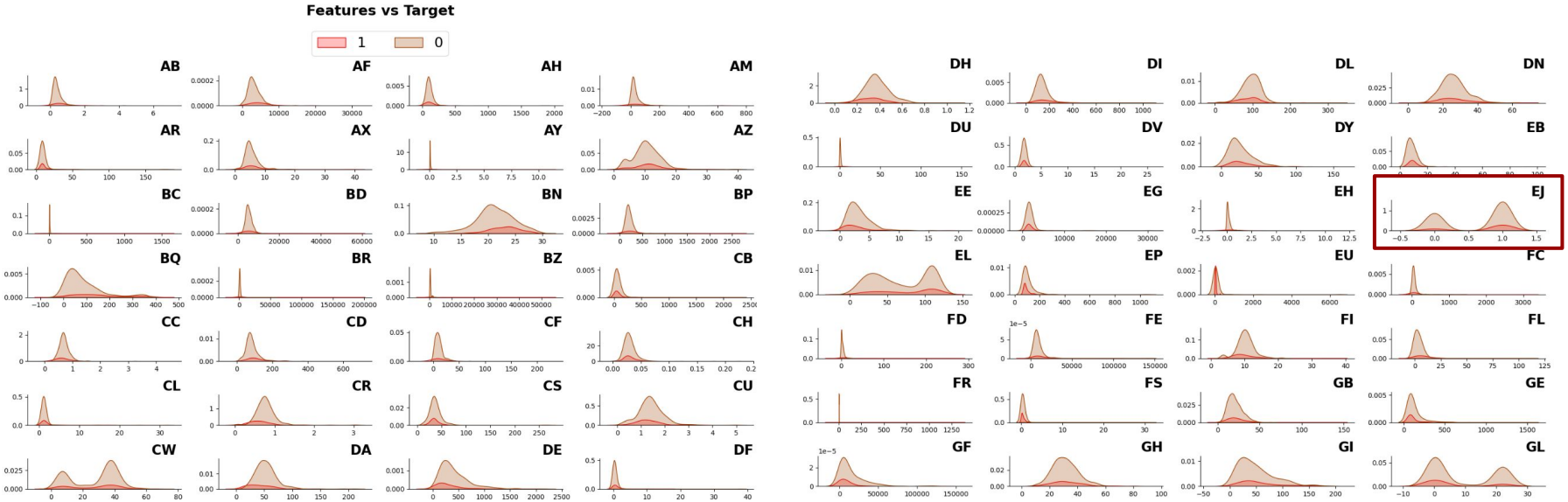
EDA – Summary of Statistics

- 56 anonymized features (AB - GL)
- All features are floats except EJ (which we one-hot encoded)

	data type	#missing	%missing	#unique	min	max	first quartile	second quartile	third quartile
id	object	0	0.000000	617	NaN	NaN	NaN	NaN	NaN
ab	float64	0	0.000000	217	0.081187	6.161666	0.252107	0.354659	0.559763
af	float64	0	0.000000	599	192.59328	28688.18766	2197.34548	3120.31896	4361.63739
ah	float64	0	0.000000	227	85.200147	1910.123198	85.200147	85.200147	113.73954
am	float64	0	0.000000	605	3.177522	630.51823	12.270314	20.53311	39.139886
...	...	...	...	...	...	...	...	...	...
ej	object	0	0.000000	2	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...
gh	float64	0	0.000000	596	9.432735	81.210825	25.034888	30.608946	36.863947
gi	float64	0	0.000000	615	0.897628	191.194764	23.011684	41.007968	67.931664
gl	float64	1	0.162075	355	0.001129	21.978	0.124392	0.337827	21.978
class	int64	0	0.000000	2	0.0	1.0	0.0	0.0	0.0

EDA – Feature Distributions

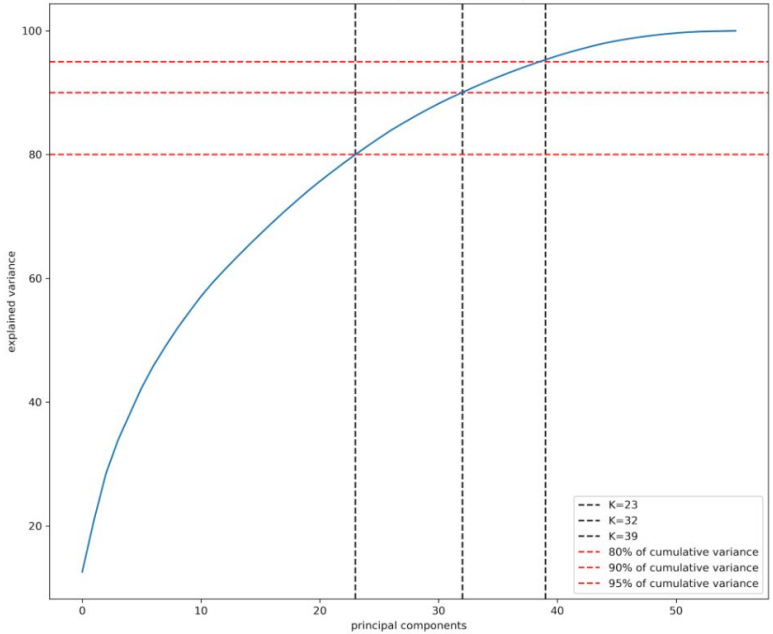
- Disproportion of label 0 and 1 observed across all features

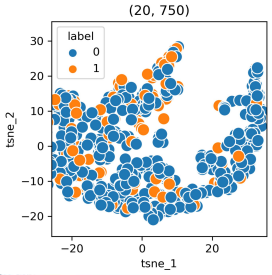


Preprocessing

- Null values:
  - Occur in nine of the features
  - KNNImputer were used to fill the null values
- Separation of data:
  - Data was split into 80/20 for train and test
- Standardization:
  - sklearn.StandardScaler to scale using mean and standard deviation
- Feature reduction
  - Tested PCA and t-SNE
  - Maintains at least 80% of the variance

Explained Variance of PCA





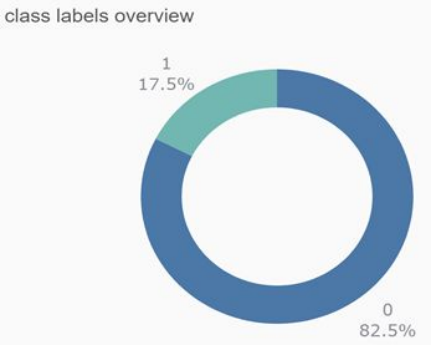
Baseline Model - Majority Vote

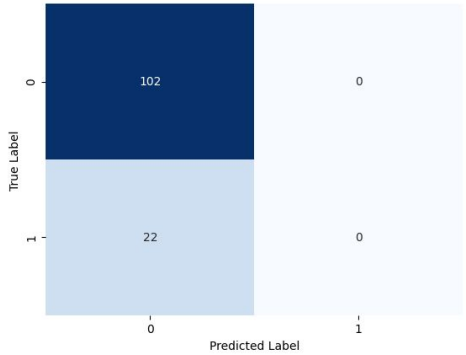
```
from sklearn.dummy import DummyClassifier

dummy_clf = DummyClassifier(strategy="most_frequent")
dummy_clf.fit(X_train, y_train)
print('Majority vote baseline train/test accuracies %.3f/%.3f' %
      (dummy_clf.score(X_train, y_train), dummy_clf.score(X_dev, y_dev)))
```

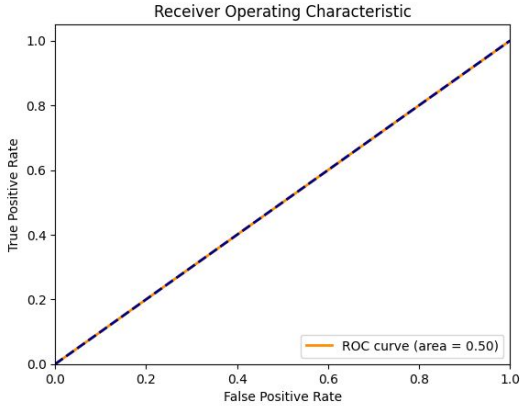
```
Majority vote baseline train/test accuracies 0.826/0.823
```

- Majority voting - selecting the most frequent label
- No learning from features, merely reflecting the distribution of our dataset
- Yields a 82% accuracy due to imbalanced datasets – benchmark for more sophisticated models

```
ROC AUC:  0.5
```







Decision Tree (dt)

model	pca	class_weight	criterion	min_impurity_decrease	max_depth	max_features	accuracy on train	accuracy on dev
dt1	N	Default (None)	Default (gini)	Default (0)	Default (None)	Default (None)	1	.81
dt2	N	Balanced	Default (gini)	Default (0)	Default (None)	Default (None)	1	.88
dt3	N	Balanced	entropy	Default (0)	Default (None)	Default (None)	1	.86
dt4	N	Balanced	entropy	.1	Default (None)	Default (None)	.84	.81
dt5	N	Balanced	Default (gini)	Default (0)	3	Default (None)	.88	.85
dt6	**N**	**Balanced**	**Default (gini)**	**Default (0)**	**Default (None)**	**sqrt**	**1**	**.89**
dt7	Y	Default (None)	Default (gini)	Default (0)	Default (None)	Default (None)	1	.8
dt8	Y	Balanced	Default (gini)	Default (0)	Default (None)	Default (None)	1	.78
dt9	Y	Default (None)	entropy	Default (0)	3	sqrt	.88	.83
dt10	Y	Default (None)	entropy	.1	3	sqrt	.83	.82
dt11	Y	Default (None)	Default (gini)	Default (0)	3	sqrt	.87	.85

- Creates a hierarchical structure that effectively classifies data based on feature decisions
- Optimized accuracy of 89% on the test dataset without PCA transformation
- Improvement over baseline model while presents a tendency towards overfitting

Decision Tree *(cont.)*

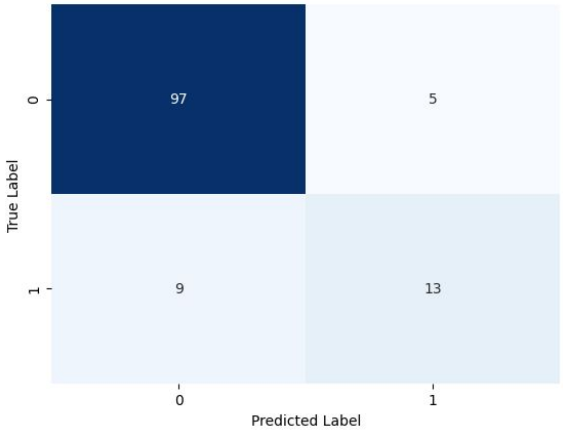
Training Data: 493, 24
Test Data: 124, 24

Best Performer

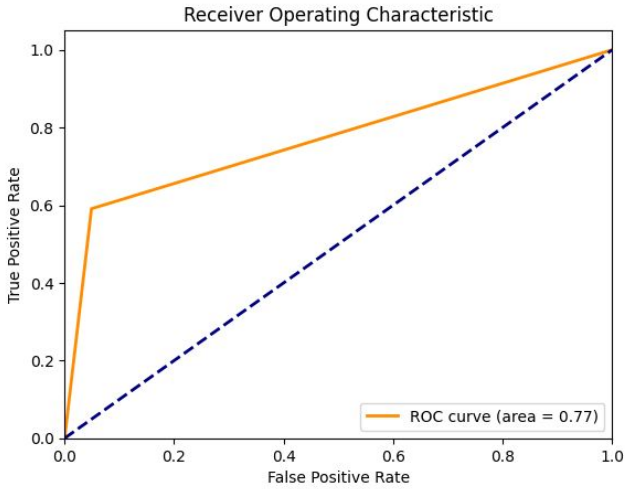
- data before pca transformation
- class_weight = 'balanced'
- criterion = gini
- min_impurity_decrease = 0
- max_depth = None
- max_features = sqrt

Model Evaluation:

```
Train/test accuracies 1.000/0.887
 Precision score: 0.819
 Recall score: 0.771
 F1 score 0.791
```




```
ROC AUC:  0.771
```



Random Forest (rf)/Ensembles

- Significant improvement over baseline model and single decision tree, leveraging multiple learning algorithms
- Random forest exhibits robust resistance to outliers and excels at mitigating overfitting, achieving an accuracy of 93% on test data

model	pca	class_weight	n_estimators	bootstrap	max_depth	max_features	accuracy on train	accuracy on dev
rf1	N	Default (None)	Default (100)	Default (T)	Default (None)	Default (None)	1	.92
rf2	N	Balanced	Default (100)	Default (T)	Default (None)	Default (None)	1	.91
rf3	N	Default (None)	50	Default (T)	Default (None)	Default (None)	1	.89
rf4	N	Default (None)	Default (100)	F	Default (None)	Default (None)	1	.92
rf5	N	Default (None)	Default (100)	Default (T)	10	Default (None)	1	.93
rf6	**N**	**Default (None)**	**Default (100)**	**Default (T)**	**10**	**sqrt**	**1**	**.93**
rf7	Y	Default (None)	Default (100)	Default (T)	Default (None)	Default (None)	1	.88
rf8	Y	Balanced	Default (100)	Default (T)	Default (None)	Default (None)	1	.85
rf9	Y	Default (None)	50	F	Default (None)	Default (None)	1	.88
rf10	Y	Default (None)	50	F	10	Default (None)	.99	.86
rf11	Y	Default (None)	50	F	Default (None)	sqrt	1	.88
Bag using dt6	N	NA	500	NA	NA	NA	1	.90
Bag using dt6	Y	NA	500	NA	NA	NA	1	.84
Adaboost using dt6	N	NA	500	NA	NA	NA	1	.83
Adaboost using dt6	Y	NA	500	NA	NA	NA	1	.84

Random Forest/Ensembles *(cont.)*

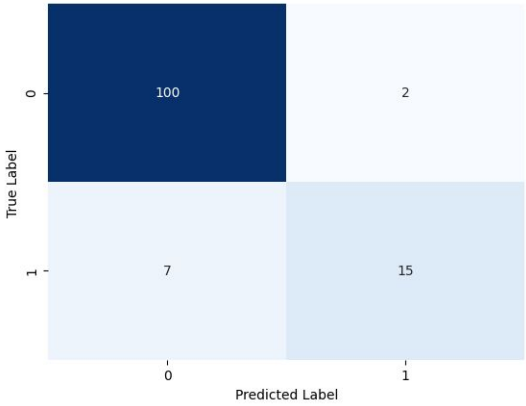
Training Data: 493, 24
Test Data: 124, 24

Best Performer

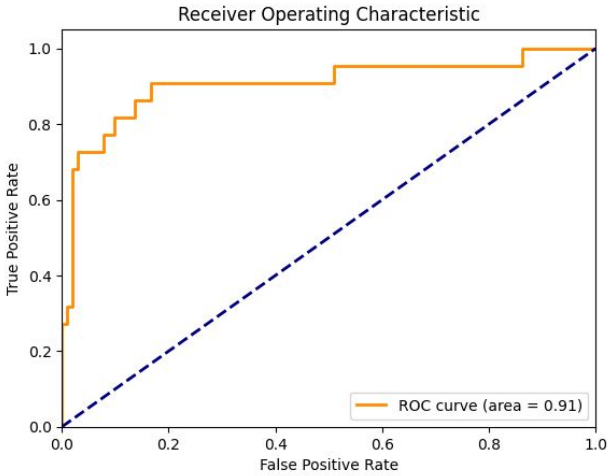
- data before pca transformation
- class_weight = 'balanced'
- n_estimators = 100
- bootstrap = True
- max_depth = 10
- max_features = sqrt

Model Evaluation:

```
Train/test accuracies 1.000/0.927
Precision score: 0.908
Recall score: 0.831
F1 score 0.863
```



ROC AUC: 0.907



Logistic Regression

Model implemented in TensorFlow

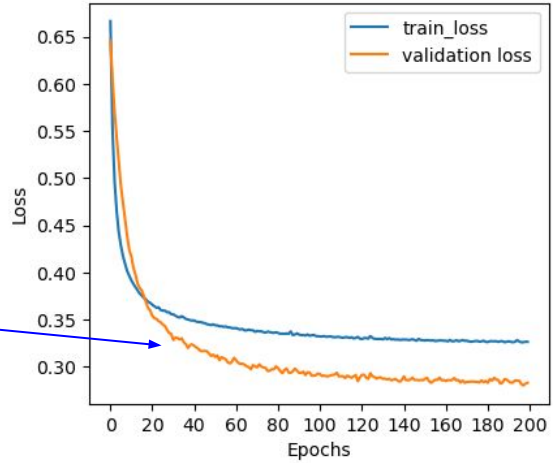
Model Parameters:

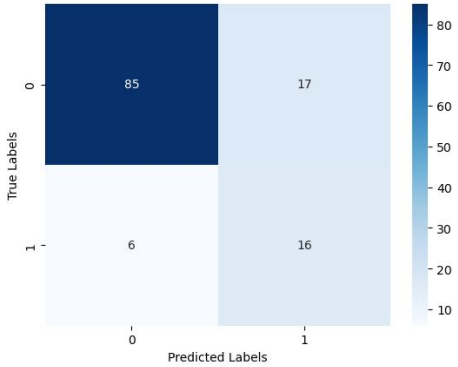
- Activation: relu
- Optimizer: Adam
- Learning rate: 0.01
- Number of Epochs: 200
- Number of K-folds: 5

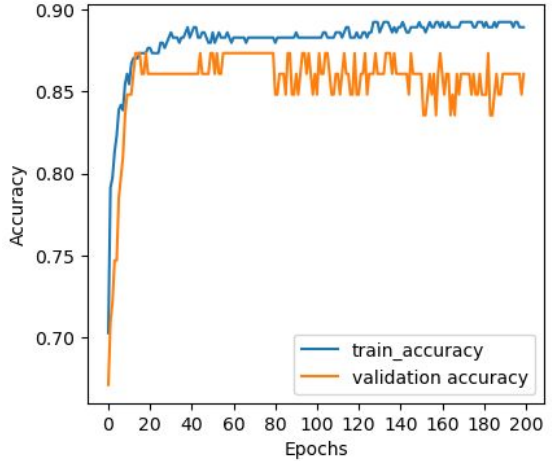
Evaluation Results:

- Binary Accuracy: 0.8387
- Precision: 0.6909
- Recall: 0.5389
- F1 Score: 0.3014

Validation loss curve is below the training loss curve, possibly indicative of underfitting → Remedy by increasing model capacity.







Feedforward Neural Network (FFNN)

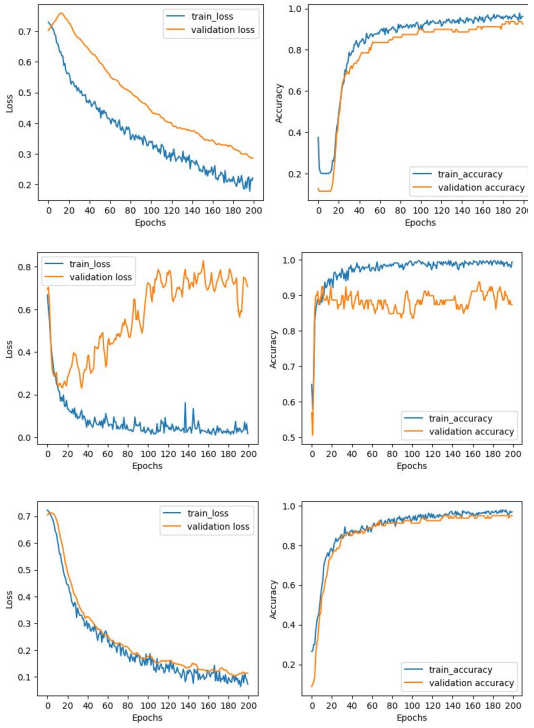
Hyperparameters used for model tuning:

- # of hidden layers
- # of units per hidden layer
- Activation (relu, tanh)
- Learning Rate
- Dropout Rate
- Number of Epochs

Addressed class imbalances by using class weights.

Observations from hand tuning:

- Lower learning rate with more training epochs led to smoother learning curves than higher learning rates with fewer epochs
- Increasing the dropout rate seemed to produce better results than reducing model complexity



FFNN: Keras Tuner HP Optimization

Keras Tuner provides an automated method for hyperparameter optimization.

```
Search space summary
Default search space size: 8
num_layers (Int)
{'default': None, 'conditions': [], 'min_value': 1, 'max_value': 3, 'step': 1, 'sampling': 'linear'}
units_0 (Int)
{'default': None, 'conditions': [], 'min_value': 5, 'max_value': 100, 'step': 5, 'sampling': 'linear'}
activation (Choice)
{'default': 'relu', 'conditions': [], 'values': ['relu', 'tanh'], 'ordered': False}
dropout (Boolean)
{'default': False, 'conditions': []}
lr (Float)
{'default': 0.0001, 'conditions': [], 'min_value': 0.0001, 'max_value': 0.01, 'step': None, 'sampling': 'log'}
units_1 (Int)
{'default': None, 'conditions': [], 'min_value': 5, 'max_value': 100, 'step': 5, 'sampling': 'linear'}
units_2 (Int)
{'default': None, 'conditions': [], 'min_value': 5, 'max_value': 100, 'step': 5, 'sampling': 'linear'}
dropout_rate (Int)
{'default': None, 'conditions': [], 'min_value': 20, 'max_value': 50, 'step': 5, 'sampling': 'linear'}
```

```
Trial 153 Complete [00h 00m 27s]
val_binary_accuracy: 0.9220430056254069

Best val_binary_accuracy So Far: 0.9301075339317322
Total elapsed time: 01h 04m 44s

Search: Running Trial #154
```

Value	Best Value So Far	Hyperparameter
3	2	num_layers
90	95	units_0
tanh	tanh	activation
True	True	dropout
0.0037935	0.0061472	lr
95	40	units_1
80	40	units_2
45	25	dropout_rate

```
Results summary
Results in my_dir\ICR
Showing 10 best trials
Objective(name="val_binary_accuracy", direction="max")

Trial 179 summary
Hyperparameters:
num_layers: 2
units_0: 100
activation: tanh
dropout: True
lr: 0.0012440718000666692
units_1: 50
units_2: 35
dropout_rate: 25
Score: 0.9327957034111023
```

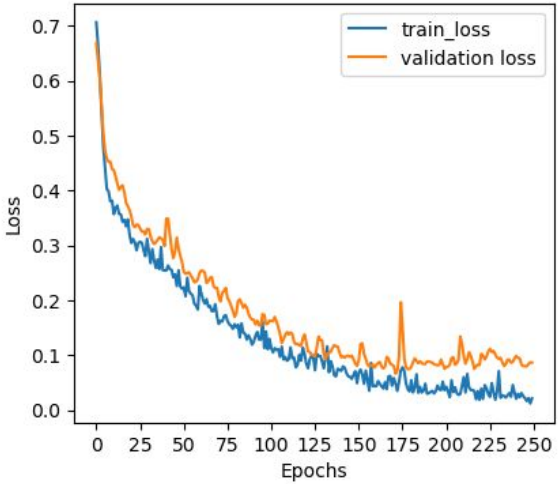
FFNN: Model Summary

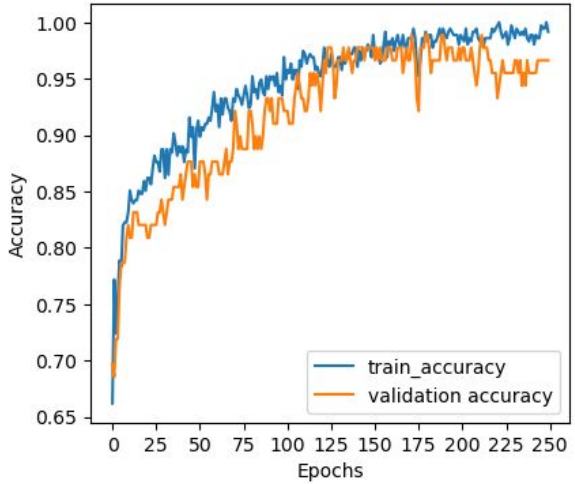
Best Parameters

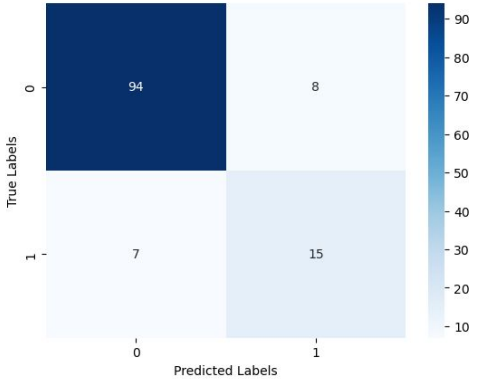
- Learning rate = 0.001244
- Activation = tanh
- optimizer = Adam
- hidden layers = [100, 50]
- dropout rate = 0.25
- epochs = 250

Model Evaluation:

- Binary Accuracy: 0.8806
- Precision: 0.6091
- Recall: 0.6895
- F1 Score: 0.3014







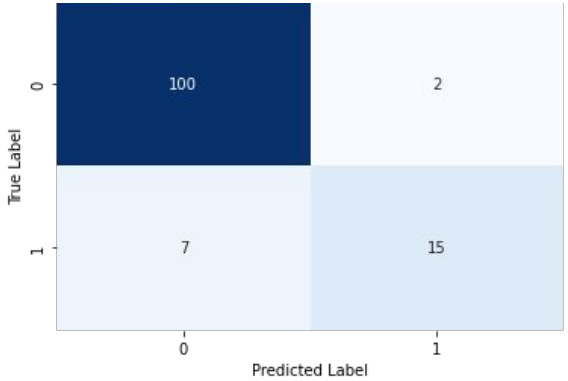
Feed Forward Neural Network: Ablation Table

Run	Activation	Hidden Layer Size	Learning rate	Dropout rate	Loss, Accuracy
1	relu	95, 10	0.0090591	0.1	[1.153681, 0.83871]
2	relu	95, 10	0.0090591	0.25	[1.99791, 0.87097]
3	tanh	95, 10	0.0090591	0.1	[0.55234, 0.87097]
4	**tanh**	**95, 10**	**0.0090591**	**0.2**	**[0.44030, 0.90323]**
5	tanh	95, 10	0.0090591	0.25	[0.456542, 0.88709]
6	tanh	85, 10	0.0090591	0.25	[0.54294, 0.87903]
7	tanh	100,50	0.0061472	0.25	[0.62982, 0.89516]

- FFNN model that classifies data based on pca reduced dimensionality features
- Accuracy data for multiple variations of hyperparameter tuning
- Optimized accuracy of 90% on the test dataset but not consistent

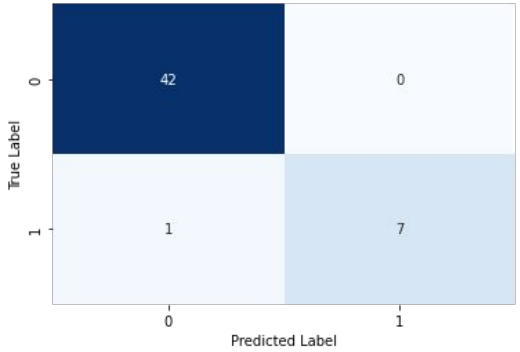
Fairness - Model is Unfair

Confusion Matrix for Population Model



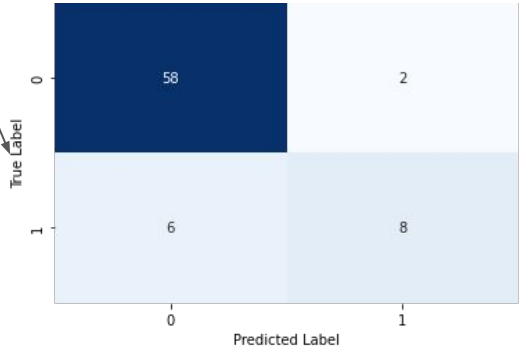
Precision score: 0.882
Recall score: 0.682
F1 score: 0.769
Accuracy: 0.927

Confusion Matrix for **Subgroup A** Model (EJ Split)



Precision score: 1.000
Recall score: 0.875
F1 score: 0.933
Accuracy: 0.98

Confusion Matrix for **Subgroup B** Model (EJ Split)



Precision score: 0.800
Recall score: 0.571
F1 score: 0.667
Accuracy: 0.891

Potential thoughts of what feature 'EJ' could be:

- **Gender**
- **Age** (Under 50 and over 50)
- **Level of Education** (Have an education and doesn't have an education)
- **Geographic Location**

Berkeley UNIVERSITY OF CALIFORNIA

Conclusion/Future Work

Random Forest is one of the powerful models that can be used to predict a medical condition using a person's health characteristics due to an imbalance within the dataset and a limited number of observations.

Future work:

- **Feature Engineering**: Develop additional features (if not anonymized)
- **Combine Data Types**: Integrate genetic, clinical, and lifestyle data for better predictions.
- **Temporal Analysis**: Use longitudinal data to track changes and predict age-related conditions.
- **Adaptive Modeling**: Design models that update over time for changing demographics.
- **Validation**: Thoroughly validate models for broader applicability.
- **Ethical Considerations**: Address biases with fairness techniques.
- **Improve Generalization**: A larger dataset aids better pattern recognition. Collect more data.

Questions

Contributions

Akhila Ganti: Introduction, EDA, Fairness, FFNN Ablation Table
Megan Nguyen: EDA, Fairness Analysis, Random Forests, Conclusion
Leslie Nie: EDA, Decision Tree, Random Forests
Kevin Stallone: EDA, Preprocessing, Coding and Research
Tim Tung: EDA, Preprocessing, Logistic Regression/FFNN