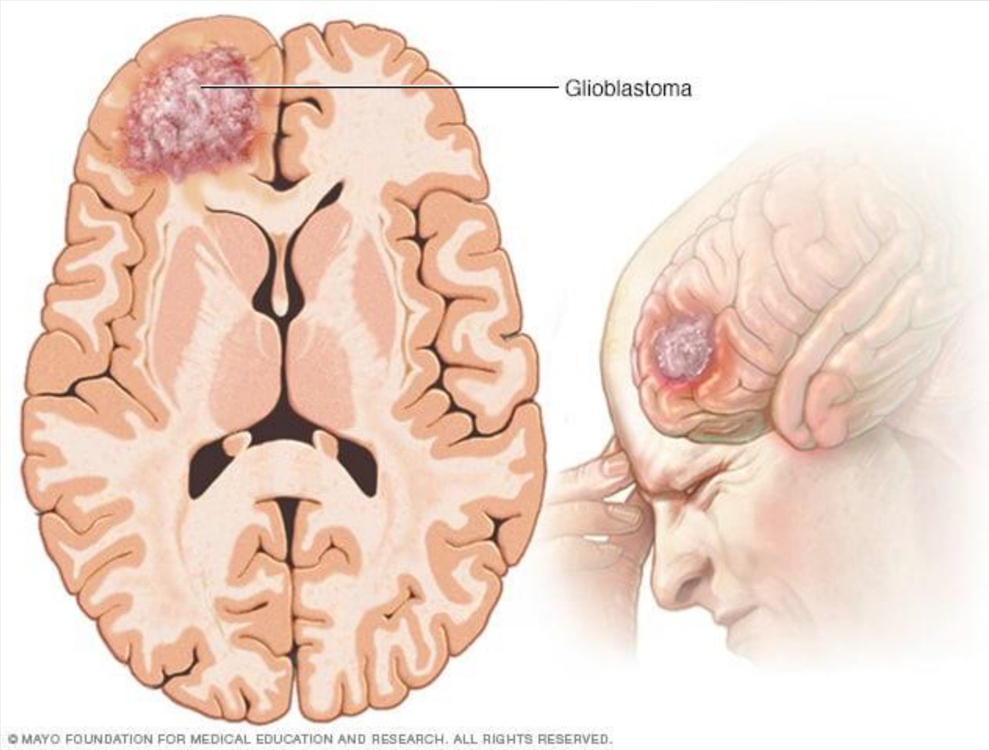
PROJECT PRESENTATION

PAKORN SAGULKOO 6581030520

PHD CANDIDATE IN BIOINFORMATICS AND COMPUTATIONAL BIOLOGY, CU

INTRODUCTION

- Brain glioma
  - One of the most common brain tumor
  - Found in both adults and children
  - Several subtypes
  - Tumor grading is essential for disease management
  - Therefore, grading classification using machine learning (ML) models could provide some medical benefits.

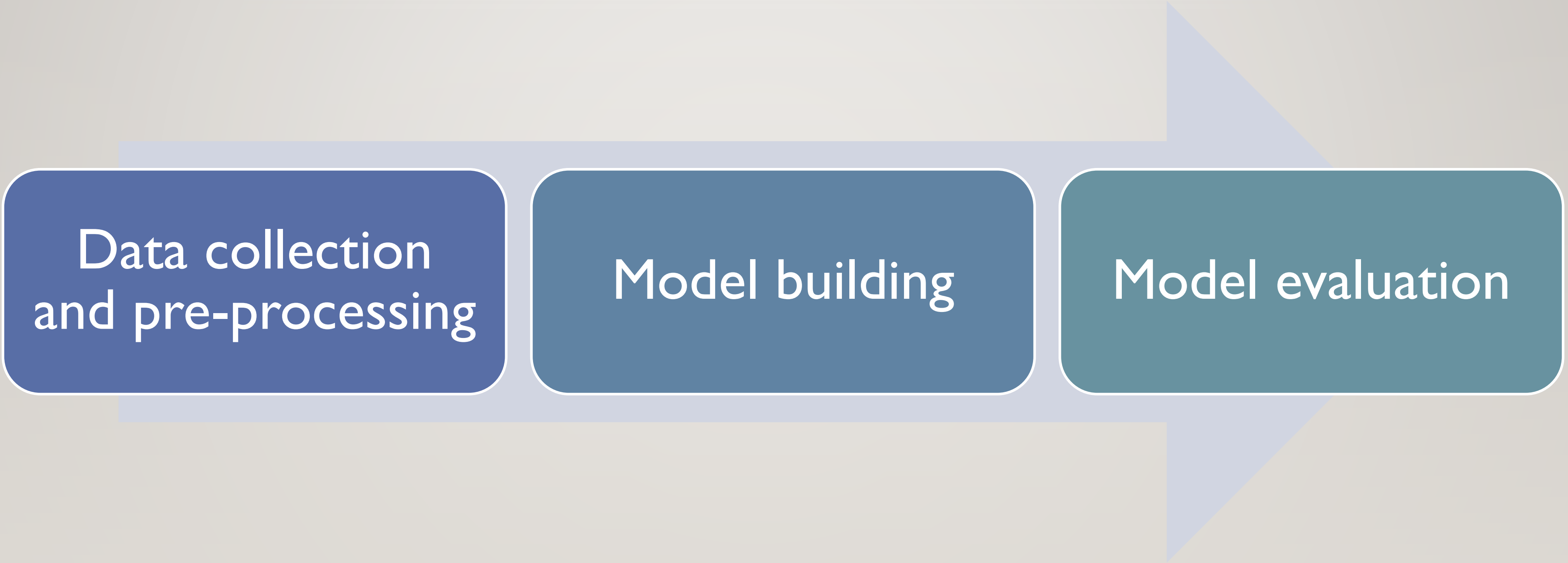


© MAYO FOUNDATION FOR MEDICAL EDUCATION AND RESEARCH. ALL RIGHTS RESERVED.

RESEARCH OBJECTIVES

- This project aimed to
  - Apply ML algorithms to classify brain glioma grading based on clinical and genetic data of glioma patients
  - Measure the model performance in each ML algorithm

MATERIALS AND METHODS



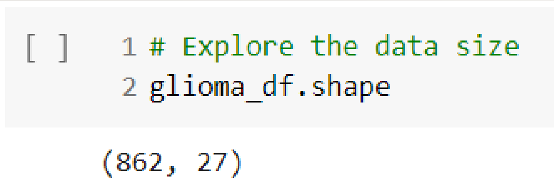
Data collection and pre-processing

Model building

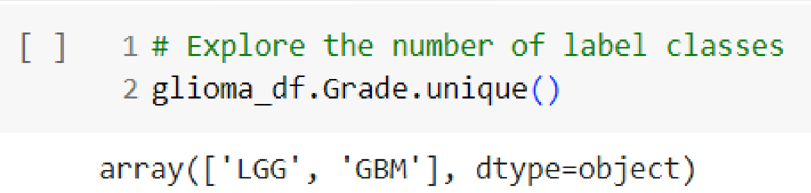
Model evaluation

MATERIALS AND METHODS

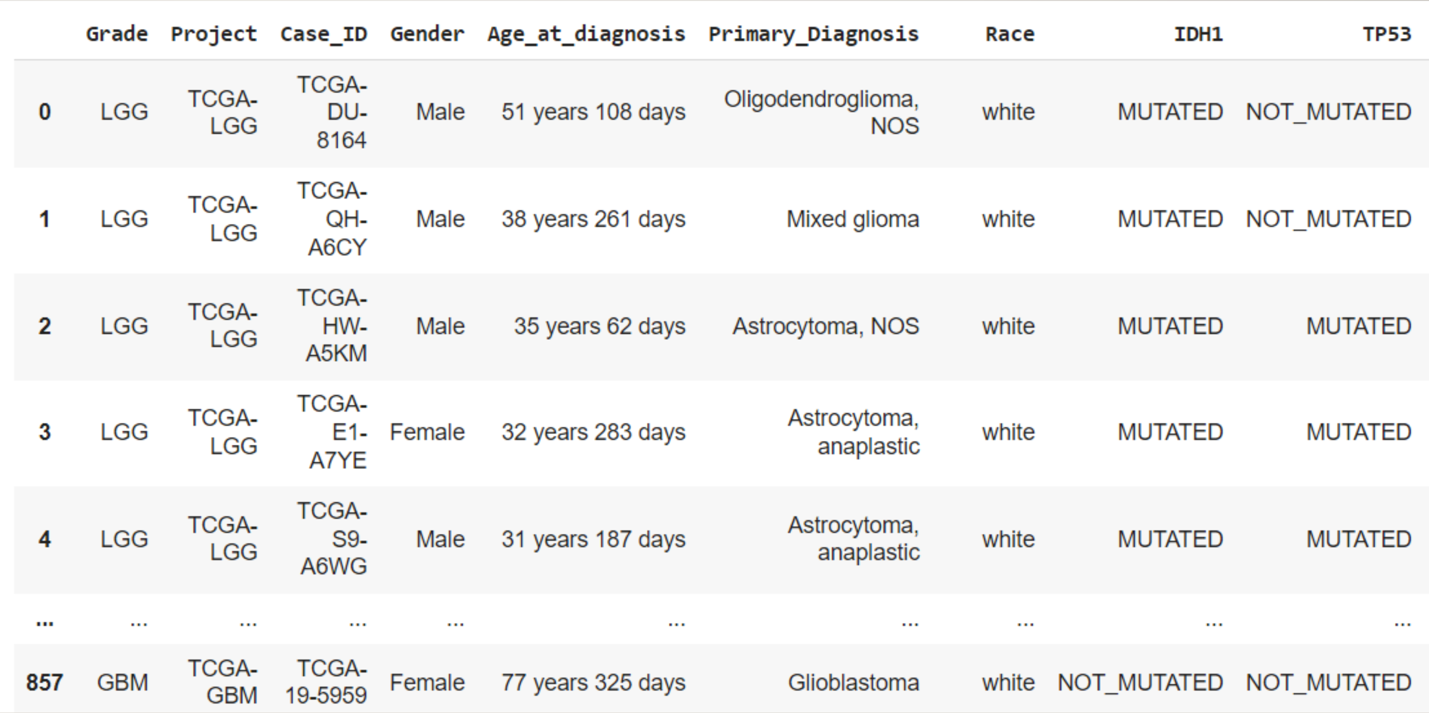
- Data collection and pre-processing
  - Original data frame



```
[ ]  # Explore the data size
     glioma_df.shape

(862, 27)
```



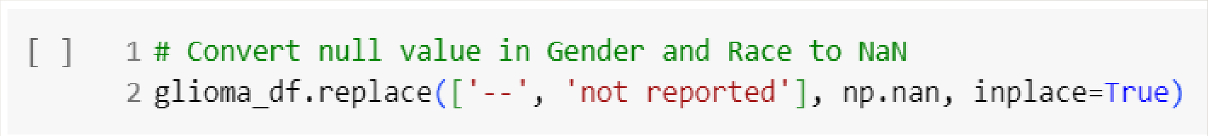
```
[ ]  # Explore the number of label classes
     glioma_df.Grade.unique()

array(['LGG', 'GBM'], dtype=object)
```



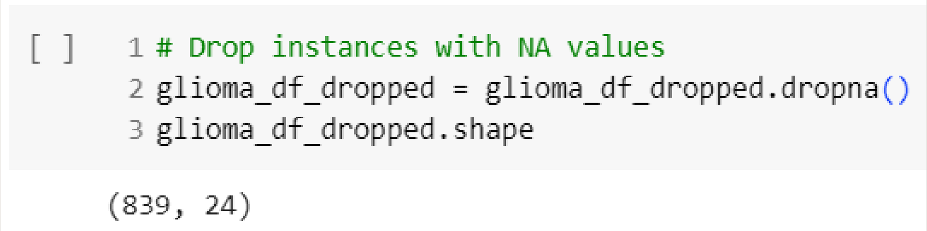
	Grade	Project	Case_ID	Gender	Age_at_diagnosis	Primary_Diagnosis	Race	IDH1	TP53
0	LGG	TCGA-LGG	TCGA-DU-8164	Male	51 years 108 days	Oligodendroglioma, NOS	white	MUTATED	NOT_MUTATED
1	LGG	TCGA-LGG	TCGA-QH-A6CY	Male	38 years 261 days	Mixed glioma	white	MUTATED	NOT_MUTATED
2	LGG	TCGA-LGG	TCGA-HW-A5KM	Male	35 years 62 days	Astrocytoma, NOS	white	MUTATED	MUTATED
3	LGG	TCGA-LGG	TCGA-E1-A7YE	Female	32 years 283 days	Astrocytoma, anaplastic	white	MUTATED	MUTATED
4	LGG	TCGA-LGG	TCGA-S9-A6WG	Male	31 years 187 days	Astrocytoma, anaplastic	white	MUTATED	MUTATED
...	...	...	...	...	...	...	...	...	...
857	GBM	TCGA-GBM	TCGA-19-5959	Female	77 years 325 days	Glioblastoma	white	NOT_MUTATED	NOT_MUTATED

MATERIALS AND METHODS

- Data collection and pre-processing
  - Data was collected from UCI Machine Learning Repository.
  - Extract several features such as sex, age at diagnosis, race, and gene mutation information
  - To handle with missing value
    - Drop instances having NA



```
# Convert null value in Gender and Race to NaN
glioma_df.replace(['--', 'not reported'], np.nan, inplace=True)
```

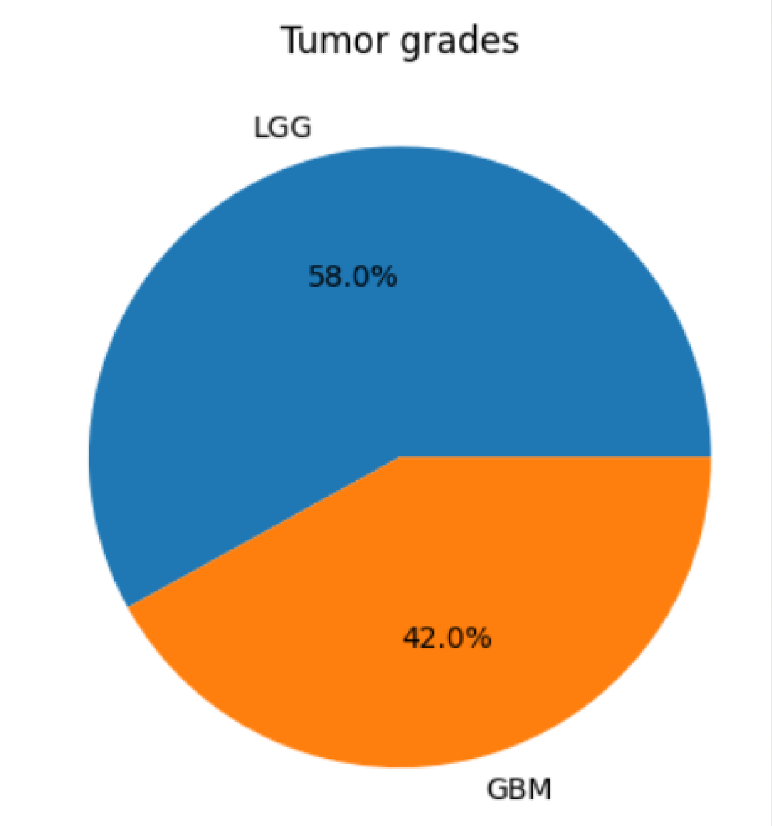


```
# Drop instances with NA values
glioma_df_dropped = glioma_df_dropped.dropna()
glioma_df_dropped.shape
```

```
(839, 24)
```

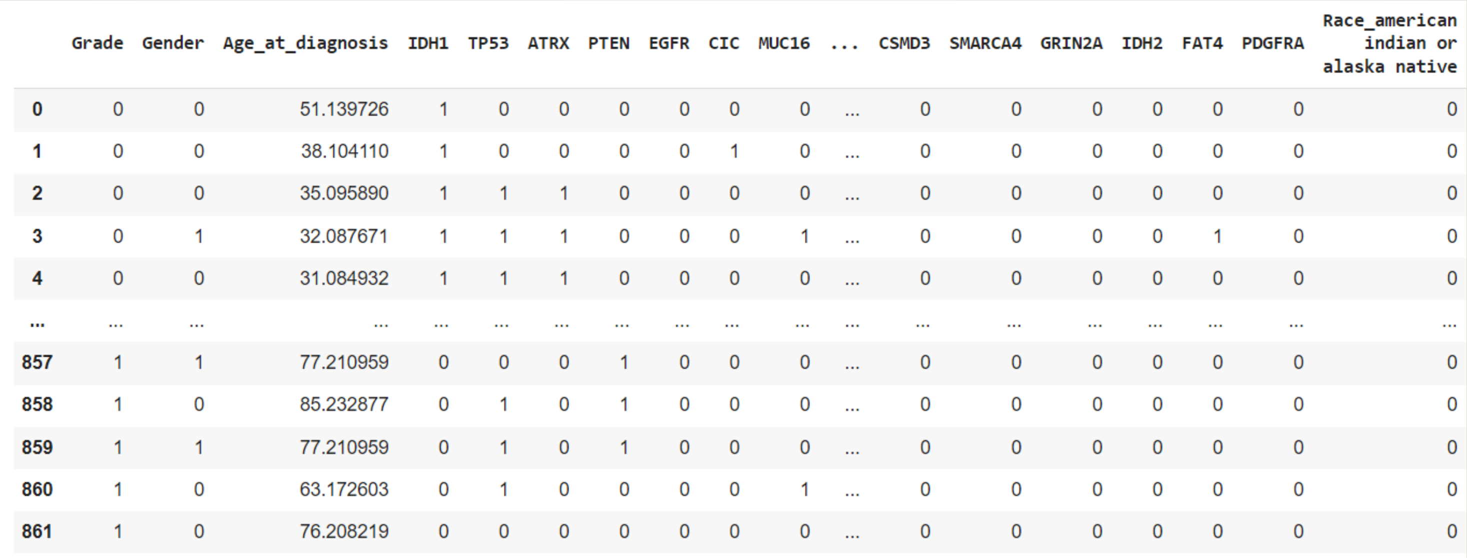
MATERIALS AND METHODS

- Data collection and pre-processing
  - Imbalance data
    - Mild imbalance: no further correction
    - The minority proportion > 0.4



MATERIALS AND METHODS

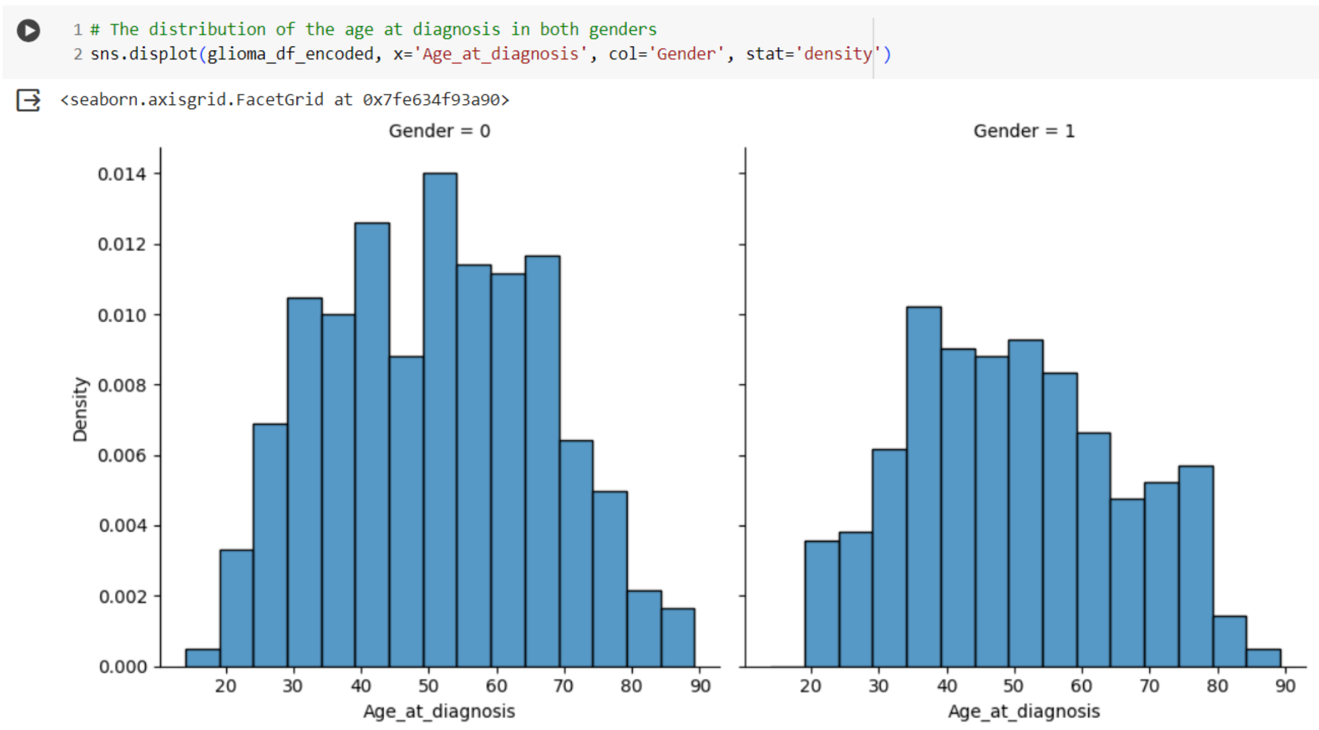
- Data collection and pre-processing
  - Convert age at diagnosis to numerical value
  - One-hot encoding of gene mutation information



	Grade	Gender	Age_at_diagnosis	IDH1	TP53	ATRX	PTEN	EGFR	CIC	MUC16	...	CSMD3	SMARCA4	GRIN2A	IDH2	FAT4	PDGFRA	Race_american indian or alaska native
0	0	0	51.139726	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0
1	0	0	38.104110	1	0	0	0	0	1	0	...	0	0	0	0	0	0	0
2	0	0	35.095890	1	1	1	0	0	0	0	...	0	0	0	0	0	0	0
3	0	1	32.087671	1	1	1	0	0	0	1	...	0	0	0	0	1	0	0
4	0	0	31.084932	1	1	1	0	0	0	0	...	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
857	1	1	77.210959	0	0	0	1	0	0	0	...	0	0	0	0	0	0	0
858	1	0	85.232877	0	1	0	1	0	0	0	...	0	0	0	0	0	0	0
859	1	1	77.210959	0	1	0	1	0	0	0	...	0	0	0	0	0	0	0
860	1	0	63.172603	0	1	0	0	0	0	1	...	0	0	0	0	0	0	0
861	1	0	76.208219	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0

MATERIALS AND METHODS

- Data collection and pre-processing
  - Exploring some statistics

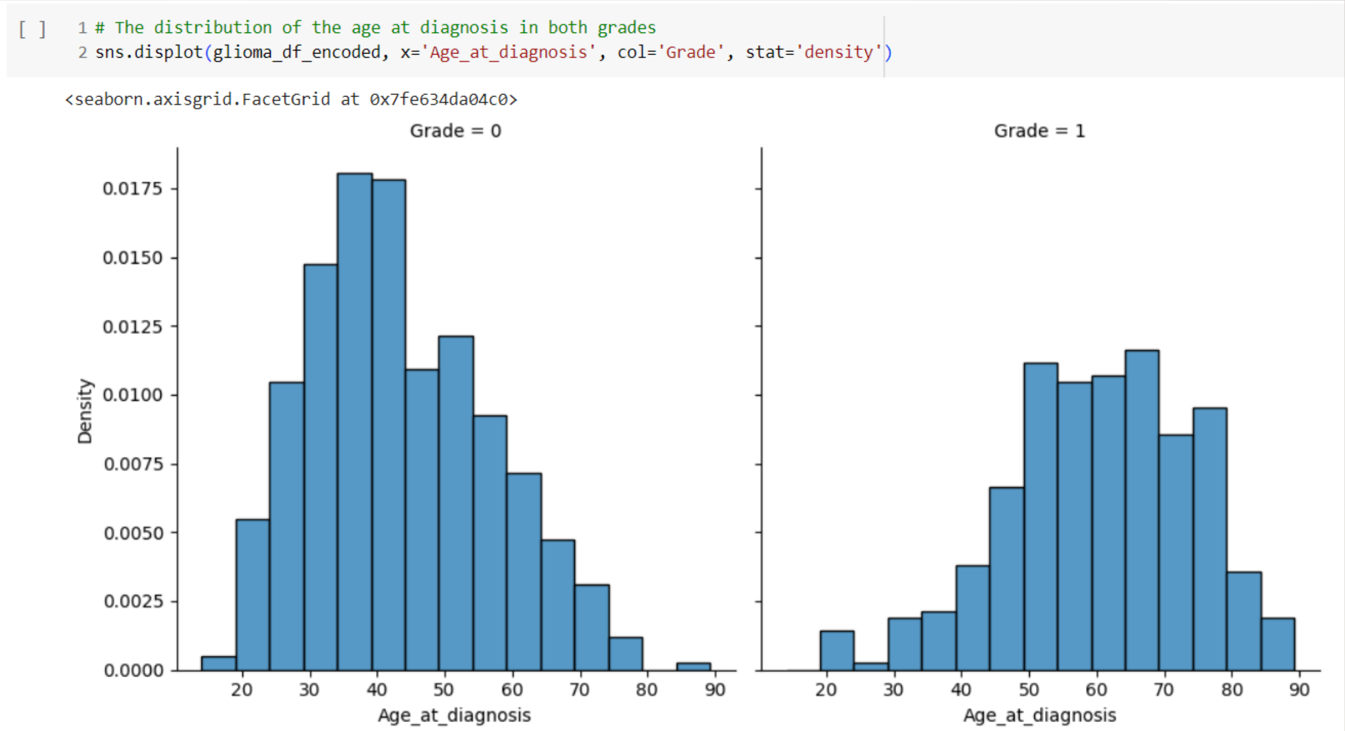


```
# The distribution of the age at diagnosis in both genders
sns.displot(glioma_df_encoded, x='Age_at_diagnosis', col='Gender', stat='density')
```

```
<seaborn.axisgrid.FacetGrid at 0x7fe634f93a90>
```

MATERIALS AND METHODS

- Data collection and pre-processing
  - Exploring some statistics

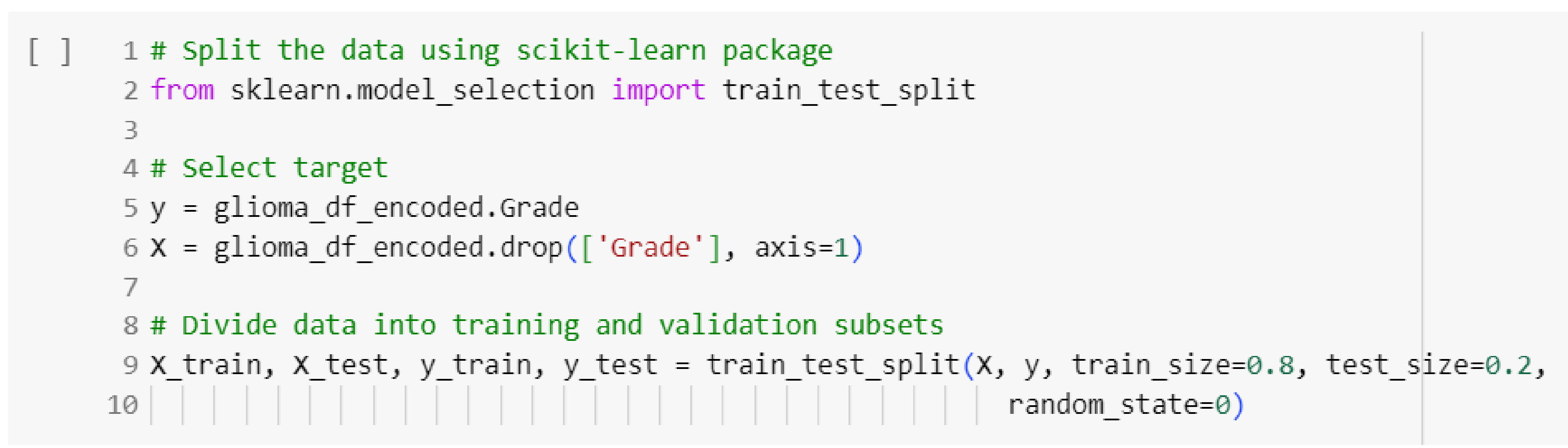


```
# The distribution of the age at diagnosis in both grades
sns.displot(glioma_df_encoded, x='Age_at_diagnosis', col='Grade', stat='density')
```

```
<seaborn.axisgrid.FacetGrid at 0x7fe634da04c0>
```

MATERIALS AND METHODS

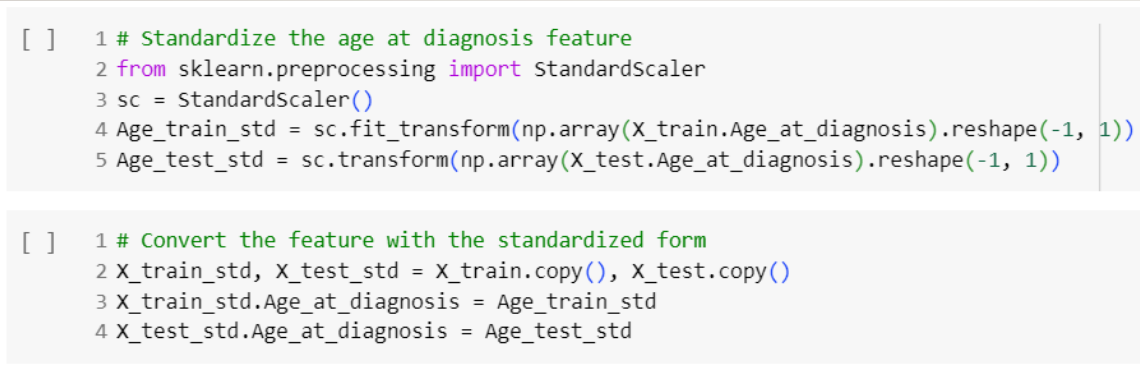
- Data collection and pre-processing
  - Split the dataset into train and test dataset



```
# Split the data using scikit-learn package
from sklearn.model_selection import train_test_split

# Select target
y = glioma_df_encoded.Grade
X = glioma_df_encoded.drop(['Grade'], axis=1)

# Divide data into training and validation subsets
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, test_size=0.2,
                                                      random_state=0)
```

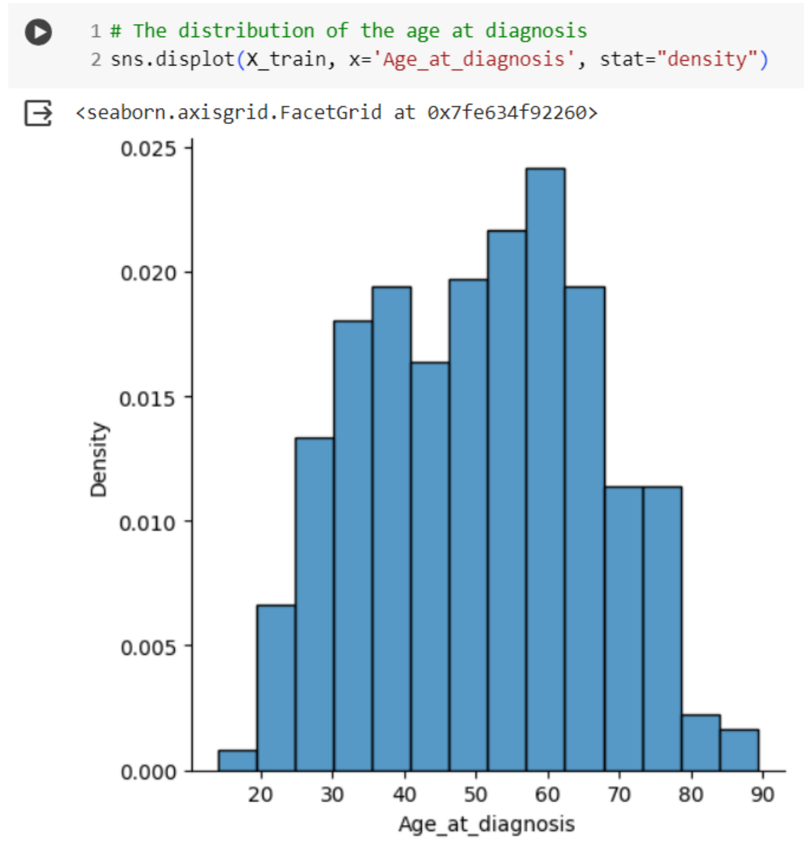
MATERIALS AND METHODS

- Data collection and pre-processing
  - Data scaling (without normalization)



```
# Standardize the age at diagnosis feature
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
Age_train_std = sc.fit_transform(np.array(X_train.Age_at_diagnosis).reshape(-1, 1))
Age_test_std = sc.transform(np.array(X_test.Age_at_diagnosis).reshape(-1, 1))
```

```
# Convert the feature with the standardized form
X_train_std, X_test_std = X_train.copy(), X_test.copy()
X_train_std.Age_at_diagnosis = Age_train_std
X_test_std.Age_at_diagnosis = Age_test_std
```

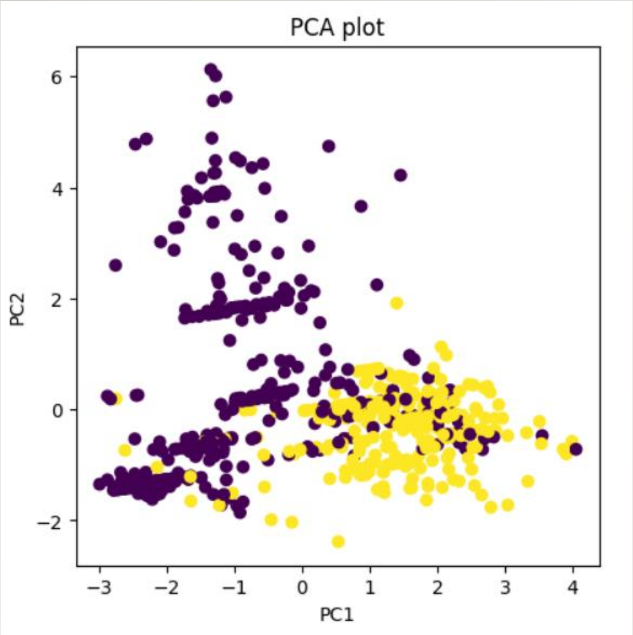


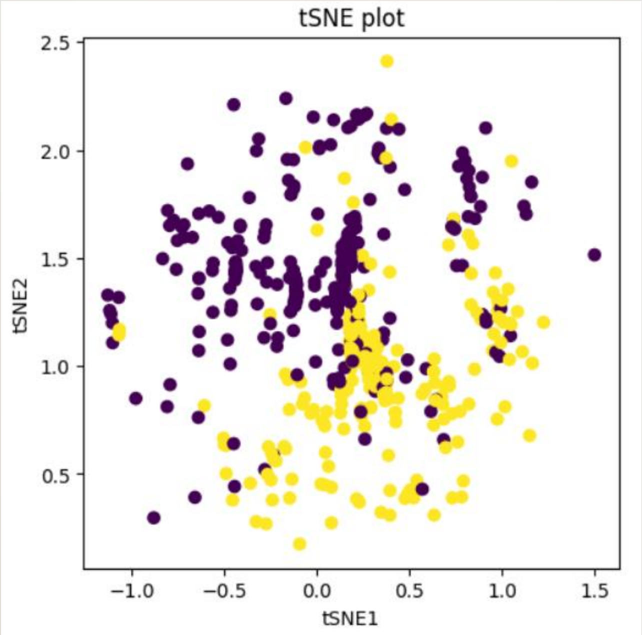
```
# The distribution of the age at diagnosis
sns.displot(X_train, x='Age_at_diagnosis', stat="density")
```

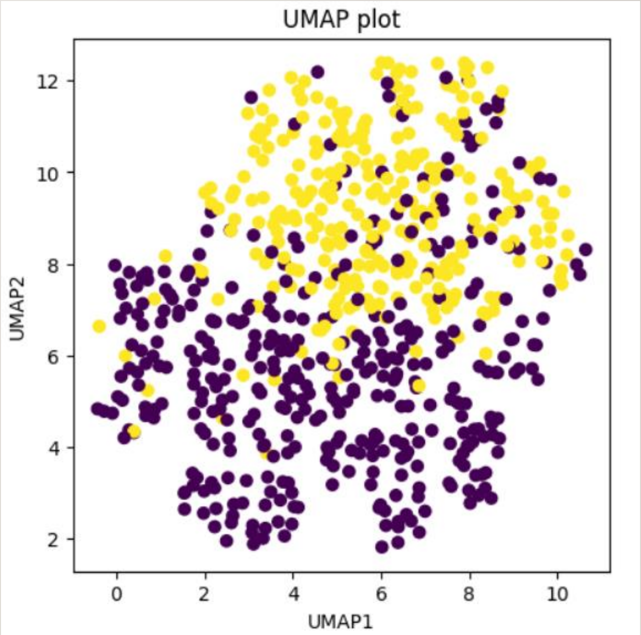
```
<seaborn.axisgrid.FacetGrid at 0x7fe634f92260>
```

MATERIALS AND METHODS

- Data collection and pre-processing
  - Dimensionality reduction

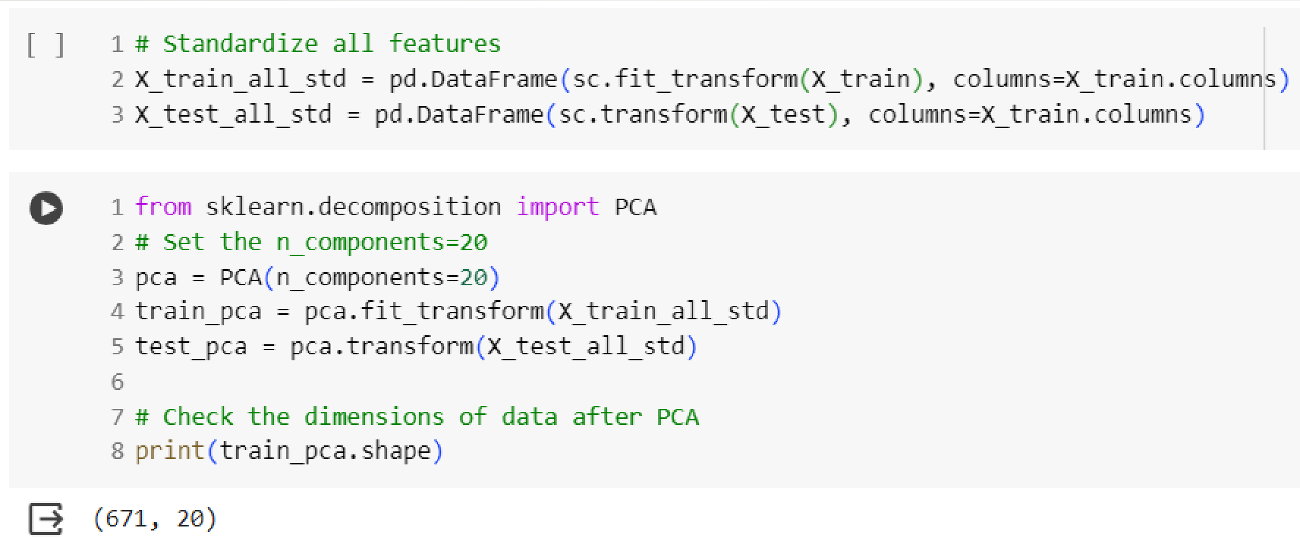






MATERIALS AND METHODS

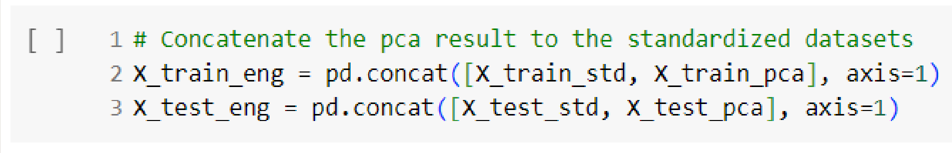
- Data collection and pre-processing
  - Feature engineering



```
# Standardize all features
X_train_all_std = pd.DataFrame(sc.fit_transform(X_train), columns=X_train.columns)
X_test_all_std = pd.DataFrame(sc.transform(X_test), columns=X_train.columns)
```

```
from sklearn.decomposition import PCA
# Set the n_components=20
pca = PCA(n_components=20)
train_pca = pca.fit_transform(X_train_all_std)
test_pca = pca.transform(X_test_all_std)

# Check the dimensions of data after PCA
print(train_pca.shape)
```

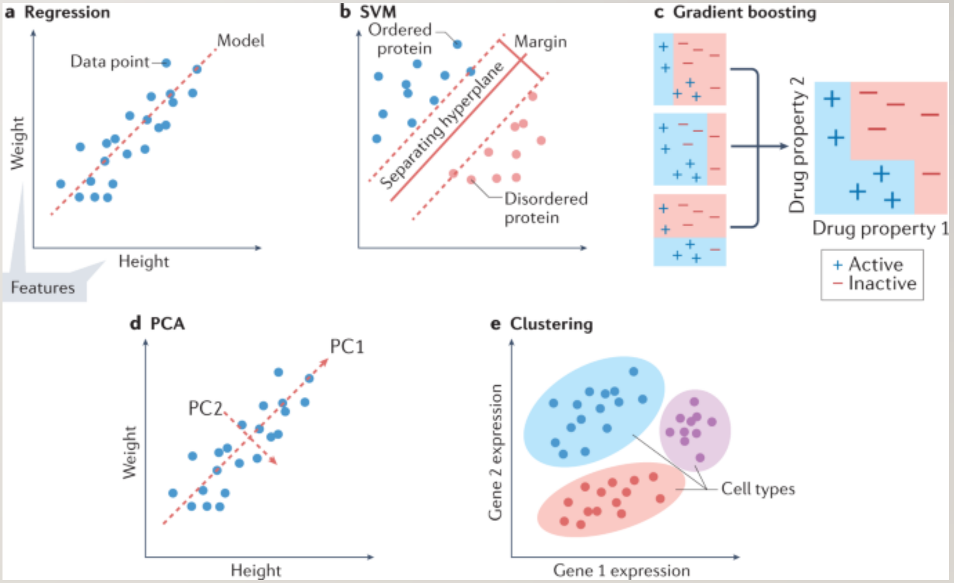
```
(671, 20)
```



```
# Concatenate the pca result to the standardized datasets
X_train_eng = pd.concat([X_train_std, X_train_pca], axis=1)
X_test_eng = pd.concat([X_test_std, X_test_pca], axis=1)
```

MATERIALS AND METHODS

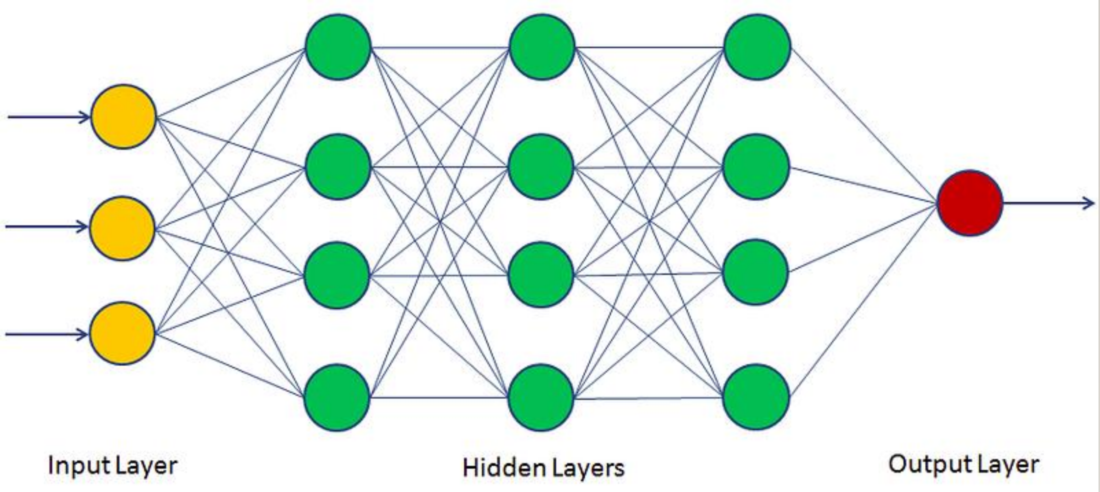
- Model building (nine model classifiers)
  - Using scikit-learn package in Google Colaboratory
  - Perform GridSearch and RandomSearch for hyperparameter tuning
  - 5-fold cross-validation
  - Nested cross-validation in each condition of data manipulation



Greener J.G. et al., 2022.

MATERIALS AND METHODS

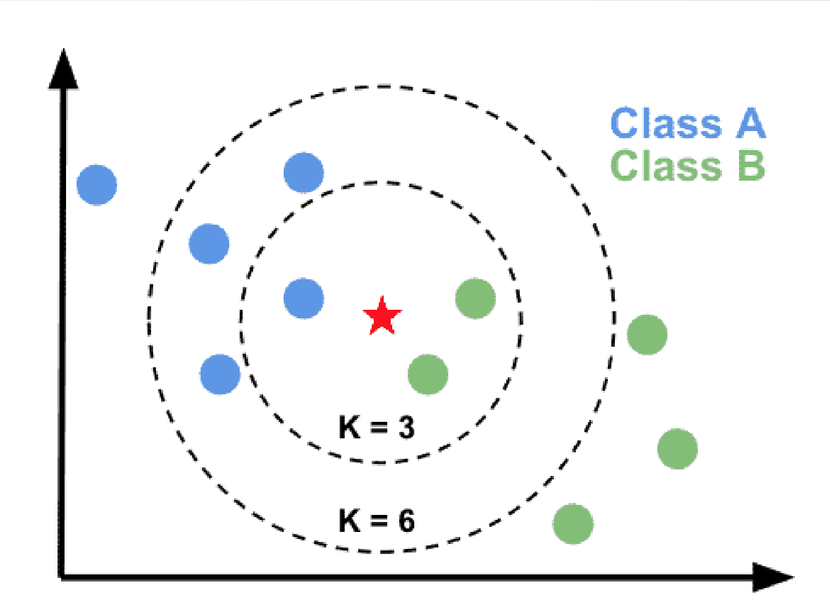
- Model building (nine model classifiers)
  - Multilayer perceptron (MLP)
    - Two hidden layers (32 and 16 nodes)
    - Learning rate: 0.06
    - Batch size: 100
    - Optimize solver: ADAM
    - Sigmoid activation function was used.



https://machinelearninggeek.com/multi-layer-perceptron-neural-network-using-python/

MATERIALS AND METHODS

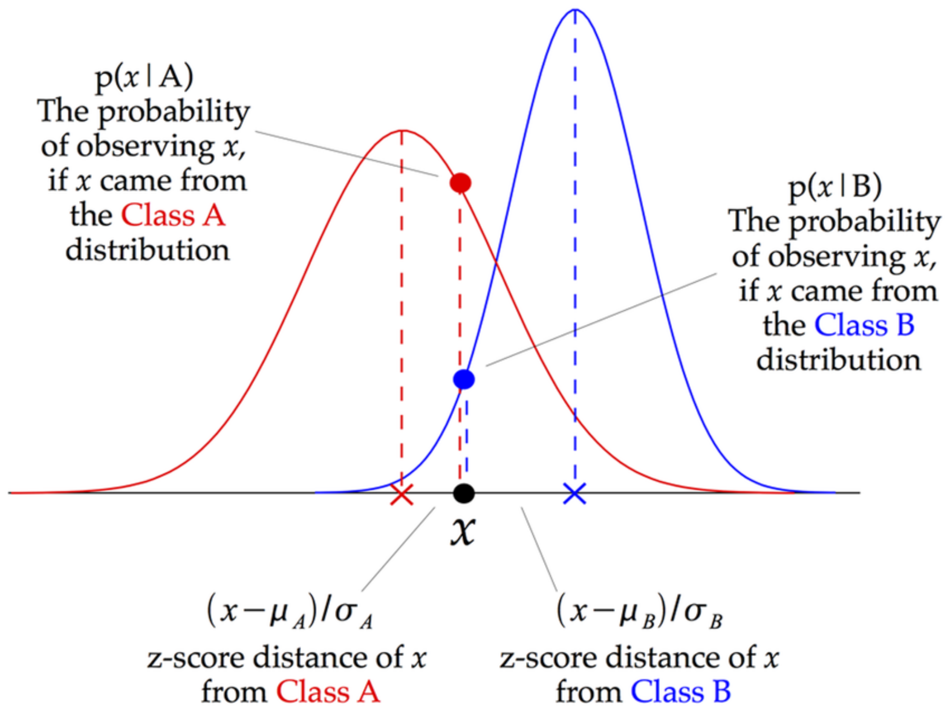
- Model building (nine model classifiers)
  - K-nearest neighbors (KNN): k = 14



https://www.jcchouinard.com/k-nearest-neighbors/

MATERIALS AND METHODS

- Model building (nine model classifiers)
  - Gaussian naïve Bayes classifier
    - Var_smooth: 0.7877

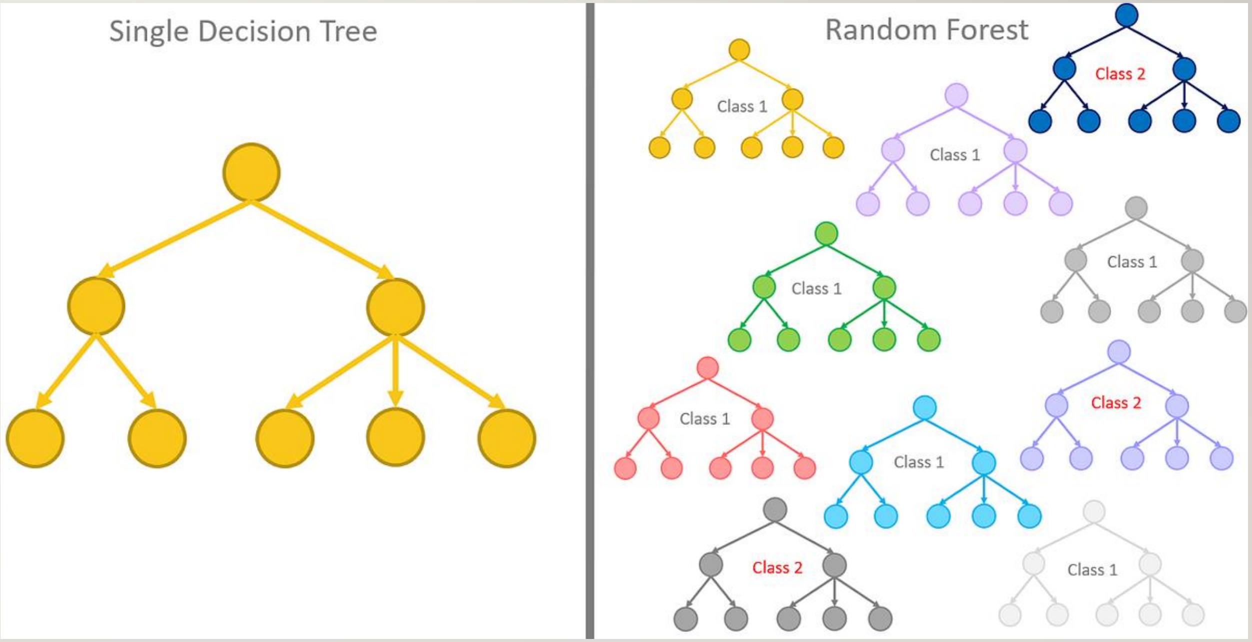


$p(x \mid A)$
The probability of observing x, if x came from the Class A distribution
$p(x \mid B)$
The probability of observing x, if x came from the Class B distribution
x
$(x-\mu_A)/\sigma_A$
z-score distance of x from Class A
$(x-\mu_B)/\sigma_B$
z-score distance of x from Class B

Raizada R. and Sange Lee Y., 20133

MATERIALS AND METHODS

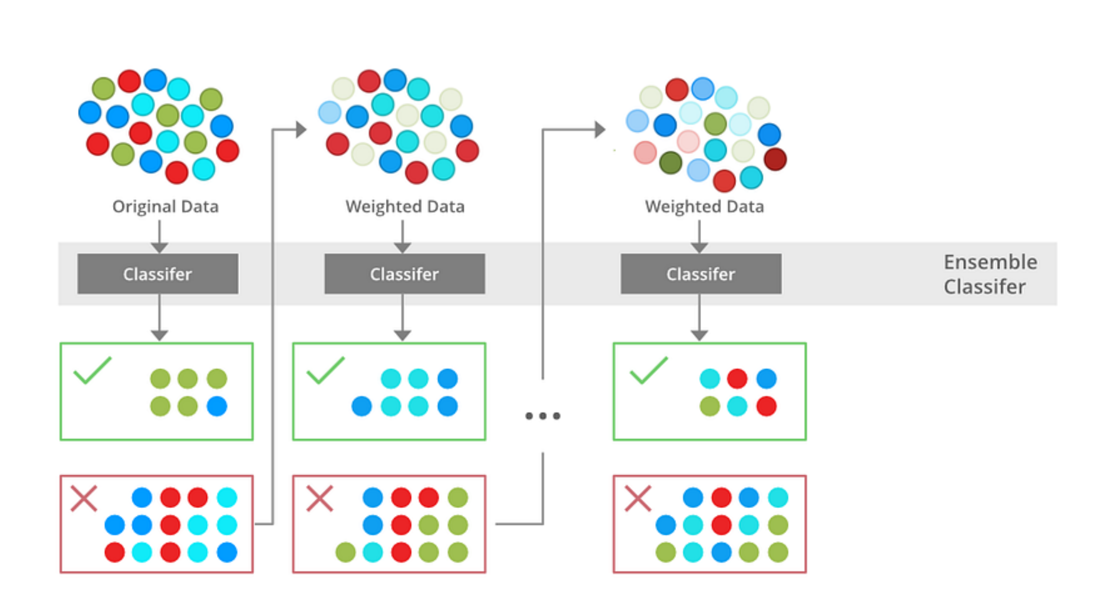
- Model building (nine model classifiers)
  - Decision tree (DT)
    - Maximum depth: 3
    - Minimum samples per leaf: 4
    - Minimum samples split: 96
  - Random forest (RF)
    - 327 trees
    - Maximum depth: 68
    - Minimum samples per leaf: 10
    - Minimum samples split: 85



https://towardsdatascience.com/from-a-single-decision-tree-to-a-random-forest-b9523be65147

MATERIALS AND METHODS

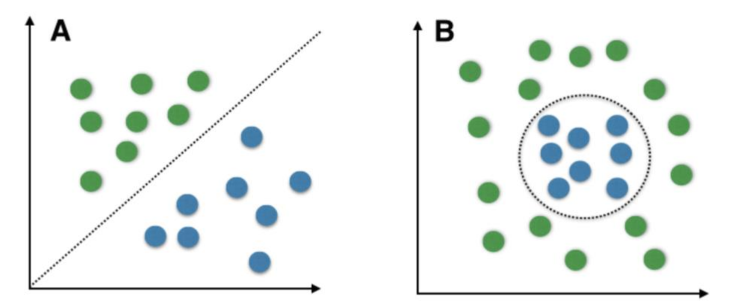
- Model building (six model classifiers)
  - XGBoost
    - 37 trees
    - Maximum depth: 1
    - Max leaf: 50



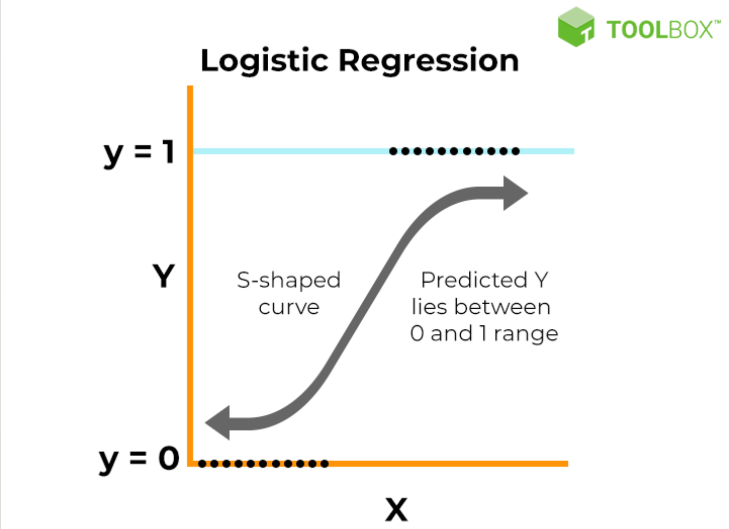
https://medium.com/@techynilesh/xgboost-algorithm-explained-in-less-than-5-minutes-b561dcc1ccee

MATERIALS AND METHODS

- Model building (nine model classifiers)
  - Support vector machine (SVM)
    - Radial basis function kernel: regularization parameter of 225.78
    - Linear kernel: regularization parameter of 7.06
  - Logistic regression (LR)
    - Regularization parameter: 1.18



https://towardsdatascience.com/from-a-single-decision-tree-to-a-random-forest-b9523be65147



https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/

MATERIALS AND METHODS

- Model evaluation
  - Receiver operating characteristic curve (ROC) and area under the curve (AUC)
  - Accuracy
  - Precision
  - Recall
  - F1 score

RESULTS AND DISCUSSION

- Model performance

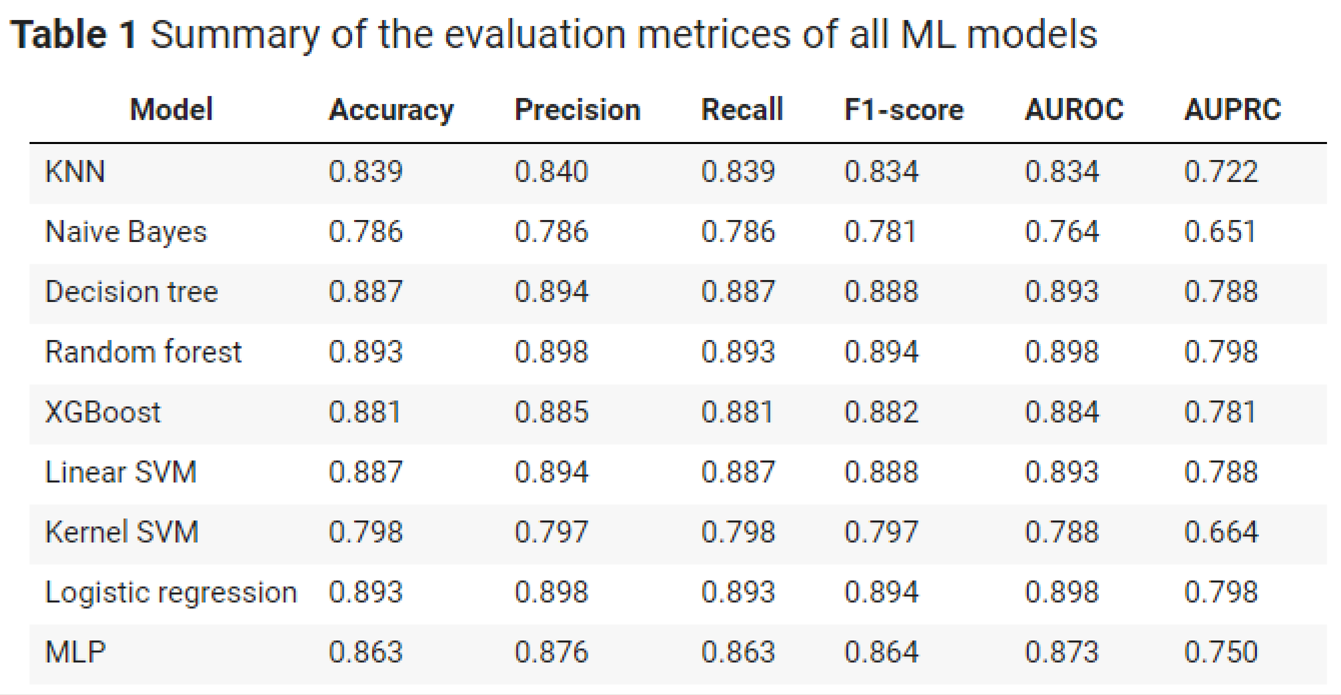
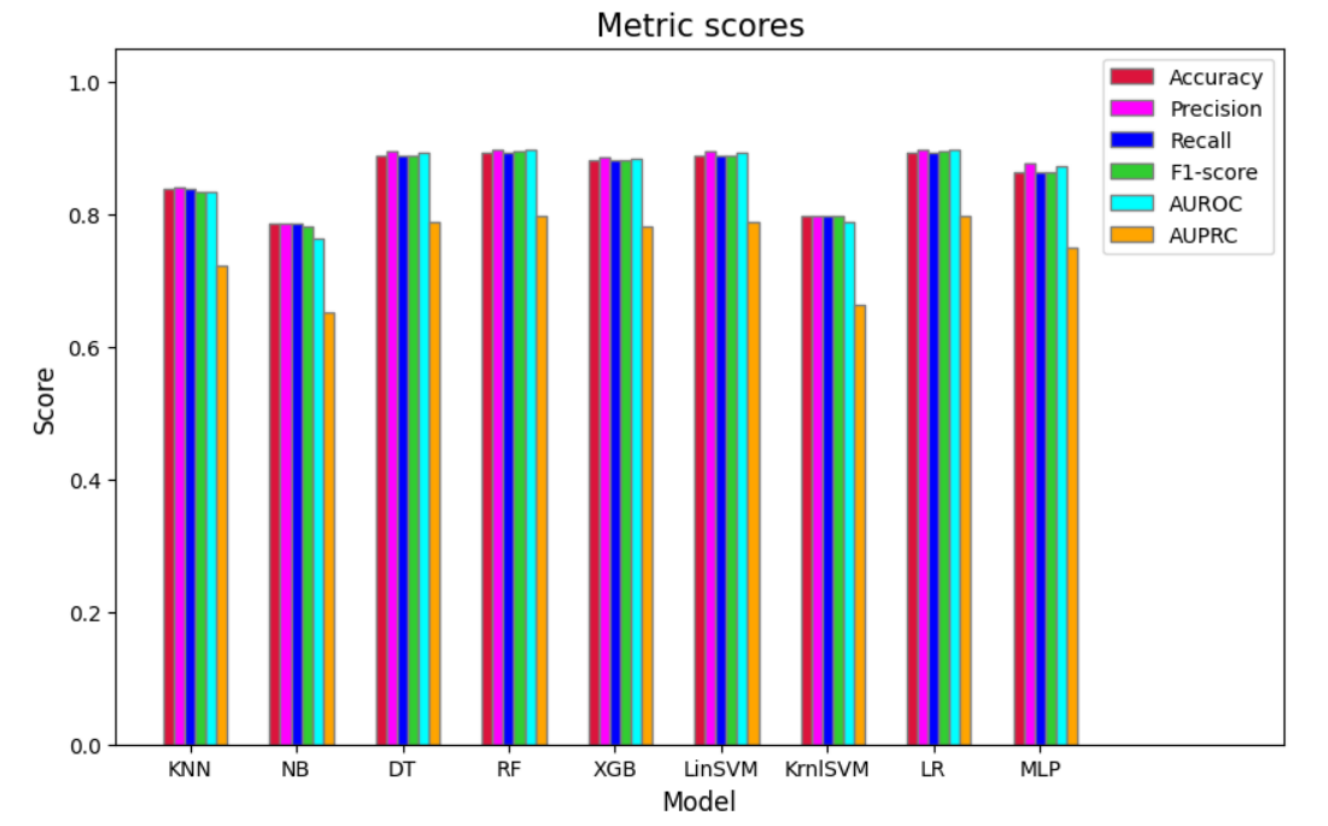


Table 1 Summary of the evaluation metrices of all ML models

Model	Accuracy	Precision	Recall	F1-score	AUROC	AUPRC
KNN	0.839	0.840	0.839	0.834	0.834	0.722
Naive Bayes	0.786	0.786	0.786	0.781	0.764	0.651
Decision tree	0.887	0.894	0.887	0.888	0.893	0.788
Random forest	0.893	0.898	0.893	0.894	0.898	0.798
XGBoost	0.881	0.885	0.881	0.882	0.884	0.781
Linear SVM	0.887	0.894	0.887	0.888	0.893	0.788
Kernel SVM	0.798	0.797	0.798	0.797	0.788	0.664
Logistic regression	0.893	0.898	0.893	0.894	0.898	0.798
MLP	0.863	0.876	0.863	0.864	0.873	0.750

RESULTS AND DISCUSSION

- Model performance



RESULTS AND DISCUSSION

- Nested cross-validation in each condition

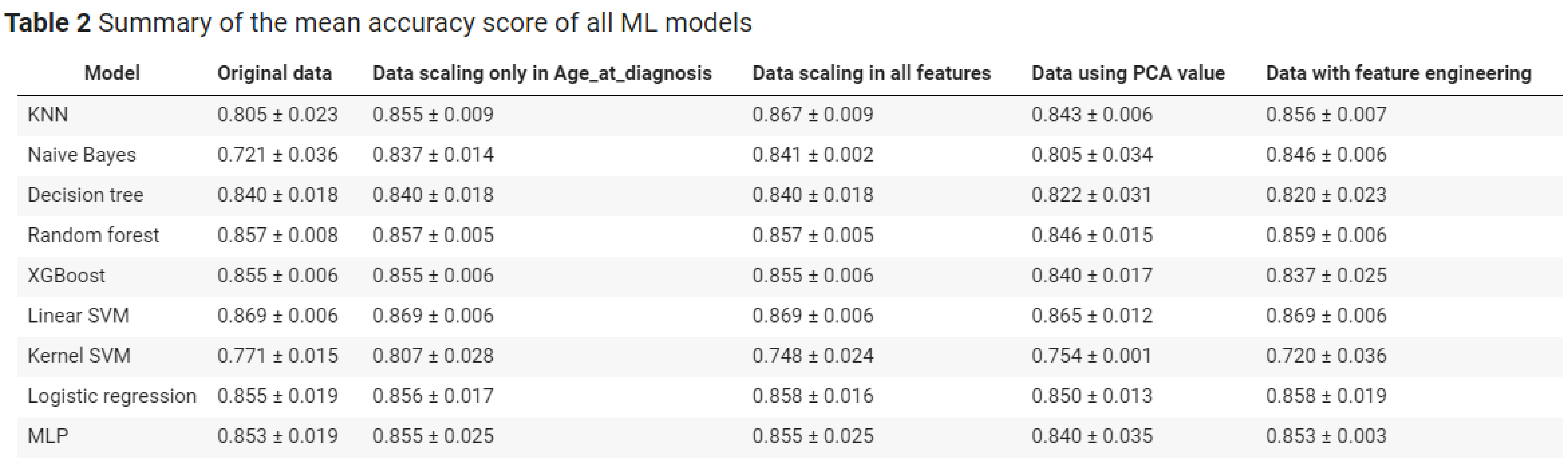
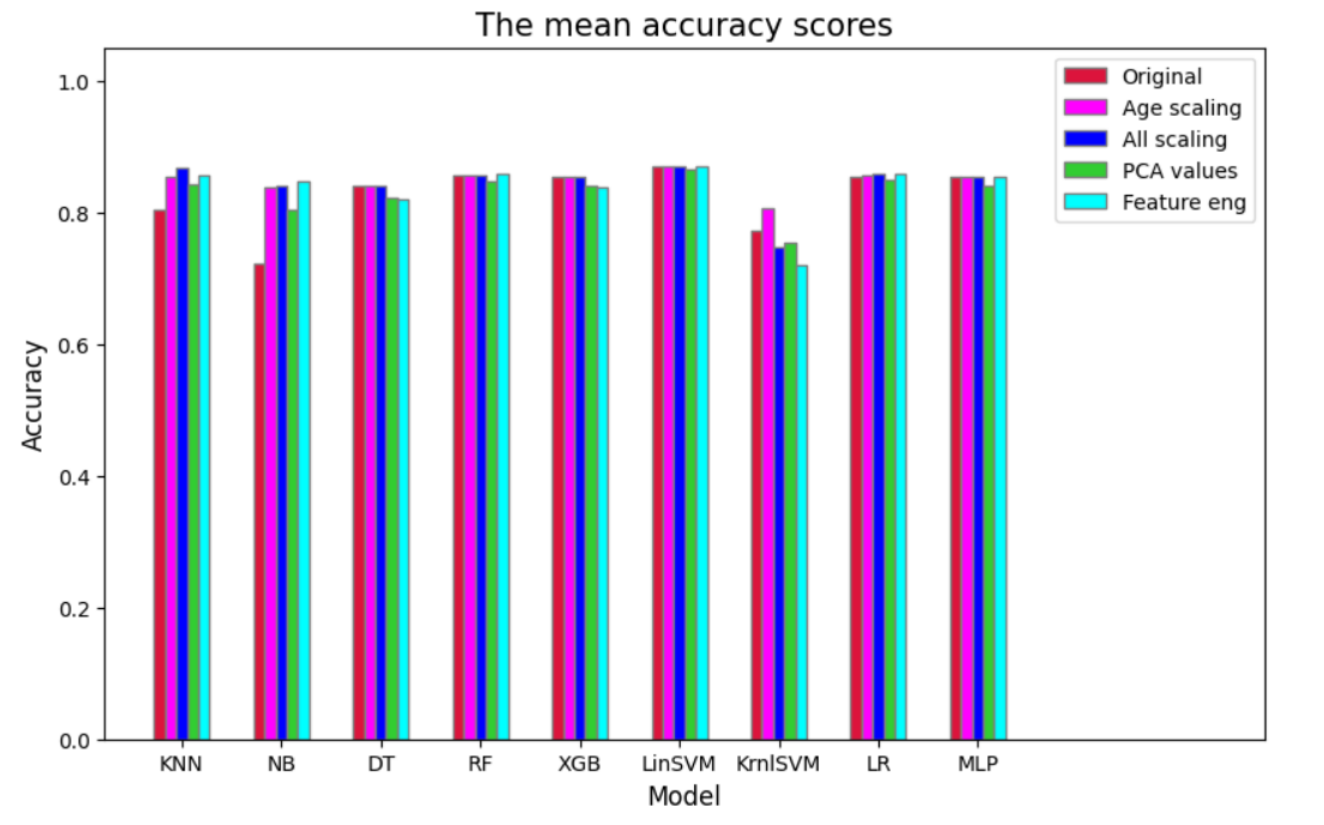


Table 2 Summary of the mean accuracy score of all ML models

Model	Original data	Data scaling only in Age_at_diagnosis	Data scaling in all features	Data using PCA value	Data with feature engineering
KNN	0.805 ± 0.023	0.855 ± 0.009	0.867 ± 0.009	0.843 ± 0.006	0.856 ± 0.007
Naive Bayes	0.721 ± 0.036	0.837 ± 0.014	0.841 ± 0.002	0.805 ± 0.034	0.846 ± 0.006
Decision tree	0.840 ± 0.018	0.840 ± 0.018	0.840 ± 0.018	0.822 ± 0.031	0.820 ± 0.023
Random forest	0.857 ± 0.008	0.857 ± 0.005	0.857 ± 0.005	0.846 ± 0.015	0.859 ± 0.006
XGBoost	0.855 ± 0.006	0.855 ± 0.006	0.855 ± 0.006	0.840 ± 0.017	0.837 ± 0.025
Linear SVM	0.869 ± 0.006	0.869 ± 0.006	0.869 ± 0.006	0.865 ± 0.012	0.869 ± 0.006
Kernel SVM	0.771 ± 0.015	0.807 ± 0.028	0.748 ± 0.024	0.754 ± 0.001	0.720 ± 0.036
Logistic regression	0.855 ± 0.019	0.856 ± 0.017	0.858 ± 0.016	0.850 ± 0.013	0.858 ± 0.019
MLP	0.853 ± 0.019	0.855 ± 0.025	0.855 ± 0.025	0.840 ± 0.035	0.853 ± 0.003

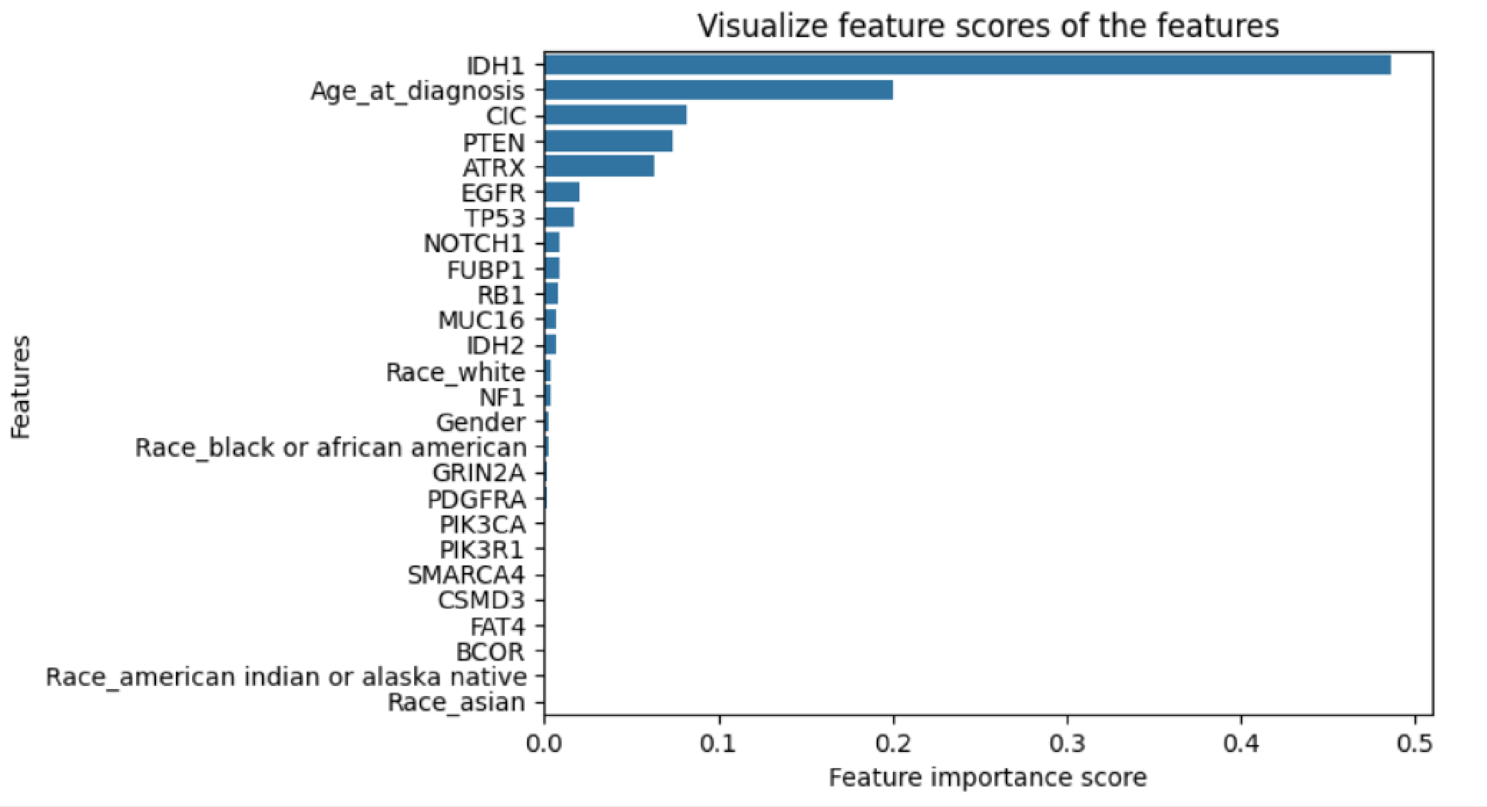
RESULTS AND DISCUSSION

- Nested cross-validation in each condition of data manipulation



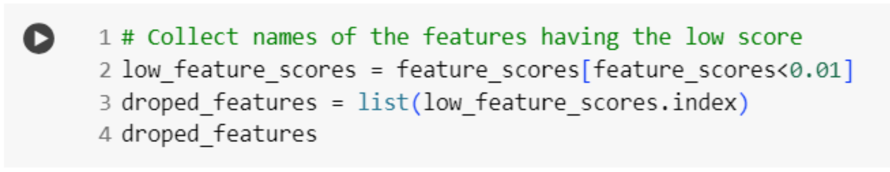
RESULTS AND DISCUSSION

- Important feature analysis

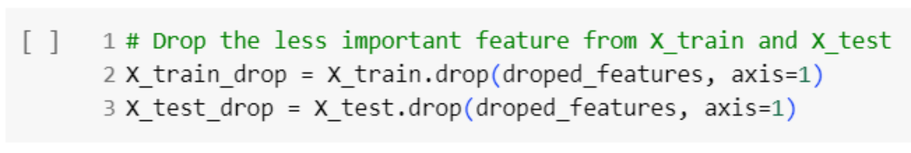


RESULTS AND DISCUSSION

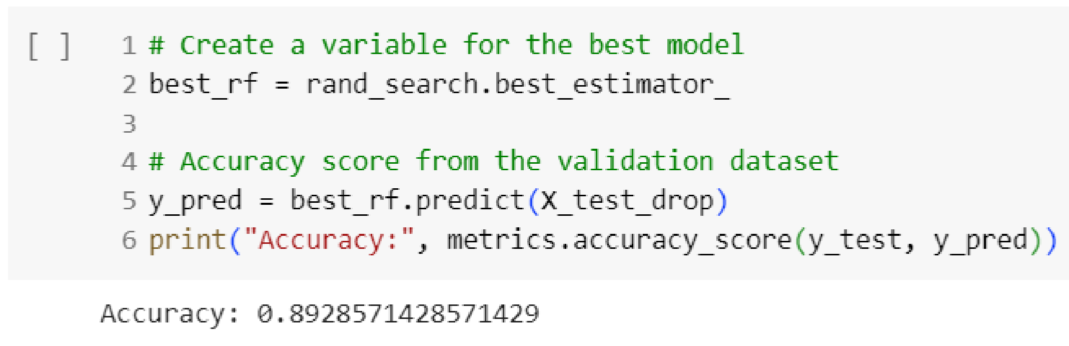
- Important feature analysis : model performance after feature selection



```
# Collect names of the features having the low score
low_feature_scores = feature_scores[feature_scores<0.01]
droped_features = list(low_feature_scores.index)
droped_features
```



```
# Drop the less important feature from X_train and X_test
X_train_drop = X_train.drop(droped_features, axis=1)
X_test_drop = X_test.drop(droped_features, axis=1)
```



```
# Create a variable for the best model
best_rf = rand_search.best_estimator_

# Accuracy score from the validation dataset
y_pred = best_rf.predict(X_test_drop)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.8928571428571429
```

CONCLUSION AND LIMITATION

- Machine learning models can improve brain tumor grading prediction based on the clinical and molecular information.
- KNN, NB, tree-based learnings, linear SVM, logistic regression, and multilayer perceptron are the suitable model to use in this case.
- However, further hyperparameter tuning should be carried out in the future work as well as additional feature collection for improving the model's performance.

THANK YOU
FOR YOUR ATTENTION

Q & A