RUTGERS
UNIVERSITY | NEW BRUNSWICK

Cricket Players (Cricketers) Data Analysis

CS 439 - Introduction to Data Science
Course Project
Professor: Gerard de Melo

Name: Smitkumar Patel
NetID: Shp109
Group: 06
Recitation: 02
Email: shp109@rutgers.edu

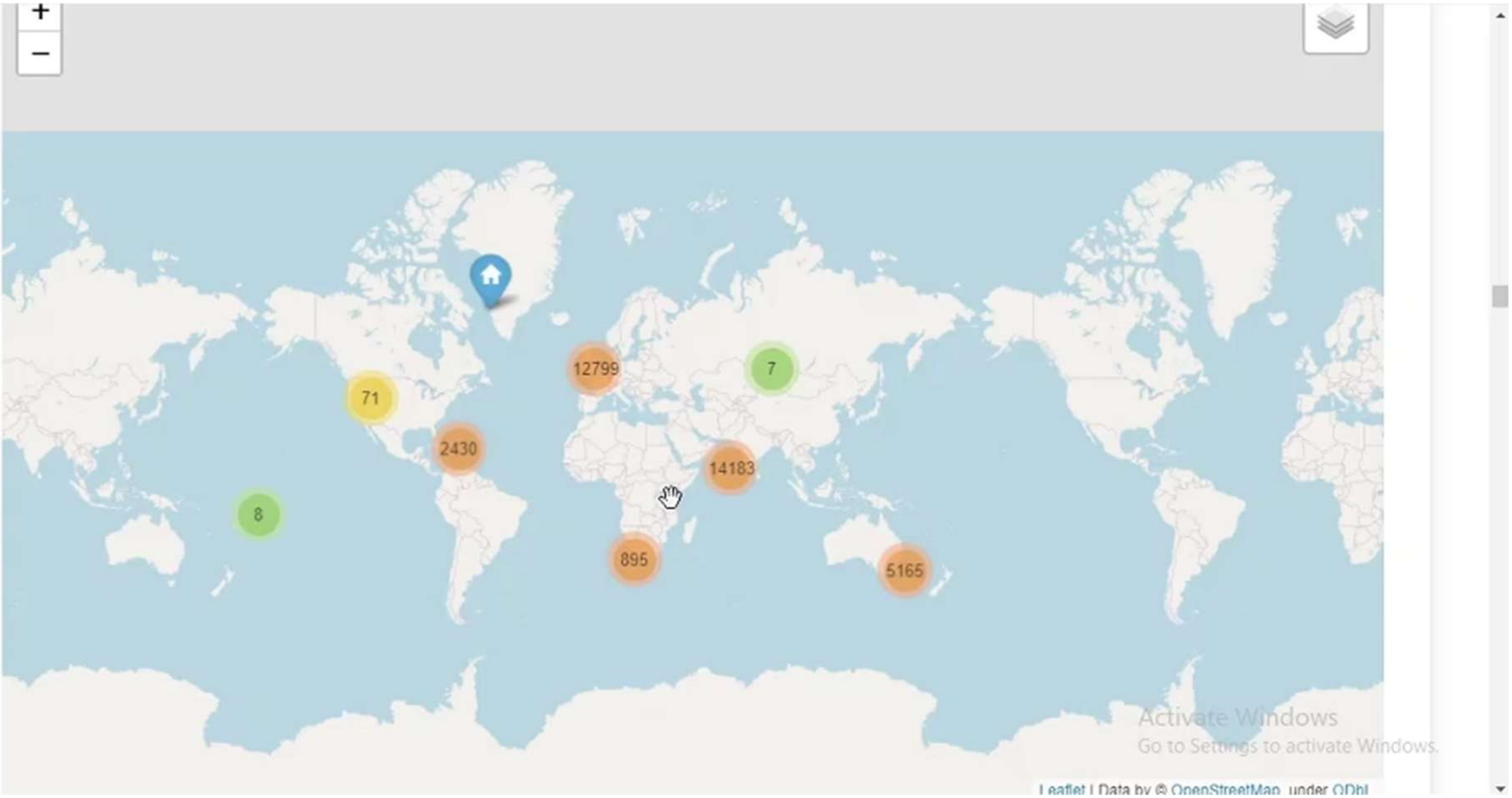
DATA SETS

- **Cricket Players Data from ESPN:**
  The data set contains information about the cricket players all over the world that played Tests, ODI, List A, First-class, T20I, and T20. There are a total of 90308 observations (rows) and 176 variables(columns).
  URL: https://data.world/raghav333/cricket-players-espn

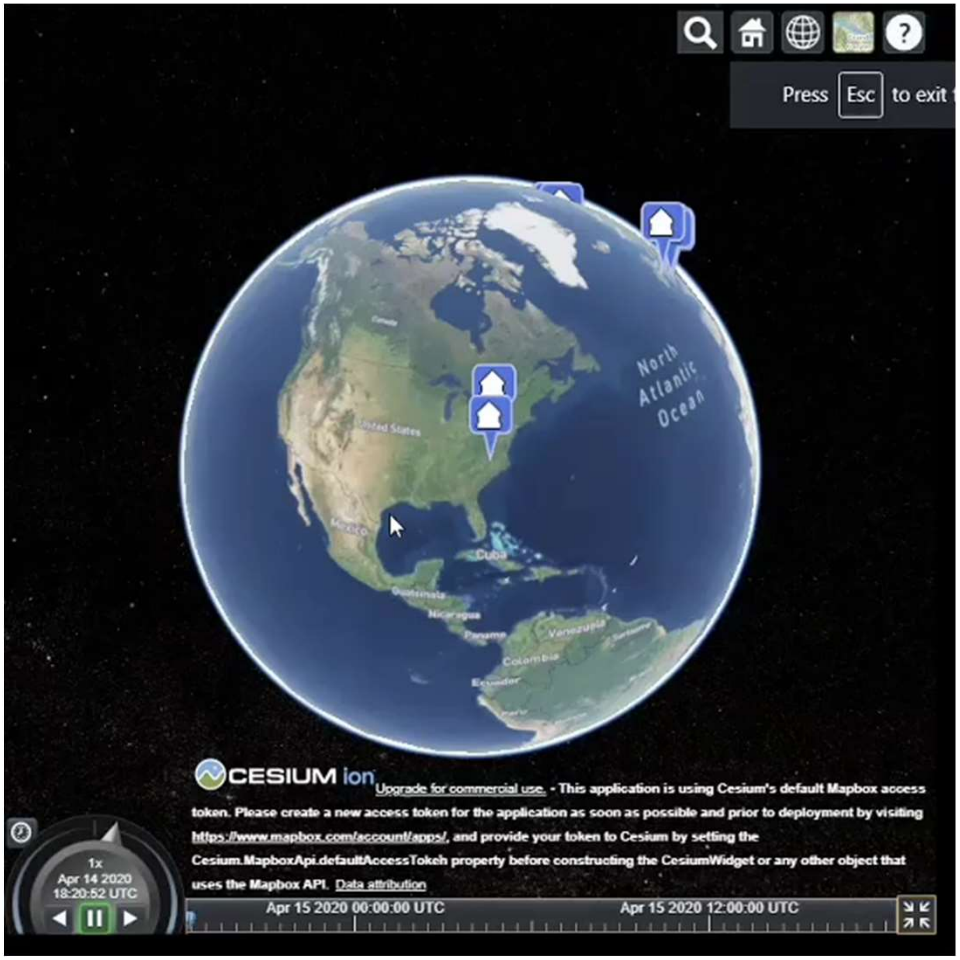
- **IPL Cricket Players Data:**
  The data set contains information about the cricket players that played IPL (Indian Premier League). There are a total of 497 observations (rows) and 7 variables(columns). URL: https://data.world/raghu543/ipl-data-till-2017/workspace/file?filename=Player.csv

Note: I left join ‘Cricket Players Data from ESPN’ and ‘IPL Cricket Players Data’. That means all records from ‘Cricket Players Data from ESPN’, and matched records from the ‘IPL Cricket Players Data’ on ‘Country’, ‘Birthdate’, and ‘Last_Name’. In addition, I created new column that contains binary labels ‘Play’ (player played IPL) and ‘Not Play’ (player did not play IPL).

Number of players from country/city/county

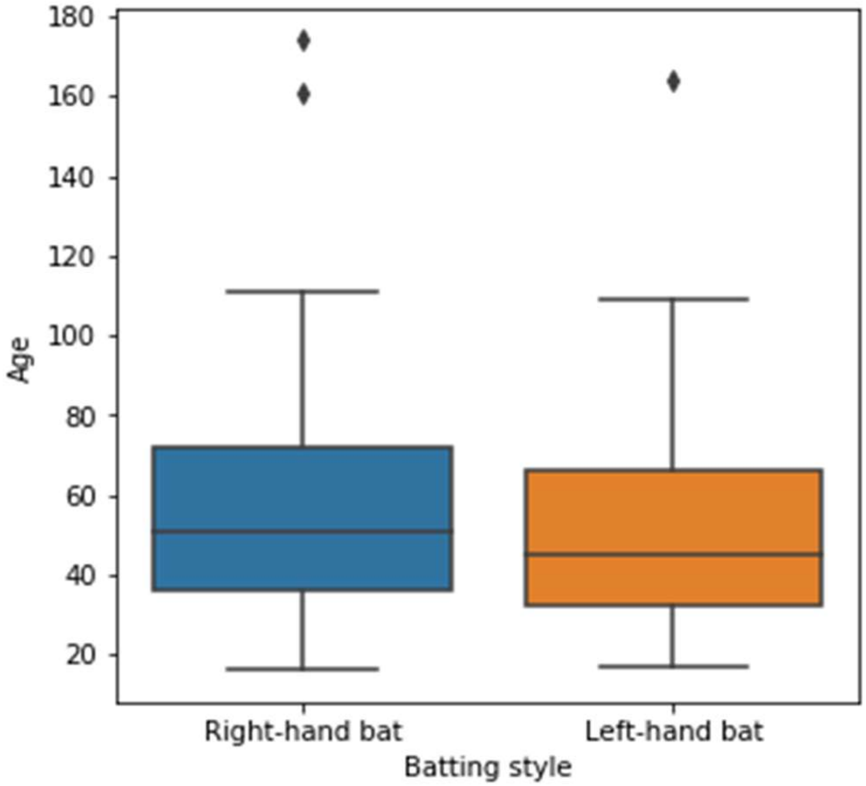

+
−
12799
7
71
2430
14183
8
895
5165
Activate Windows
Go to Settings to activate Windows.
Leaflet | Data by © OpenStreetMap, under ODbL

Top 10 batsmen and bowlers in Test, ODI, and T20

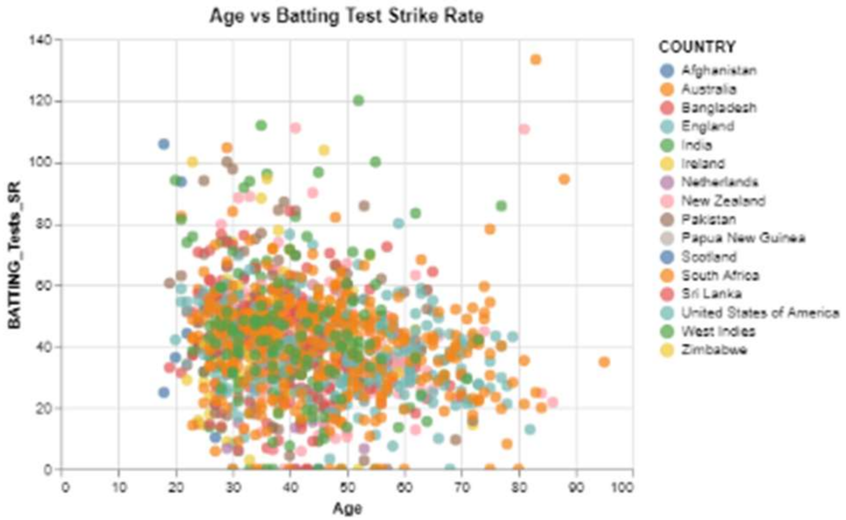


Upgrade for commercial use. - This application is using Cesium's default Mapbox access token. Please create a new access token for the application as soon as possible and prior to deployment by visiting https://www.mapbox.com/account/apps/, and provide your token to Cesium by setting the Cesium.MapboxApi.defaultAccessTokeh property before constructing the CesiumWidget or any other object that uses the Mapbox API. Data attribution

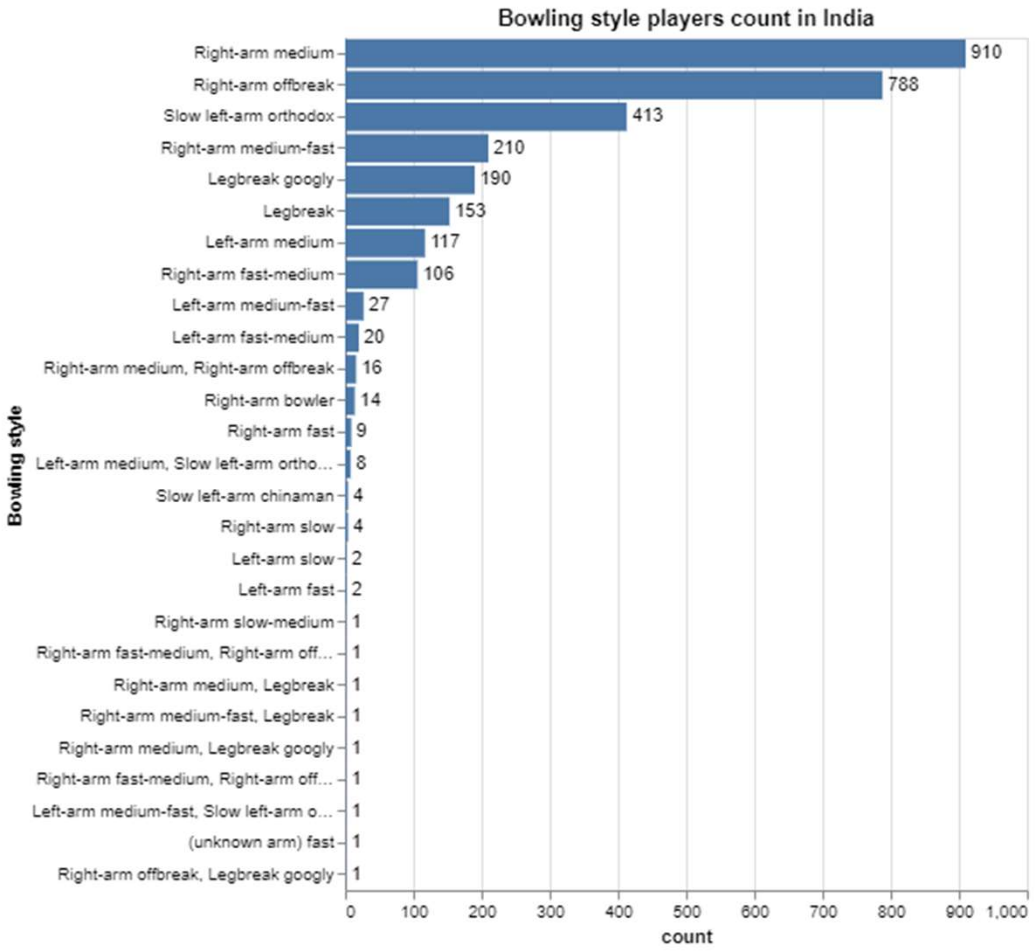
Age of players for various Batting style:



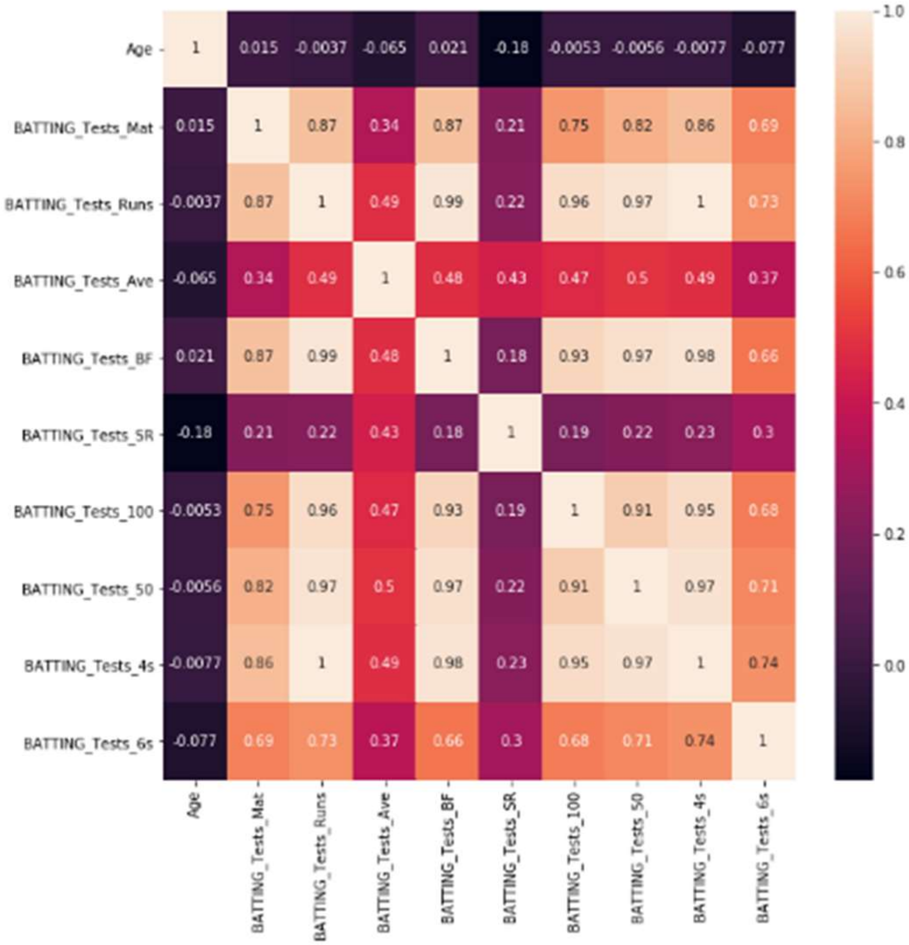
Age vs Batting Test Strike Rate (Test Run/Ball faced in Test):



Bowling style players count in India



Test Batting Data Correlations



	Age	BATTING_Tests_Mat	BATTING_Tests_Runs	BATTING_Tests_Ave	BATTING_Tests_BF	BATTING_Tests_SR	BATTING_Tests_100	BATTING_Tests_50	BATTING_Tests_4s	BATTING_Tests_6s
Age	1	0.015	-0.0037	-0.065	0.021	-0.18	-0.0053	-0.0056	-0.0077	-0.077
BATTING_Tests_Mat	0.015	1	0.87	0.34	0.87	0.21	0.75	0.82	0.86	0.69
BATTING_Tests_Runs	-0.0037	0.87	1	0.49	0.99	0.22	0.96	0.97	1	0.73
BATTING_Tests_Ave	-0.065	0.34	0.49	1	0.48	0.43	0.47	0.5	0.49	0.37
BATTING_Tests_BF	0.021	0.87	0.99	0.48	1	0.18	0.93	0.97	0.98	0.66
BATTING_Tests_SR	-0.18	0.21	0.22	0.43	0.18	1	0.19	0.22	0.23	0.3
BATTING_Tests_100	-0.0053	0.75	0.96	0.47	0.93	0.19	1	0.91	0.95	0.68
BATTING_Tests_50	-0.0056	0.82	0.97	0.5	0.97	0.22	0.91	1	0.97	0.71
BATTING_Tests_4s	-0.0077	0.86	1	0.49	0.98	0.23	0.95	0.97	1	0.74
BATTING_Tests_6s	-0.077	0.69	0.73	0.37	0.66	0.3	0.68	0.71	0.74	1

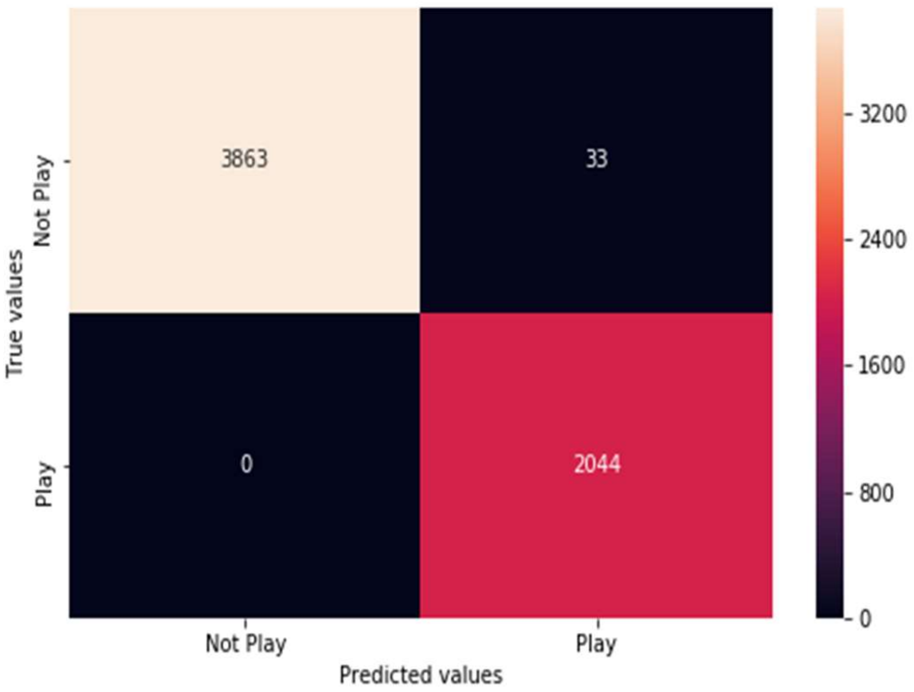
Machine learning algorithms to classify IPL status to be able to know player will play IPL or not

Note: All the accuracy and area under roc curve are acquire by 10-fold cross validation on training dataset.

Model	Average Accuracy	Average Area under Roc Curve	Note
LogisticRegression	0.966788161	0.990200503	
Support Vector Classification	--	--	Took too much time to fit training data
K-nearest neighbors	0.981100088	0.994323267	
Decision Tree	0.989266064	0.994361751	
Random Forest	0.99040264	0.999867555	
Stochastic Gradient Descent	0.899986776	0.964317684	
Gaussian Processes classification	--	--	Took too much time to fit training data
Naive Bayes	0.821736747	0.937844178	
Adaboost	0.9989056	1	
Gradient Boosting	0.979584458	0.995913181	
Histogram-Based Gradient Boosting	0.99360147	0.999849177	
Neural network models	0.939638222	0.977137591	

Voting Classifier model on Test Data

I used voting classifier to use majority vote to predict the IPL status because Random Forest, Adaboost, and Histogram-Based Gradient Boosting model have better accuracy and area under ROC curve.



Acknowledgments

https://geopy.readthedocs.io/en/stable/
https://nbviewer.jupyter.org/github/python-visualization/folium/blob/master/examples/MarkerCluster.ipynb
https://python-visualization.github.io/folium/quickstart.html
https://jupyter-gmaps.readthedocs.io/en/latest/tutorial.html
https://altair-viz.github.io/
https://sandcastle.cesium.com/
https://elitedatascience.com/imbalanced-classes
https://towardsdatascience.com/how-to-visualize-a-decision-tree-in-5-steps-19781b28ffe2
https://medium.com/@dtuk81/confusion-matrix-visualization-fc31e3f30fea
https://github.com/justmarkham/scikit-learn-videos/blob/master/09_classification_metrics.ipynb
https://github.com/justmarkham/scikit-learn-videos/blob/master/08_grid_search.ipynb

Libraries used:

Pandas, Geopy, Seaborn, Folium, Numpy, Matplotlib, Gmaps, Ipython, Altair, re

Scikit-learn libraries used:

train_test_split, classification_report, confusion_matrix, accuracy_score, resample, shuffle, metrics, cross_val_score, Tree, RandomizedSearchCV, GridSearchCV, plot_tree, enable_hist_gradient_boosting, RBF, LogisticRegression, Support Vector Classification (SVC), K-nearest neighbor(KNeighborsClassifiers), Decision Tree, Random Forest(RandomForestClassifier), Stochastic Gradient Descent(SGDClassifier), Gaussian Processes classification(GaussianProcessClassifier), Naive Bayes(GaussianNB), Adaboost(AdaBoostClassifier), Gradient Boosting(GradientBoostingClassifier), Histogram-Based Gradient Boosting(HistGradientBoostingClassifier), Neural network models(MLPClassifier)

Tools used:

Jupyter Notebook, Visual Studio Code

Programing languages used:

Python, HTML, JavaScript