Classifying Bacteria Species

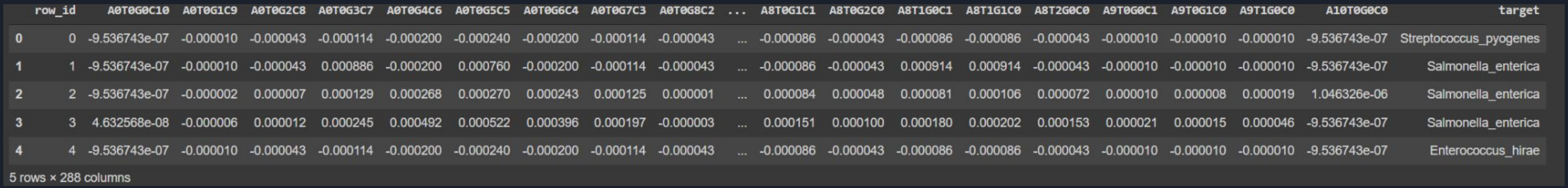
Demetrius Parker, David Camacho, and Simon Mai

Background

Project description: Our task is to classify 10 different bacteria species using data from a genomic analysis technique that has some data compression and data loss.

Project details: The dataset in the project has 288 columns and 200000 rows. It has a column called “target,” which is the y (label), and a column called “row_id,” with the rest of the 286 columns being the X (tags). The tags are the ten different kinds of bacteria species.

Goal: Create a predictive model to predict the bacteria species based on 10-mer snippets of DNA.

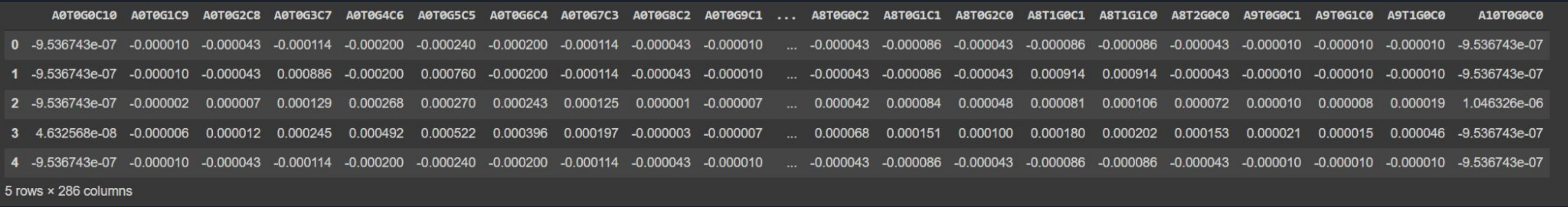


	row_id	A0T0G0C10	A0T0G1C9	A0T0G2C8	A0T0G3C7	A0T0G4C6	A0T0G5C5	A0T0G6C4	A0T0G7C3	A0T0G8C2	...	A8T0G1C1	A8T0G2C0	A8T1G0C1	A8T1G1C0	A8T2G0C0	A9T0G0C1	A9T0G1C0	A9T1G0C0	A10T0G0C0	target
0	0	-9.536743e-07	-0.000010	-0.000043	-0.000114	-0.000200	-0.000240	-0.000200	-0.000114	-0.000043	...	-0.000086	-0.000043	-0.000086	-0.000086	-0.000043	-0.000010	-0.000010	-0.000010	-9.536743e-07	Streptococcus_pyogenes
1	1	-9.536743e-07	-0.000010	-0.000043	0.000886	-0.000200	0.000760	-0.000200	-0.000114	-0.000043	...	-0.000086	-0.000043	0.000914	0.000914	-0.000043	-0.000010	-0.000010	-0.000010	-9.536743e-07	Salmonella_enterica
2	2	-9.536743e-07	-0.000002	0.000007	0.000129	0.000268	0.000270	0.000243	0.000125	0.000001	...	0.000084	0.000048	0.000081	0.000106	0.000072	0.000010	0.000008	0.000019	1.046326e-06	Salmonella_enterica
3	3	4.632568e-08	-0.000006	0.000012	0.000245	0.000492	0.000522	0.000396	0.000197	-0.000003	...	0.000151	0.000100	0.000180	0.000202	0.000153	0.000021	0.000015	0.000046	-9.536743e-07	Salmonella_enterica
4	4	-9.536743e-07	-0.000010	-0.000043	-0.000114	-0.000200	-0.000240	-0.000200	-0.000114	-0.000043	...	-0.000086	-0.000043	-0.000086	-0.000086	-0.000043	-0.000010	-0.000010	-0.000010	-9.536743e-07	Enterococcus_hirae

5 rows × 288 columns

Details about Data

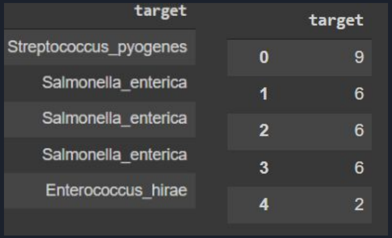
For the data, we needed to remove some columns from it so that it would not interfere with the accuracy we get at the end after training our models. The removal of the “row_id” and “target” column was needed for the X (tags).



	A0T0G0C10	A0T0G1C9	A0T0G2C8	A0T0G3C7	A0T0G4C6	A0T0G5C5	A0T0G6C4	A0T0G7C3	A0T0G8C2	A0T0G9C1	...	A8T0G0C2	A8T0G1C1	A8T0G2C0	A8T1G0C1	A8T1G1C0	A8T2G0C0	A9T0G0C1	A9T0G1C0	A9T1G0C0	A10T0G0C0
0	-9.536743e-07	-0.000010	-0.000043	-0.000114	-0.000200	-0.000240	-0.000200	-0.000114	-0.000043	-0.000010	...	-0.000043	-0.000086	-0.000043	-0.000086	-0.000086	-0.000043	-0.000010	-0.000010	-0.000010	-9.536743e-07
1	-9.536743e-07	-0.000010	-0.000043	0.000886	-0.000200	0.000760	-0.000200	-0.000114	-0.000043	-0.000010	...	-0.000043	-0.000086	-0.000043	0.000914	0.000914	-0.000043	-0.000010	-0.000010	-0.000010	-9.536743e-07
2	-9.536743e-07	-0.000002	0.000007	0.000129	0.000268	0.000270	0.000243	0.000125	0.000001	-0.000007	...	0.000042	0.000084	0.000048	0.000081	0.000106	0.000072	0.000010	0.000008	0.000019	1.046326e-06
3	4.632568e-08	-0.000006	0.000012	0.000245	0.000492	0.000522	0.000396	0.000197	-0.000003	-0.000007	...	0.000068	0.000151	0.000100	0.000180	0.000202	0.000153	0.000021	0.000015	0.000046	-9.536743e-07
4	-9.536743e-07	-0.000010	-0.000043	-0.000114	-0.000200	-0.000240	-0.000200	-0.000114	-0.000043	-0.000010	...	-0.000043	-0.000086	-0.000043	-0.000086	-0.000086	-0.000043	-0.000010	-0.000010	-0.000010	-9.536743e-07

5 rows × 286 columns

The “target” column is the y (label) that contain the ten different kinds of bacteria species. Encoded the categorical features from y using LabelEncoder() method fit_transform().



target
Streptococcus_pyogenes
Salmonella_enterica
Salmonella_enterica
Salmonella_enterica
Enterococcus_hirae

	target
0	9
1	6
2	6
3	6
4	2

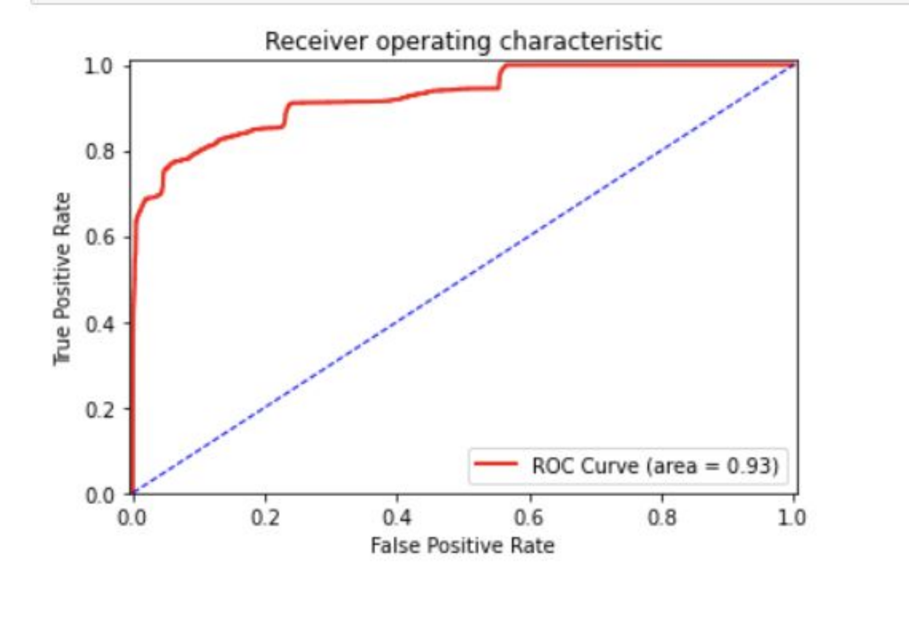
ML Models Used

- Logistic Regression
  - Is a supervised learning. It is the process of modeling the probability of a discrete outcome given an input variable.
- ANN with Keras
  - Is a high-level API built on Tensorflow. Allows the building of simple or complex neural networks within a few minutes.
- ANN with SciKitLearn
  - Is a general machine learning library built on NumPy and has a lot of utilities for pre and post-processing of data. Used to construct traditional models.

Logistic Regression

Predict the category of the bacteria samples using logistic regression.

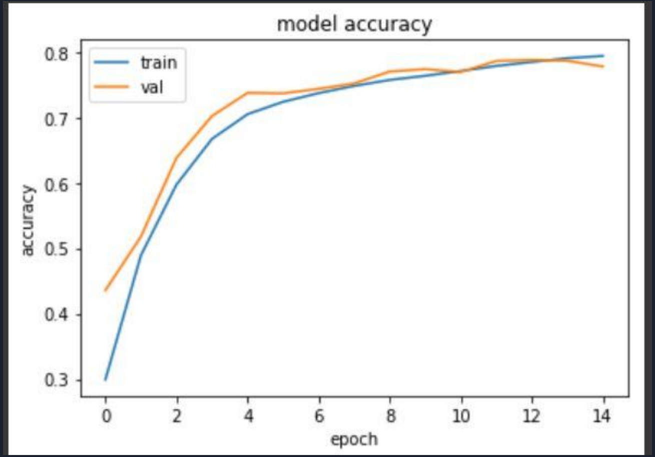
- ROC: 0.93
- Accuracy evaluation: 65.8%
- 10-fold Cross validation: 67.5%

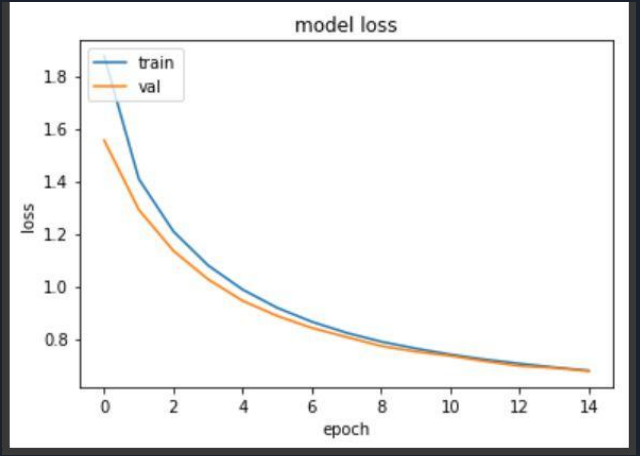


Logistic Regression ROC Curve

ANN with Keras

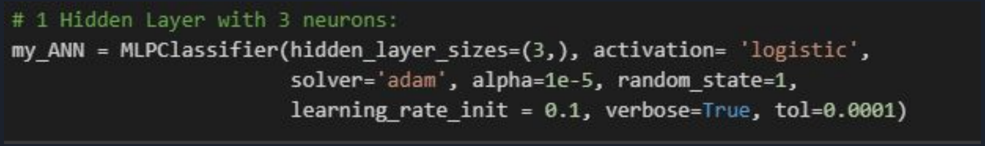
- Compiled the model (loss='categorical_crossentropy', metrics=['accuracy'], optimizer='adam')
- Designed the ANN Structure with 286 inputs, 10 outputs, and 100 hidden neurons
    - First Layer: does not require any processing
    - Second Layer: add a hidden layer (Dense(hidden_neurons, input_dim = input_size)) and ‘sigmoid’ activation
    - Third Layer: output layer (Dense(out_size, input_dim = hidden_neurons)) and ‘softmax’ activation
- The accuracy is: 0.776675
- Model Accuracy and Model Loss across each epoch





ANN with SciKitLearn

Used SciKitLearn MLPClassifier to train only training set with one hidden layer and three neurons.



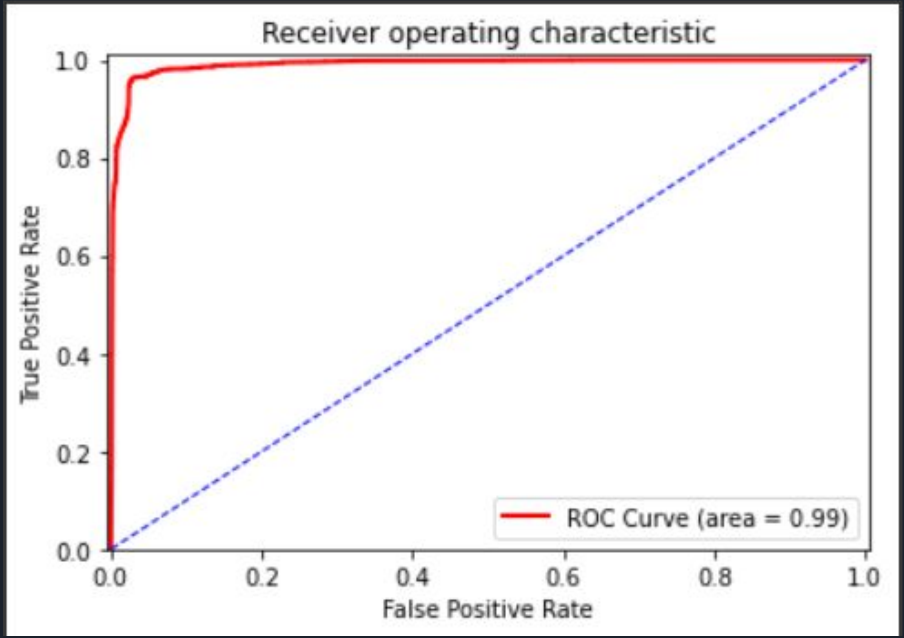
```
# 1 Hidden Layer with 3 neurons:
my_ANN = MLPClassifier(hidden_layer_sizes=(3,), activation= 'logistic',
                       solver='adam', alpha=1e-5, random_state=1,
                       learning_rate_init = 0.1, verbose=True, tol=0.0001)
```

The trained training set was used to compare the “predicted labels” with the “actual labels” and return the accuracy of 0.77115.

ANN with SciKitLearn ROC curve

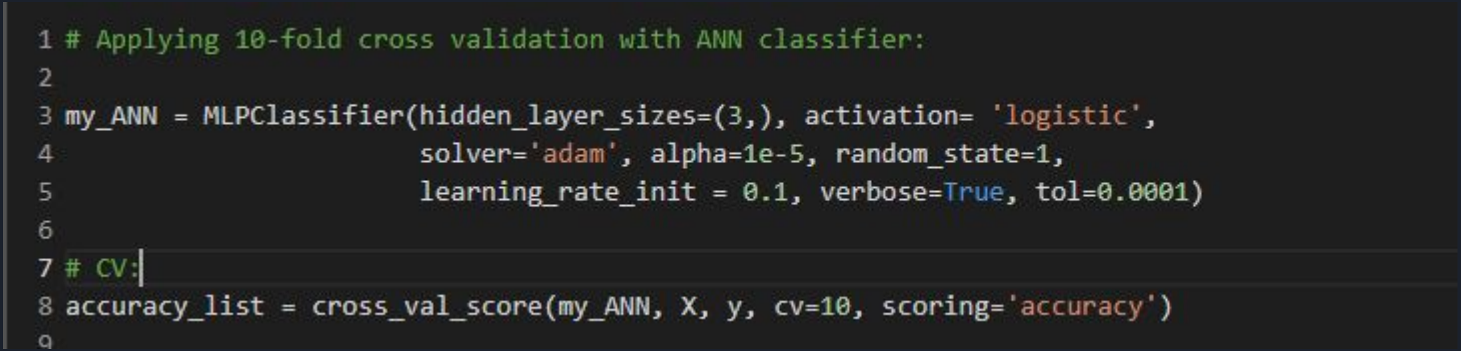
Predict the category of the bacteria samples using logistic regression.

- ROC: 0.990765



10-fold cross validation with ANN classifier

Used the same SciKitLearn MLPClassifier from before and applied 10-fold cross validation.



```
# Applying 10-fold cross validation with ANN classifier:

my_ANN = MLPClassifier(hidden_layer_sizes=(3,), activation= 'logistic',
                       solver='adam', alpha=1e-5, random_state=1,
                       learning_rate_init = 0.1, verbose=True, tol=0.0001)

# CV:
accuracy_list = cross_val_score(my_ANN, X, y, cv=10, scoring='accuracy')
```

Using the list of accuracies found using 10-fold cross-validation, the accuracy is found using the mean() method which gives us an accuracy of 0.792455.

Final Results

SciKitLearn ANN model without Cross-Validation accuracy 0.77115.

SciKitLearn ANN model with a Cross-Validation accuracy of 0.792455.

Keras ANN model accuracy 0.776675.

Logistic Regression without Cross-Validation accuracy 0.6578166.

Logistic Regression with Cross-Validation accuracy 0.67575.

From the accuracy of all our implemented models, the best was the SciKitLearn ANN model, with a Cross-Validation accuracy of 0.792455.

Final Results

The Keras ANN model with an AUC of 77% was the worst. At the same time, the Logistic Regression model with an AUC of 93% did better. But, it was the SciKitLearn ANN model with an AUC of 99% that was the best.

Thank You!