STA 141C Final Project

Yishen Huang (912094269)
Jiannan Zhao (913038538)

Data Description

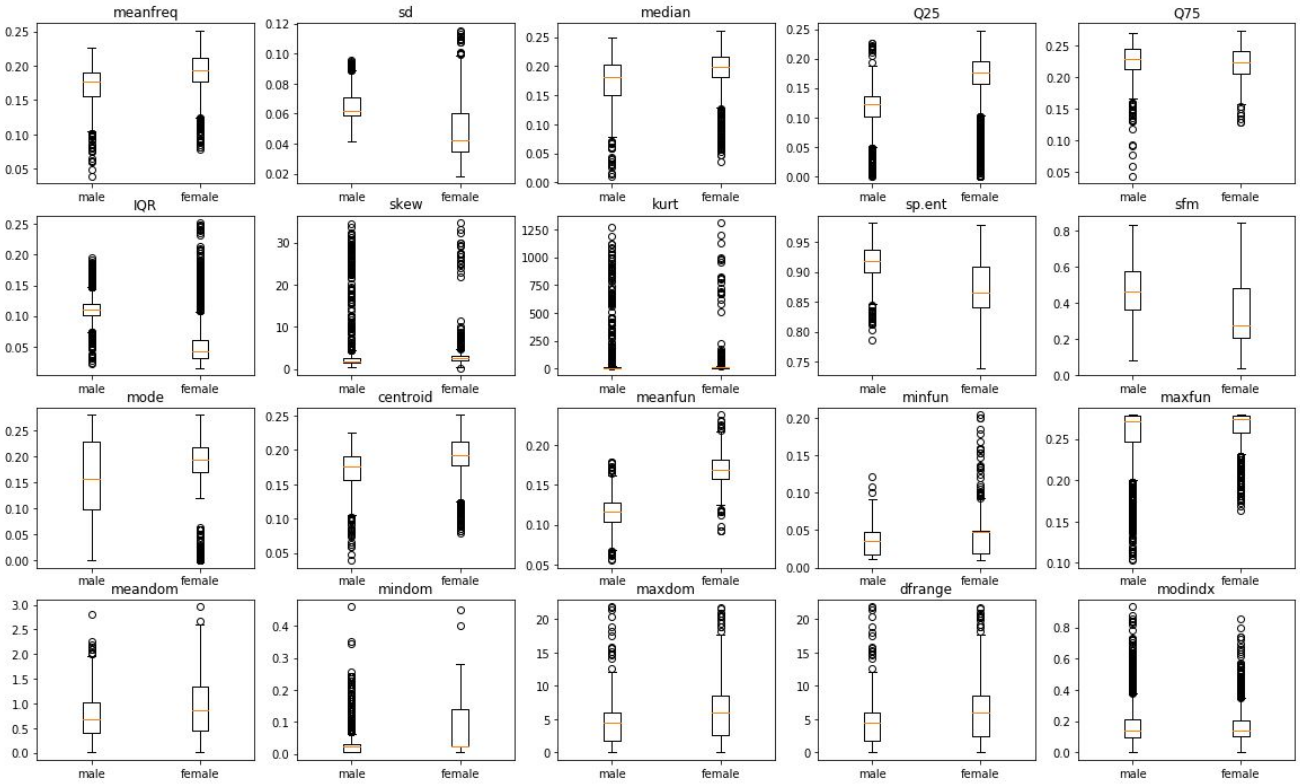
The original data contains 3168 human voices samples represented in twenty acoustic features and one corresponding feature "Label" which indicate genders (male or female). The description of features is shown below.

- Meanfreq: the mean of frequency (kHz)
- Sd: the standard deviation of frequency
- Median: the median value of frequency (kHz)
- Q25: the first quantile of frequency (kHz)
- Q75: the third quantile of frequency (kHz)
- IQR: interquantile range of frequency (kHz)
- Skew: skewness
- Kurt: kurtosis
- Sp.ent: spectral entropy
- Sfm: spectral flatness
- Mode: mode frequency
- Centroid: frequency centroid
- Meanfun: average of fundamental frequency measured across acoustic signal
- Minfun: minimum fundamental frequency measured across acoustic signal
- Maxfun: maximum fundamental frequency measured across acoustic signal
- Meandom: average of dominant frequency measured across acoustic signal
- Mindom: minimum of dominant frequency measured across acoustic signal
- Maxdom: maximum of dominant frequency measured across acoustic signal
- Dfrange: range of dominant frequency measured across acoustic signal
- Modindx: modulation index. Calculated as the accumulated absolute difference between adjacent measurements of fundamental frequencies divided by the frequency range
- Label: male and female

Goal

For this project, we compared four models which are CNN, SVM, KNN, and Logistic Regression to test the corresponding accuracy which is the correct fraction of classifying the gender of voice and compared the advantages and disadvantages of them respectively. We choose these four models because they are all representative models in classification.

Prediction the Importance of Features Before Modeling



Before modeling, we first predicted the importance of different features by box-plot. In the above box-plots, each one contained one feature corresponding to male and female. Firstly, we compared the mean difference between male and female of each feature. If the mean difference is large, it is easy to classify whether the voice comes from male or female. Then the feature could be important to separate gender. For example, features which have large mean difference include “sd,” “sp.ent,” “IQR,” “meanfun” and “maxdom.” Secondly, “overlapping” could also be a reference to judge if the feature should be dropped or not. “Overlapping” means the inter-quantile of one gender is larger than another one. For example, the feature “mode,” the inter-quantile of male is larger than that of female, which makes it easy to misclassify. “Overlapping” plots also include features like “meandom,” “maxdom” and “dfrange.” Thirdly, outliers affect the importance of features as well. For features like “skew”, it contains too many outliers which will affect the weights of this feature in the model. The correctness of our prediction is shown in the **Extra Tree for Feature Extraction** section.

Cross Validation

Cross-validation can be used to compare the performances of different predictive modeling procedures. In our four models, we used 10-fold cross-validation and each subset includes 316 test data and 2852 training data. Using cross-validation, we could objectively compare these four methods in terms of their respective fraction of misclassified gender.

Extra Tree for Feature Extraction (package: sklearn.ensemble import ExtraTreesClassifier)

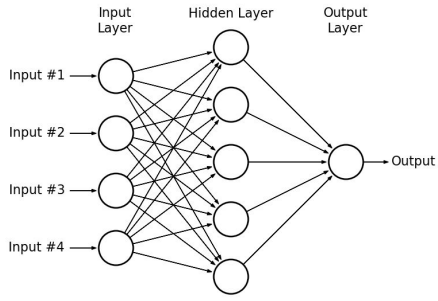
We implemented Extra Tree for feature extraction. Extra Tree implements a meta estimator that fits a number of randomized decision trees on various sub-samples of the dataset. The main effect of Extra Tree is to do classification, but it also works for selecting the importance of all features. By Extra Tree, we obtained importance for each feature as the table below.

feature	modindx	maxfun	meandom	centroid	skew	kurt	maxdom
importance	0.00669	0.00763	0.01191	0.01225	0.01246	0.01273	0.01289
feature	minfun	median	mode	mindom	Q75	dfrange	meanfreq
importance	0.01294	0.01310	0.01361	0.01507	0.01526	0.01695	0.01769
feature	sp.ent	sfm	sd	Q25	IQR	meanfun	
importance	0.03417	0.05253	0.0600	0.10041	0.27388	0.29770	

The five most important features which are “sfm,” “sd,” “Q25,” “IQR” and “meanfun.” Compared to the above prediction part, we can roughly conclude that we have the correct predictions about feature selection shown by the Extra Tree. On the other hand, “mean difference,” “overlapping” and “outliers” can be reliable evidence for us to predict the importance of features before modeling.

CNN (package: keras)

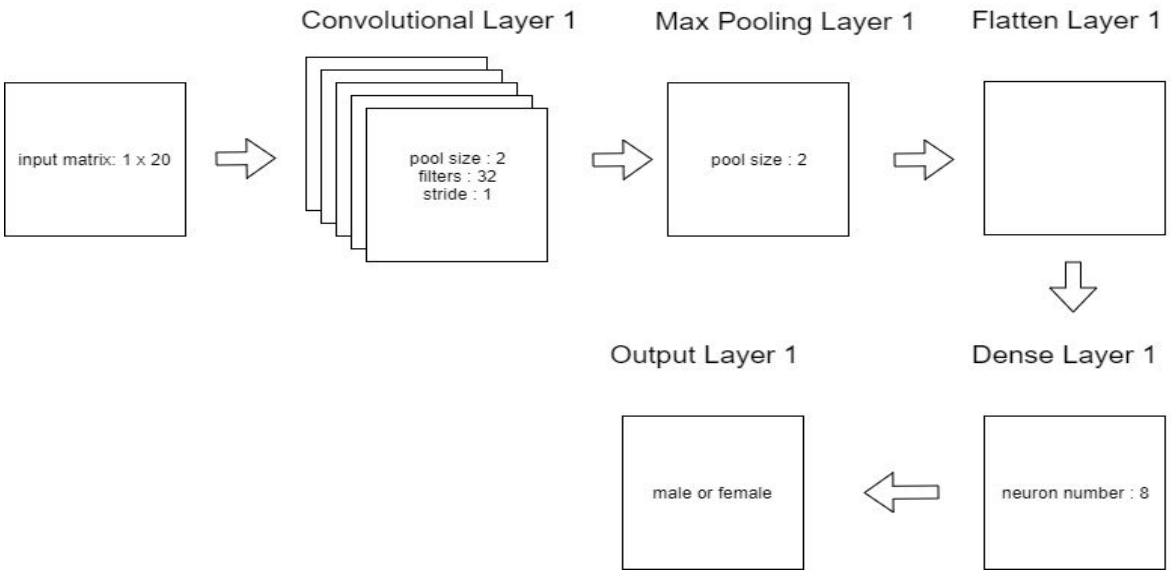
Convolutional Neural Network (CNN) which is so far been most popularly used for analyzing images. This model is a class of deep, feed-forward artificial neural networks which has some specialization for being able to pick out or detect patterns and make sense of them. Neural networks are an interconnected web of notes, which are called neurons. A neuron network main function is to receive a set of inputs, perform complex calculations and then give an output as demonstrated by the diagram below.



CNN includes multiple kinds of layers including convolutional layers, dense layers, dropout layers, flatten layers and so on, but the most important ones are convolutional layers. Usually, CNN contains two major parts: a convolutional part (which has convolutional layers and maxpooling layers) and an artificial neural network (ANN) part.

CNN structure:

The CNN model we implemented includes 1 convolutional layer, 1 maxpooling layer, 1 flatten layer, and 1 dense layers as the following diagram.



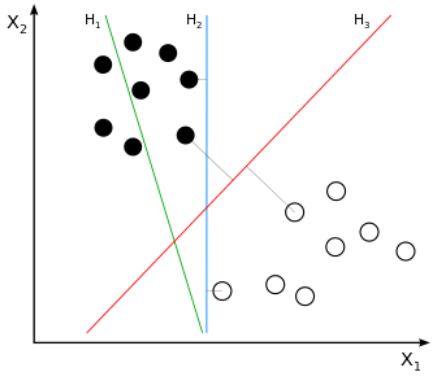
Since the applying different sizes of input for CNN requires changing the structure of it. We only used the original data as the input which has 20 features. The confusion matrix of the prediction result for 316 testing data is demonstrated as the following.

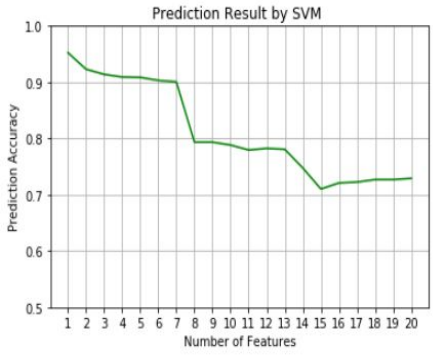
Result:

Accuracy(subset)	0.95569	0.97151	0.97468	0.94936	0.96202
Accuracy(subset)	0.95253	0.97468	0.94303	0.97784	0.96835
Total Accuracy (average)	0.96297				

From the result of CNN we can tell this model has very high accuracy in classification.

SVM (package: sklearn.svm)



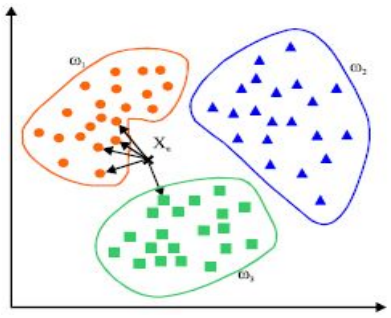


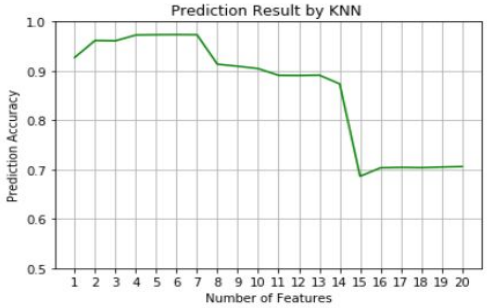
In SVM, it needs to construct one hyperplane or sets of hyperplane to classify or separate in high-dimension space by maximizing the distance between hyperplane and the nearest data points of any class. For left picture, it shows H1 can not separate black and white dots and although H2 classify these two kinds of dots, it does not maximize the distance between dots and H2. H3 is the best plane to classify these dots.

Result:

We tried to implement the classification by using SVM. By above right plot, the X-axis shows the number of ordered important features contained in the model got by the Extra Tree and Y-axis is the predicted accuracy. It shows the trend of accuracy increasing continually as the decrease in the number of features. It means SVM does not depend on much the correlations between different features but it does depend on the feature selections before modeling. Until the number of features equals seven, the accuracy reaches at least 90% which containing features "meanfreq," "sp.ent," "sfm," "sd," "Q25," "IQR," and "meanfun." Then the predicted accuracy reaches its maximum 0.9522 when the model only contains one feature "meanfun."

KNN (package:sklearn.neighbors.KNeighborsClassifier)



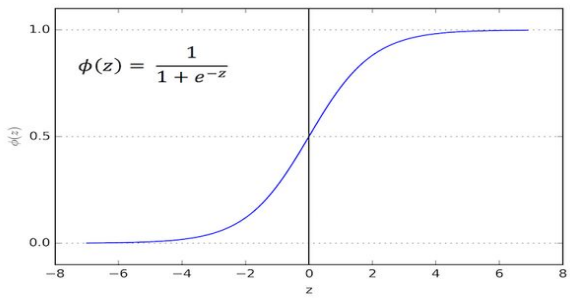


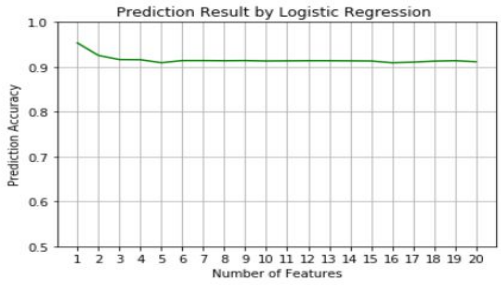
In KNN, it aims to classify one point by getting the mean of its surrounding points. By the left picture, it should determine how many (K) clusters in advance. The first step is to pick K random points which will be considered as the centroids (m1,m2... mk) of the initial clusters (C1, C2...Ck). Then, the second step is, for each observation, to assign it to one of (C1, C2...Ck), depending on which cluster centroid is closest to that observation. The third step is, for each cluster, to compute the mean of all observations belonging to that cluster. Then these mean values will be the new centroids. Lastly, repeating the second and third steps repeatedly until the mean of each cluster will not change anymore.

Result:

We tried to implement the classification by using KNN. By the above right plot, the X-axis shows the number of ordered important features contained in the model got by the Extra Tree and Y-axis is the prediction accuracy. We can see the accuracy remain stable until the number of features to 15. Then it reaches its maximum of 0.973 when the number of features is 5 and it decreases slowly to the end. This means KNN depends on the correlation between different features and it also does depend on feature selections. It is different from the plot of "SVM" which has an increasing continually. The features are "sfm," "sd," "Q25," "IQR," and "meanfun" when accuracy reaches its highest point.

Logistic Regression (package:sklearn.linear_model.LogisticRegression)





By the left picture, it is the logistic function of logistic regression which is used to estimate the probability of a binary response based on one or more predictors. The coefficients of the logistic regression implement the iterative method such as gradient descent. It starts with a tentative solution then revises it to see if it improves the result, and repeat it until no any improvement is made which means it converges to one value.

Result:

We tried to implement the classification by using the logistic regression. By the right plot above, the X-axis shows the number of ordered important features contained in the model got by the Extra Tree and Y-axis is the predicted accuracy. We can directly see it is quite different from the plots of SVM and KNN that the starting accuracy point is higher than 0.9 and it remains stable until the number of features reduces to 2. The accuracy reaches the maximum value 0.9537 when the model only contains one feature. Then we can conclude that logistic regression does not depend on the feature selection as well as the correlations between features.

Evaluation

	CNN	SVM	KNN	Logistic
Running time	**754.16s**	**53.95s**	**2.9s**	**3.72s**
Accuracy(max)	**96.30%**	**95.22%**	**97.37%**	**95.37%**

In practice, if considering running time, KNN and Logistic regression cost short times to finish classifications. Among all four models, KNN has the highest accuracy. When considering the features of data, SVM depends on features selections but not depends on the correlations between features. KNN not only depends on the feature selections but also correlations between features. Logistic regression neither depends on feature selections nor correlations between features.

Difficulty

Compared with other three models, CNN was more challenge to be constructed because we needed to decide the layers according to the input of the model.

Reference

https://en.wikipedia.org/wiki/File:Svm_separating_hyperplanes_(SVG).svg
https://www.mathworks.com/matlabcentral/fileexchange/63621-knn-classifier
https://www.quora.com/Why-is-logistic-regression-considered-a-linear-model
https://www.kaggle.com/primaryobjects/voicegender/data