DSH - Automatic Classification of Depression from Speech Variables

Daniel Gareev [1], Boghos Youssif[2, 3], Alex Borgognoni[5],

[1]0180226556, [2]0180701856 [3]0181077940
{daniel.gareev.001, boghos.youseef.001, & alex.borgognoni.001}@student.uni.lu

Abstract

Clinical depression is a major health concern and one of the prominent cause of mental disturbance worldwide. Currently, the diagnosis of depression, due to its complex clinical characterisation, is a difficult and time consuming task. In clinical practice, rating for depression depends on the categorical assessment of a set of specific symptoms and a clinical patient history. In recent years, the automatic detection of depression from the speech signal is getting an increased awareness. In this paper, we first classify whether a person is depressed based on the number of the speech and non-speech related features. Next, we try to find which features are more responsible for depression from speech and which classifiers give the most accurate results. Finally, we explore a gender effect in speech variables and depression levels by analysing each of the gender samples separately.

1 Introduction

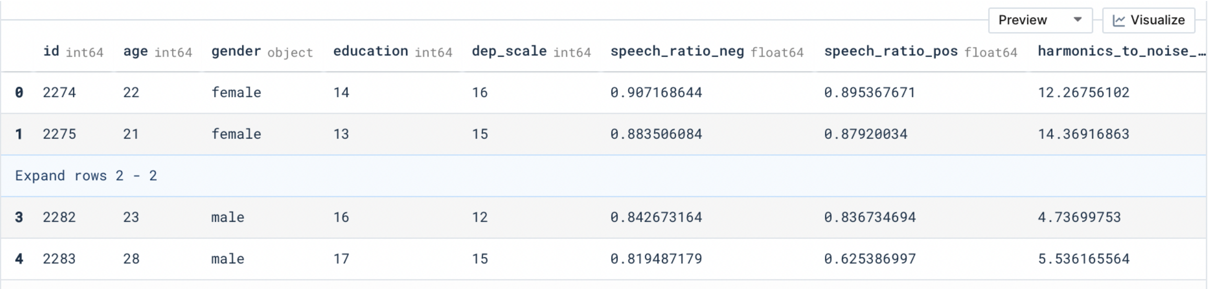
Depression is a mental disorder that presents state of low mood, negative thoughts, mental disturbance, difficulty in maintaining concentration and cognitive difficulties (Mantri et al., 2013). The World Health Organization (WHO) predicts that by 2030 depression will be the second leading cause of disability worldwide (Friedrich, 2017).

In clinical practice, rating for depression depends on the categorical assessment of a set of specific symptoms and a clinical patient history. However, many of the key symptoms, such as altered mood and motivation, are not physical in nature and assigning a categorical variable to them introduces bias in the assessment procedure (Cummins et al., 2015). Due to these difficulties, there is a need for an automated depression diagnostics to find a set of biological, physiological and behavioural properties linked to the depression to aid clinical assessment.

In recent years, the automatic detection of depression from the speech signal has been getting an increased awareness (Sturim et al., 2011) (Mantri et al., 2013) (Nilsonne, 1987) (Vázquez-Romero and Gallardo-Antolín, 2020) (Mantri et al., 2013). In this paper, we rely on the dataset of the study that analysed the acoustic speech variables of depression in a large sample of patients. In the study, recordings were made during recounting positive and negative events from the past experiences of patients. We first classify whether a person is depressed based on the number of the speech and non-speech related features. Next, we look at which features are more responsible for depression from speech and which classifiers give good results. By identifying the correlated features from speech one can save the life of a patient. Finally, we explore a gender effect in speech variables and depression levels.

2 Dataset

The dataset that we rely on in this study contains speech, non-speech features and clinical variables from participants of a clinical depression analysis study. In the study, participants were asked to perform a verbal task (recounting a positive, and negative event). The _pos and _neg tags indicate whether the feature is linked to the positive or negative event experienced by a patient. Speech recordings were made of 121 patients. The recordings were analyzed using a computer program(s) which extract acoustic parameters from the audio, such as the percent pause time, the voice fundamental frequency distribution, the rate of change



	id int64	age int64	gender object	education int64	dep_scale int64	speech_ratio_neg float64	speech_ratio_pos float64	harmonics_to_noise_…
0	2274	22	female	14	16	0.907168644	0.895367671	12.26756102
1	2275	21	female	13	15	0.883506084	0.87920034	14.36916863
Expand rows 2 - 2								
3	2282	23	male	16	12	0.842673164	0.836734694	4.73699753
4	2283	28	male	17	15	0.819487179	0.625386997	5.536165564

Figure 1: The sample of the dataset

of the voice fundamental frequency and the average speed of voice change. There are in total 105 speech features associated with each positive or negative events.

The and demographic and clinical variables of patients are included in the dataset. Each patient has an ID number, age, gender, education level. We assume that the only categorical variable is gender and the rest are numerical. The study also determines a depression scale of each of the patients. The patients are classified into a group with depression scale > 17 (clinically depressed patients) and ≤ 17 (controls). Therefore, there are 56 clinically depressed patients and 65 controls. The sample of the dataset can be seen in Figure 1.

3 Exploratory Data Analysis (EDA)

In this section, we explore the dataset in detail. Here, we demonstrate how we analyzed the dataset variables, data skewness, gender distribution, the correlation of depression with the demographic variables (such as age and education), as well as correlation between the variables.

3.1 Data Exploration

First, we checked the skewness of the variables by plotting the distribution plots (Figure 15 in the appendix. The dataset contains mostly adults in the age group between 20 to 40 years old.

The females made up over 75% of the total participants as can be seen in (Figure 2). The average age of the participants is 24 years. While for the female participants the average age is almost 24, the average age for male participants the average age is close to 26. We show the detailed distribution in the Table 1.

The Figure 16 (in the appendix) shows depression is at it's highest between the ages of 20 and 30. Therefore, people in the age between 20 to 30 years are more likely to be depressed than any other group.

Table 1: Gender Distribution

Gender	Number of Participants	Average Age
Male	94	23.7
Female	27	25.8
Total	121	24.1

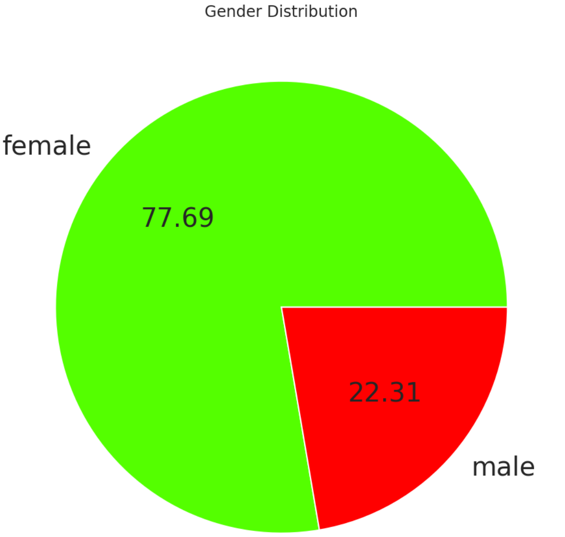


Figure 2: Gender distribution.

When looking at the depression by education, the Figure 17 (in the appendix) demonstrates that the depression tapers off at the higher end of the education spectrum. People are more likely to be depressed in the middle of the educational career. Moreover, the level of depression tends to decrease towards the end of the educational path.

Female participants are the most depressed in the ages of 18 to 21, while male participants are the most depressed in the ages of 27-29 as can be seen in Figure 18 (in the appendix).

In Figure 19 (in the appendix) we can observe that females are more depressed than males.

3.2 Feature Correlation

We created a feature correlation matrix to show the difference in depression with respect to some of the demographic variables (such as gender and age) and speech variables (Figure 20 in the appendix). We can see that the depression is the most correlated with one of the speech variables, but not with the demographic features.

Then, we created the correlation matrix for each of the genders (Figure 21 in the appendix). When we separated the genders, interesting results arose. We can see that while for male participants most of the variables are positively correlated, for the female participants the majority of values are negatively correlated.

4 Data Processing

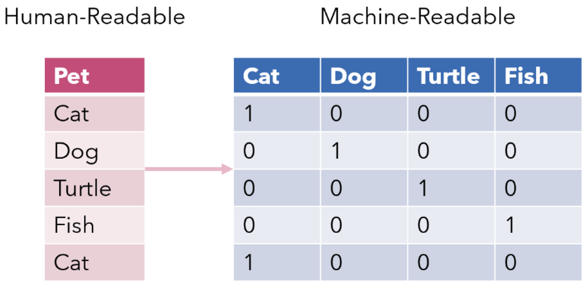
In this section, we cover data processing before we begin to train the models. Here, we describe in-detail how we approached the feature extraction, data cleaning, handling missing values, outlier detection, normalization and splitting data.

4.1 Feature Extraction

Our dataset contains text or categorical values (non-numerical values) in some of the columns. There are a few models can handle categorical values but most of the models expect numerical values to be handled in. It is important to understand various options for encoding categorical (non-numerical) variables into numerical as each method can affect a model in a variety of ways.

First, we converted the variable gender into binary with label encoding. Label encoding involves converting each value in a column to a number. It assigns all the unique values of the feature a positive integer value $(0, 1, ..)$ in an alphabetical order. Therefore, as only two genders are present in the dataset, male and female will be assigned values 0 and 1.

Next, we converted categorical variable education to one-hot encoding. We assume that in our dataset education represents a ranking, so it makes sense to create a dummy variable for each unique value of the categorical feature so that a model can make sense of that. The label encoding is a simple method, but the model can misinterpret the data ass having an order in it. One-hot encoding, otherwise known as dummy variables, is a method that helps to convert categorical variables into binary columns, where a 1 shows that a row belongs to that category. The Figure 3 shows how one-hot encoding works. However, even though this method eliminates the issue of order, it adds more dimensions to the dataset, which makes it hard optimize a model.



Human-Readable

Pet
Cat
Dog
Turtle
Fish
Cat

Machine-Readable

Cat	Dog	Turtle	Fish
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0

Figure 3: One-hot encoding. Figure source: (Ye,)

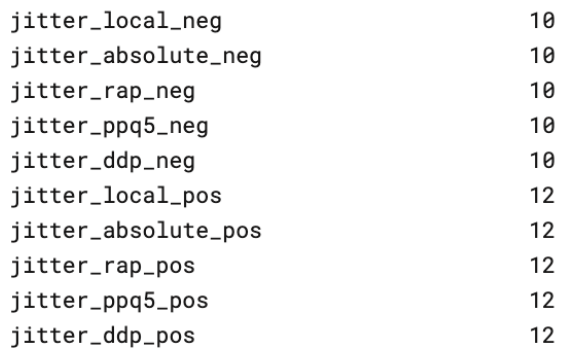
Next, we dropped the ID column of the participants as we do not need it for predicting. Then, we created a binary column that would indicate whether a person is depressed or not based on the depression (depression > 17 indicates that a patient is clinically depressed patients and ≤ 17 indicates that a patient is a control.

4.2 Handling Missing Values

Next, we had to explore if there are any missing values in the dataset. Some of the models, such as XGBoost can handle missing values. However, most of the models cannot process data with missing values and without doing imputation first.

We first checked if there are any missing values, which are represented with NaN or None. As we found out, we have ten columns with 10-12 missing values representing speech variables that are shown in Figure 4. The columns represent five kinds of jitter measurements, which are acoustic characteristics of voice signals. In Figure 5, we can observe individual rows with the missing values. The missing values represent only 0.4% of the dataset. The values are most likely missing because the speech has not been decoded for these features. This means that the nature of the missing data is related to the observed data but not the missing data (data is missing at random). As we don't have that many participants, we want to avoid dropping rows with missing values. Therefore, it makes sense to impute the value based on the other values in that column and row rather than just leaving them as NA's.

We used Multiple Imputation using MICE (Multiple Imputation by Chained Equations) to



```
jitter_local_neg                          10
jitter_absolute_neg                       10
jitter_rap_neg                            10
jitter_ppq5_neg                           10
jitter_ddp_neg                            10
jitter_local_pos                          12
jitter_absolute_pos                       12
jitter_rap_pos                            12
jitter_ppq5_pos                           12
jitter_ddp_pos                            12
```

Figure 4: Missing values.

impute the missing data (Azur et al., 2011). Multiple imputation is a process where the missing values are filled multiple times by running multiple regression models for each missing value conditionally depending on the observed (non-missing) values. It works with the assumption that the missing data are missing at random. Multiple imputation has a set of the pros over naive single imputation methods (e.g., mean, median) as it also enables to capture statistical variance in data.

4.3 Outlier Detection

Outliers are the datapoints that are significantly distant from other observations or overall pattern present in a dataset. These extreme values can mainly occur due to errors in measurement or misinterpretation in data collection. They might impact the model performance and accuracy, and it's important to handle them before moving forward to the model training.

We first explored demographic variables, such as age of the participants before moving to other features. In the Figure 6, the age variable is visualized as a box plot. This allows us to clearly see two outlier values that are specially distant from the prevalent age distribution. We found out that 75% of the participants are up to 26 years old, but the maximum age is 75. We decided on a limit of 40 years old, and dropped all outliers that are older than that. After dropping the outlier values, the dataset had 119 rows in total.

Some of the models have an assumption that the feature columns are independent of each other. It can also affect how the model generalize the problems, which can have an impact on how it performs in the production environment. Multicollinearity occurs when two or more explanatory variables in a dataset are linearly related. Therefore, when two features have high correlation, we can drop one of the two features as they have almost the same effect on the dependent variable.

We checked the correlations between our numerical features and explored which features are highly correlated. We used a correlation matrix and convert the correlations to their absolute values in order to deal with negative correlations. Then, we dropped the columns with the correlation of 95% and above. In Figure 7, we can observe the columns with the highest correlation.

Next, we checked outliers in the speech features. As we have many features, we needed to have an automated way to determine whether or not the features contain skewed distributions and if they contain any outliers. We used Isolation Forests for detecting outliers. It is an unsupervised algorithm based on trees. It isolates anomalies from the rest of the observations by randomly splitting the dataset region into smaller pieces (or many isolation trees).

4.4 Normalization

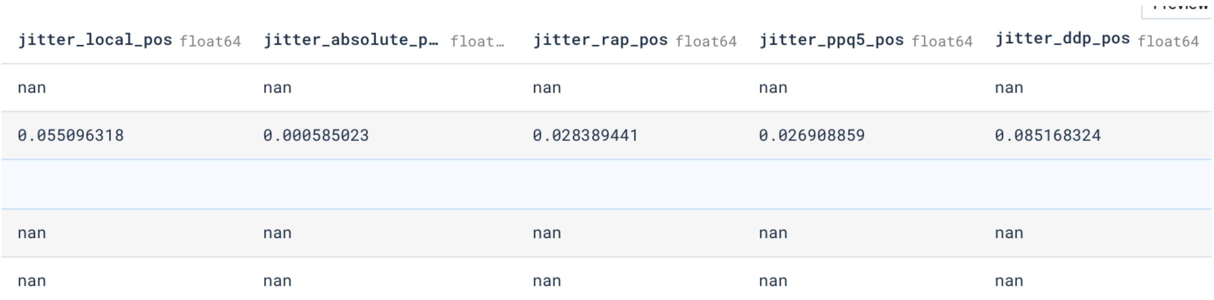
Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values.

Most classifiers use some form of a distance calculation and each numeric feature tends to have different ranges. Scaling these features to a common scale helps to ensure that each feature's contribution is weighted proportionally.

In our dataset, many of the speech features have different ranges. Therefore, we scaled these numerical columns to a common scale. We accomplished this by changing the range of each features to 0 till 1. We can observe the sample of the dataset after normalization in the Figure 8.

4.5 Splitting Data

As we want to see how models will perform with demographic, clinical and speech variables and only speech variables, we separate the initial dataset in two separate ones. The first dataset contains only speech features. The second dataset contains both speech and non-speech features (i.e., demographic and clinical). To understand model performance, dividing the dataset into a training set and a test set is a good strategy. We split both



jitter_local_pos float64	jitter_absolute_p… float…	jitter_rap_pos float64	jitter_ppq5_pos float64	jitter_ddp_pos float64
nan	nan	nan	nan	nan
0.055096318	0.000585023	0.028389441	0.026908859	0.085168324
nan	nan	nan	nan	nan
nan	nan	nan	nan	nan

Figure 5: Missing rows.

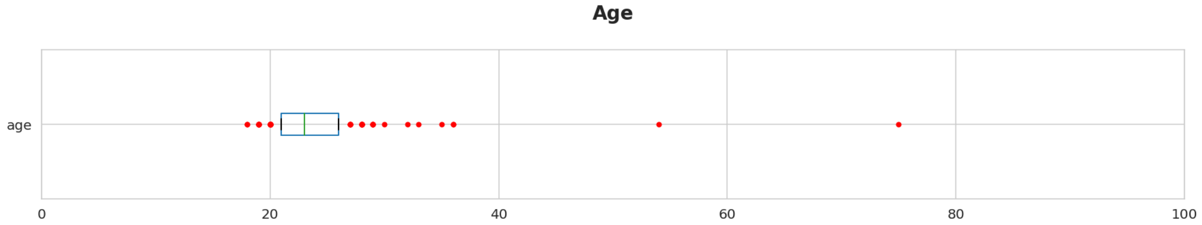


Figure 6: Age box-plot.

datasets on training and testing sets, with the testing dataset being 20%.

5 Data Modelling

5.1 Baseline Model: Support Vector Machine (SVM)

As baseline model, we use a Support vector machine, a supervised machine learning algorithm which proves to be very effective for binary classification tasks, which is essentially what we are trying to do, we want to predict if a person is depressed (1) or not (0). The reason we chose this approach as baseline model is because it yields the best accuracy score with the lowest training times. More details on this in the next section where we talk about the individual accuracy an training time results of every model.

5.1.1 Modeling Full Dataset

To build our model, we use the SVM module from the scikit library. Next, we create a support vector classifier object by specifying that we want a linear kernel. Then, we fit the model on the train set and perform prediction on the test set. We first fit the full dataset (both speech and non-speech features). Also, we measure the model's training speed. In the first iteration, the SVM model gave us an accuracy of about 62% with a training time of 0.004 seconds.

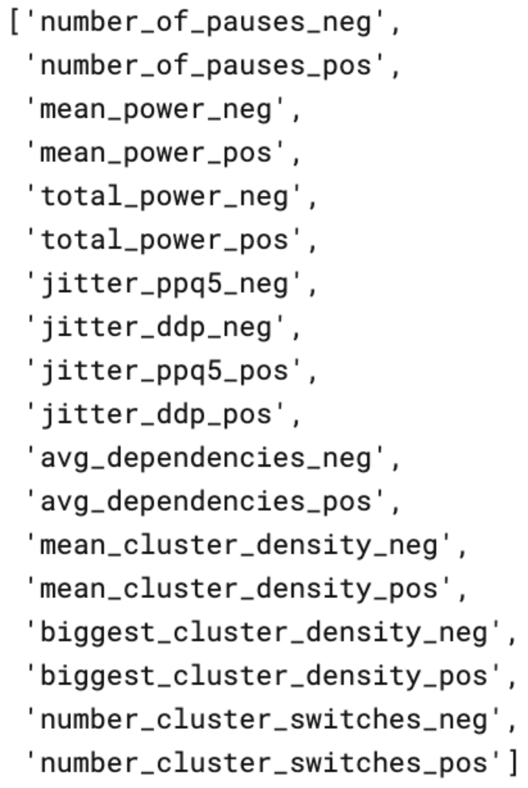
Once having fitted our linear SVM it is possible to access the classifier coefficients on the trained model. We can determine feature importance by comparing the size of these coefficients to each other. By looking at the SVM coefficients we can check the most important features and remove the not so important ones (which have less variance). As we can see in Figure 9, *conjuction_rate_pos* and *verb_rate_neg* are the most contributing features. These variables deal with the frequency of the verbs in the speech and the conjunction rate in speech.

5.1.2 Optimizing Model

To optimize the results of the model, we used the following two techniques:

1. K-fold cross validation: This technique randomly splits the the dataset into K groups or so-called folds. 1 fold is then used as test set and the remaining 4 are used as training set. This is then repeated K times until every fold has been used exactly once as test set

2. Hyper-parameter tuning: This method allows to find the best parameters for your model (e.g. find the ones that give the model the best accuracy). We specified a testing range for all of the parameters that the SVM model takes and try out all the different permutations in order to find the best combination.



```
['number_of_pauses_neg',
 'number_of_pauses_pos',
 'mean_power_neg',
 'mean_power_pos',
 'total_power_neg',
 'total_power_pos',
 'jitter_ppq5_neg',
 'jitter_ddp_neg',
 'jitter_ppq5_pos',
 'jitter_ddp_pos',
 'avg_dependencies_neg',
 'avg_dependencies_pos',
 'mean_cluster_density_neg',
 'mean_cluster_density_pos',
 'biggest_cluster_density_neg',
 'biggest_cluster_density_pos',
 'number_cluster_switches_neg',
 'number_cluster_switches_pos']
```

Figure 7: Correlated columns.

Cross-fold validation as well as hyperparameter-tuning can be implemented simultaneously using the *GridSearchCV()* function from the sklearn library. We also specified that we want to optimize the precision and recall of the model.

After optimizing the model, the accuracy remained at 62%. We can clearly see that there is some room for improvement. The precision and recall of the model were at 63% and 62%. This means we have to use yet another approach to improve the results. Next, we will try to extract other features from the dataset instead of focusing on improving the baseline model.

5.2 Alternative Models

We also experimented with other models to see which one would yield the best results. The models that we used are XGBoost/Decision Tree Classifier , K-NN classifier, Deep-learning with Keras (The Sequential Model), Naive Bayes. Here, we provide a short overview of these models.

5.2.1 XGBoost

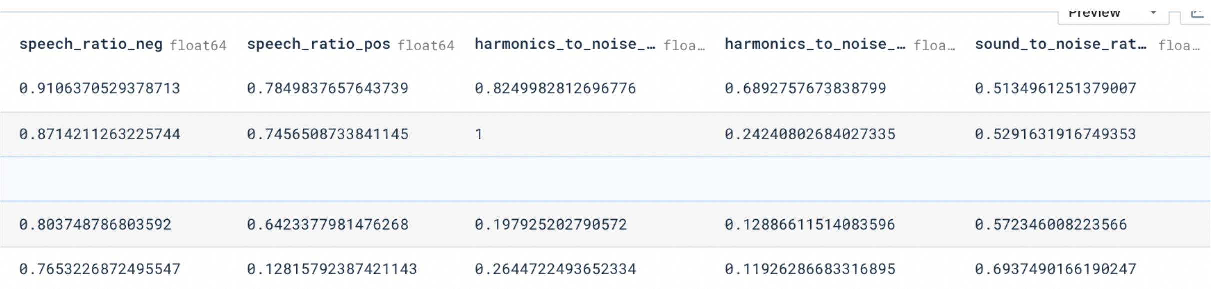
XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost has become one of the most used tools in machine learning. It consist of an ensemble of decision trees, where each new tree depends on the evaluation of the previous one. This sequential way of adding classifiers its called boosting, but unlike traditional boosting, with XGBoost it is possible to run it in parallel, since in the construction of the trees each branch is trained independently.

5.2.2 K-nearest neighbors

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm. It works by classifying objects into groups based on the object's nearest neighbors (k to be exact) in a feature space (for example in 2D this would be a plane and distance would be measured with euclidian distance usually). In terms of implementing this approach, first we import the function from the SKLearn library and create the model by setting the of neighbors to 3. We chose 3 since it seems to be a good starting value since it is the next odd number after 1 (to ensure it is always clear into which category it is classified), and 1 is too much of a small number to really say if the newly added data should belong to that class. For recap, this means that if at least 2 out of the 3 nearest neighbors of the point we want to predict are depressed, our new data point will be categorized as 'depressed'. Also, like mentioned previously, we use the euclidian distance as distance evaluation metric. The euclidian distance (in 2 dimensional space) is defined as in figure 11:

5.2.3 Naive Bayes

The naive bayes model is a classification technique based on Bayes theorem, which describes the probability of an event, based on prior knowledge of conditions that might be related to the event. For example, we know that obese people are more likely to have a heart attack since we also know that they (probably) had a high fat and sugar diet, which is proven to clog vital arteries. The 'naive' part relates to the model's property to assume that all the features are independent, which is pretty rare in real-life. Take for example an apple: it's red, round, and about 3 inches



speech_ratio_neg float64	speech_ratio_pos float64	harmonics_to_noise_… floa…	harmonics_to_noise_… floa…	sound_to_noise_rat… floa…
0.9106370529378713	0.7849837657643739	0.8249982812696776	0.6892757673838799	0.5134961251379007
0.8714211263225744	0.7456508733841145	1	0.24240802684027335	0.5291631916749353
0.803748786803592	0.6423377981476268	0.197925202790572	0.12886611514083596	0.572346008223566
0.7653226872495547	0.12815792387421143	0.2644722493652334	0.11926286683316895	0.6937490166190247

Figure 8: A sample of the dataset after normalization.

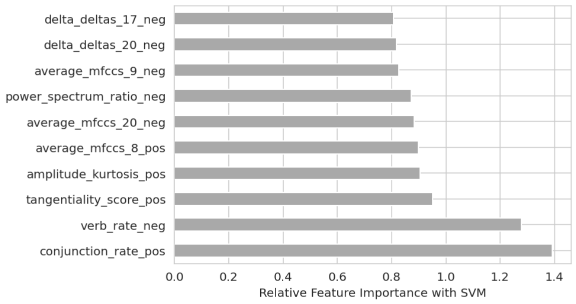
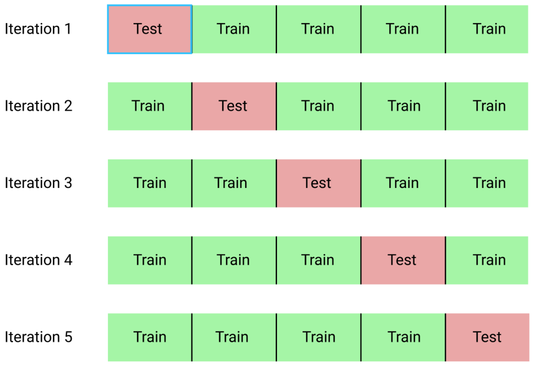


Figure 9: Feature importance for the SVM.



Iteration 1	Test	Train	Train	Train	Train
Iteration 2	Train	Test	Train	Train	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Train	Train	Test	Train
Iteration 5	Train	Train	Train	Train	Test

Figure 10: K-fold cross-validation. Figure source: (static.oschina.net,)

in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’. Again, this model was simply implemented by importing the function from the Sklearn Library.

5.2.4 Keras Deep-learning

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. Deep learning is a subfield of machine learning and is inspired by the structure and function of the human brain. It tries to mimic the network of neurons which allows us humans to learn new skills, hence the name ’artificial neural network’. The main advantage of neural networks compared to regular machine learning algorithms is scalability and performance. Where as machine learning models have a limit in performance after a certain amount of data provided, i.e. the model will not improve further although it’s getting more training data, Neural networks can be fed an unlimited amount of data and still further improve their results. The way a neural network is implemented is just as quick and easy as a regular machine learning model. We simply import the sequential model from the Keras library, create it and fit it with our training data. The only difference is training time, which is considerably larger.

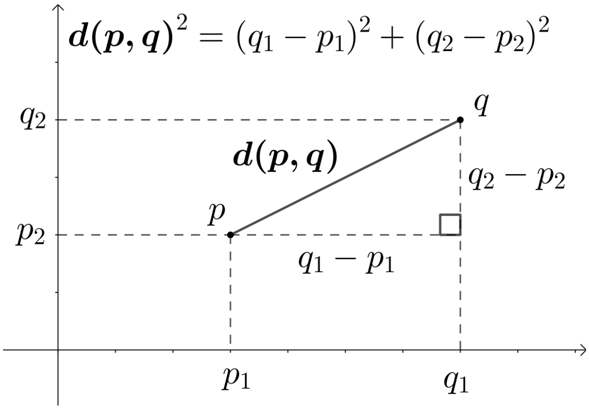


Figure 11: Euclidian distance. Figure source: (Wikipedia,)

5.2.5 Decision tree

Decision trees use a tree like structure to model decisions and their possible outcomes. They are a non-parametric supervised learning method used for classification and regression. Figure 12 shows a simple example for a decision tree on whether or not to go and play tennis.

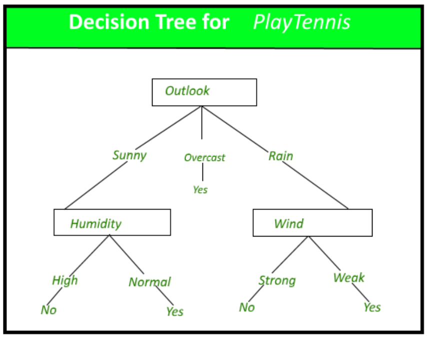


Figure 12: Decision tree example. Figure source: (GeeksForGeeks,)

5.2.6 Alternative model results

Training was performed on the entire dataset in order to be able to compare the training time and accuracy scores. We show the results of the models in the Table 2.

Table 2: Other Models Results

Model	Accuracy	Training time (in seconds)
KNN	0.58	0.0008
Keras NN	0.58	8.2000
Naive Bayes	0.625	0.0024
Decision-tree	0.33	0.0065
XGBoost	0.5	0.1863

Comparing the results of the other models, we can claim that the results are optimized with the SVM model. For the task of depression classification, it gives the best results compared to the other models we have explored.

5.2.7 Modeling Speech Features

By removing the non-speech features from the dataset we want to see if the accuracy of our model would increase, i.e. if speech features alone have more predictive power than combining all the features. But, after fitting the new model, the results obtained actually slightly decreased to 58%, but with the trade-off of having less feature dimensions in the dataset. The precision and recall of the model were at 57% and 66%.

5.2.8 Modelling Speech Features with Gender Differences

Depression might be experienced differently by male and female participants. The way participants recount their experiences might also be different for each of the gender. Moreover, some of the speech variables are related to the tone, timbre and frequency of the sounds, which might differ for each of the genders.

Up until now we trained the model without separating genders, i.e. using the dataset consisting of samples of both male and female. As over 75% of the participants in the study are females, we decided to split it into two distinct datasets, one for each gender with the speech features only. After splitting the dataset, we only have 27 male participants and 92 female participants. We then create two separate SVM models and fit the female and male datasets. For predicting the depression for each of the genders, we will not detect outliers and remove highly correlated features, as we don't have that many observations for each of the participants.

Predicting the results for male participants gives us an accuracy of around 33%. The accuracy for the male participants is very low. This is most likely due to the fact that we have very few observations of the male subjects. The *average_mfccs_9_neg*, and *adjective_rate_pos* are the most contributing features for the male participants (Figure 13).

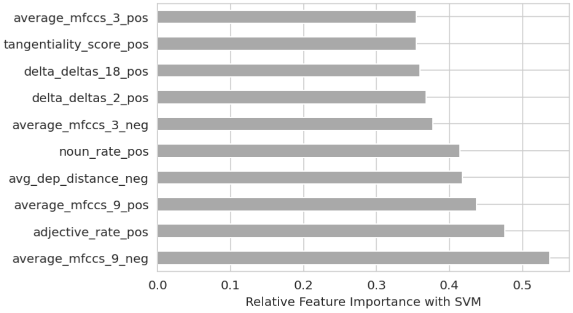


Figure 13: Relative Feature Importance for Males

Training on the female dataset gives us better results than male. Here we get an accuracy score of about 58%. The *verb_rate_neg*, and *conjuction_rate_pos* are the most contributing features for the female participants (Figure 14).

We then use these 10 most important features and create an individual dataset with them. Training the model with this dataset yields the best result with 68% accuracy. The precise results can be found in Section 5.3 below.

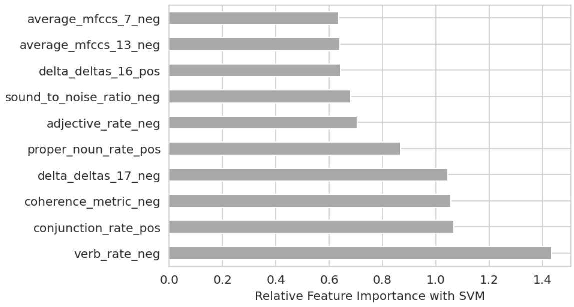


Figure 14: Relative Feature Importance for Females

5.3 Results

The results of all our methods can be found in the Table 3.

Table 3: Final model results

Model	precision	recall	f1-score	support
SVM (females, 10 best features)	0.68	0.68	0.68	19
SVM (males, 10 best features)	0.33	0.33	0.33	6
SVM (complete dataset)	0.63	0.62	0.62	24
SVM (complete dataset, w/o param. opt.)	0.63	0.58	0.33	24

5.4 Discussion

In this paper, we have presented the approach to classify whether a participant is depressed using a set of demographic and speech features. Next, to improve the model, we can try to experiment with different combinations of clinical, demographic and speech features. For example, there might be a certain correlation of specific speech variable to the age of the participant. Next, we can try to improve the performance by using different pre-processing and feature extraction methods (e.g., finding the most contributing speech features, removing collinear speech features from a specific gender) Further, it collecting a larger dataset to improve the accuracy and generalizability of the model will help a lot to optimize the model. Moreover, considering positive and negative events separately for both genders can add a boost to the model performance and make it less biased. First, it will allow to make a distinction on depressed and non-depressed people experience negative and positive events (e.g., a depressed person can experience negative events differently than positive ones). We also saw that certain speech variables of the positive events have a linear dependence with the same speech variable of the negative events, which means they do not contribute to the model performance. By splitting the data into separate events, it allows to handle multicollinearity easily. Moreover, as males and females can experience positive and negative events differently, it's important to consider this approach with separated genders.

6 Conclusion

At the end of our work, two general trends emerged as the most important factors in determining the accuracy of a machine learning model:

- The size of the data set used.
- The type of model used.

Our data set does not fulfill the first factor, but we were able to freely choose the machine learning model that we thought was best based on experimental results (sections 5.2 & 5.3), and that is the SVM model.

Another important aspect of machine learning that we learned was the pre-processing of the dataset. Datasets are not created ready to be plugged in into a model, they sometimes contain NaN values, inconsistencies (e.g., outliers), and values that are not suitable for the model training (e.g. the gender, which a string object).

Finally, the results that we had with other models (**Table** 2 in section 5.2) were varying between 33% (the lowest score was with the decision-tree model) and 62.5% (the highest score with the Naive Bayes model). Our base model, however, outperformed all of those results with the highest accuracy rating of 68%. But that was only when it was trained on the subset of our dataset with female participants only. Females comprised 77.69% of our dataset and since there were difference between the feature correlation between genders, it makes sense that our model achieved higher training accuracy with only female samples.

7 Table of collaborations

In this section, we briefly describe what each of us did in the group for the assigned project (Table 4).

Table 4: Table of collaborations

Name	Task
Daniel	In the notebook, I did Exploratory Data Analysis (EDA), Data Preprocessing, Data Modeling (splitting the data, outlier detection and Support Vector Machine (SVM)) and Gender Difference. Boghos also contributed to Exploratory Data Analysis (EDA) with several graphs and Alex contributed to Gender Difference. In the final report, I wrote Abstract, Introduction, Data Processing and Discussion. I also helped Boghos with Dataset, Exploratory Data Analysis (EDA) section and Alex with Modelling section.
Boghos	Data exploration (I wrote sections 2 & 3 & conclusion in addition to the figures which I generated using the code I wrote in the notebook plus the notebook section of dataset presentation) I also trained and tested the Bayesian model with the decision tree model in the notebook as well as created the table of final results comparing all the models in the notebook.
Alex	In this project, my work was mostly on model presentation and the initial implementation. I also helped optimizing our baseline SVM model with regards to genders. Other than that, I helped everywhere my help was needed to finish things. In this document, I wrote Sections 5.1, 5.2 and 5.3 almost entirely (with the exception of a few corrections made by Daniel).

References

[Azur et al., 2011] Azur, M. J., Stuart, E. A., Frangakis, C., and Leaf, P. J. (2011). Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research*, 20(1):40–49.

[Cummins et al., 2015] Cummins, N., Scherer, S., Krajewski, J., Schnieder, S., Epps, J., and Quatieri, T. F. (2015). A review of depression and suicide risk assessment using speech analysis. *Speech Communication*, 71:10–49.

[Friedrich, 2017] Friedrich, M. J. (2017). Depression is the leading cause of disability around the world. *Jama*, 317(15):1517–1517.

[GeeksForGeeks,] GeeksForGeeks. Decision tree example. https://www.geeksforgeeks.org/decision-tree/.

[Mantri et al., 2013] Mantri, S., Agrawal, P., Dorle, S. S., Patil, D., and Wadhai, V. M. (2013). Clinical depression analysis using speech features. In *2013 6th International Conference on Emerging Trends in Engineering and Technology*, pages 111–112. IEEE.

[Nilsonne, 1987] Nilsonne, Å. (1987). Acoustic analysis of speech variables during depression and after improvement. *Acta Psychiatrica Scandinavica*, 76(3):235–245.

[static.oschina.net,] static.oschina.net. K-fold cross validation example. https://my.oschina.net/Bettyty/blog/751627.

[Sturim et al., 2011] Sturim, D., Torres-Carrasquillo, P. A., Quatieri, T. F., Malyska, N., and McCree, A. (2011). Automatic detection of depression in speech using gaussian mixture modeling with factor analysis. In *Twelfth Annual Conference of the International Speech Communication Association*.

[Vázquez-Romero and Gallardo-Antolín, 2020] Vázquez-Romero, A. and Gallardo-Antolín, A. (2020). Automatic detection of depression in speech using ensemble convolutional neural networks. *Entropy*, 22(6):688.

[Wikipedia,] Wikipedia. Euclidian distance example. https://en.wikipedia.org/wiki/Euclidean_distance.

[Ye,] Ye, A. Stop one-hot encoding your categorical variables. https://radiant-brushlands-42789.herokuapp.com/towardsdatascience.com/stop-one-hot-encoding-your-categorical-variables-bbb0fba89809.

Appendix

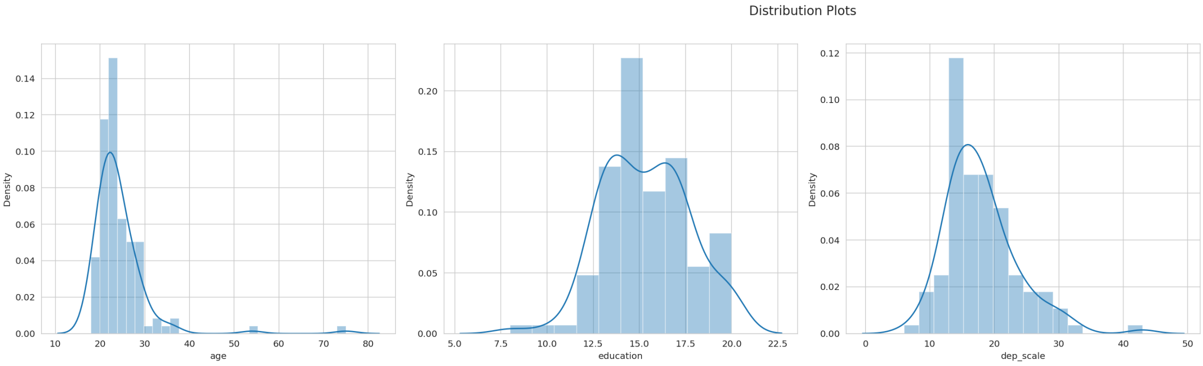


Figure 15: Distribution plots

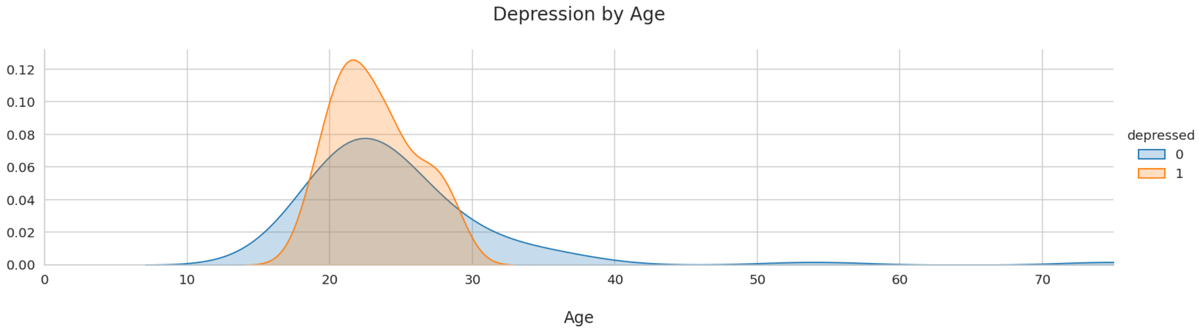


Figure 16: Depression by age.

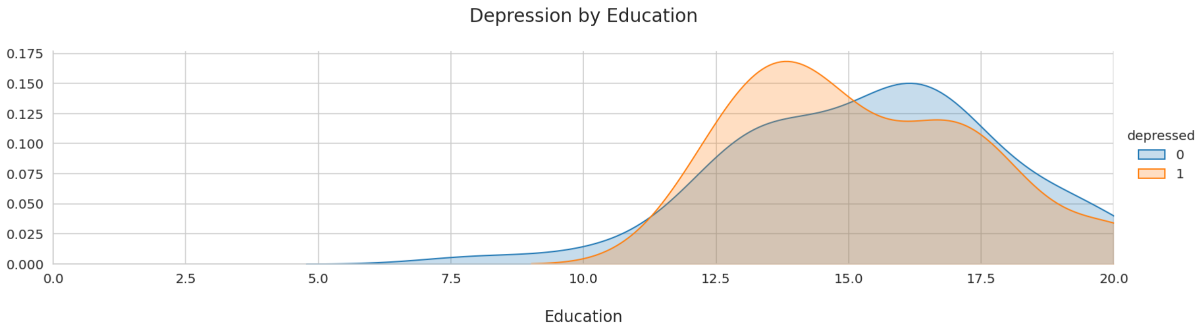


Figure 17: Depression by education.

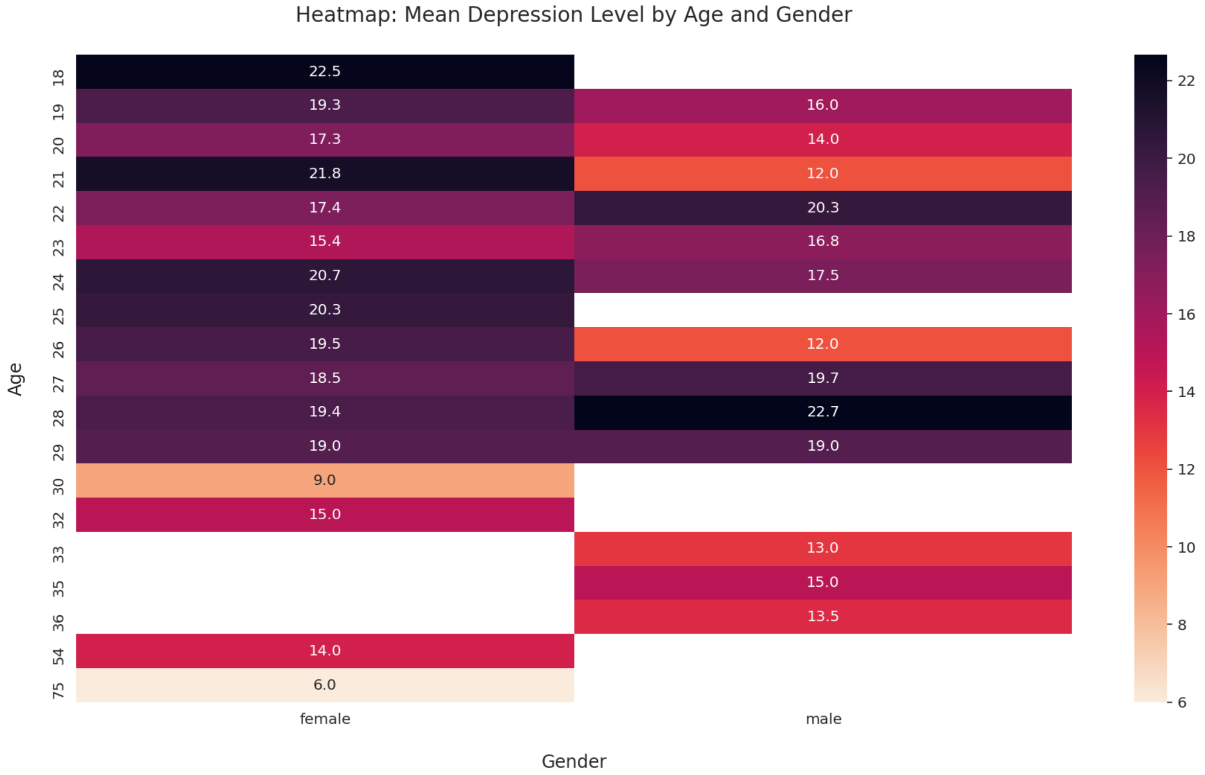


Figure 18: Mean depression level by age and gender.

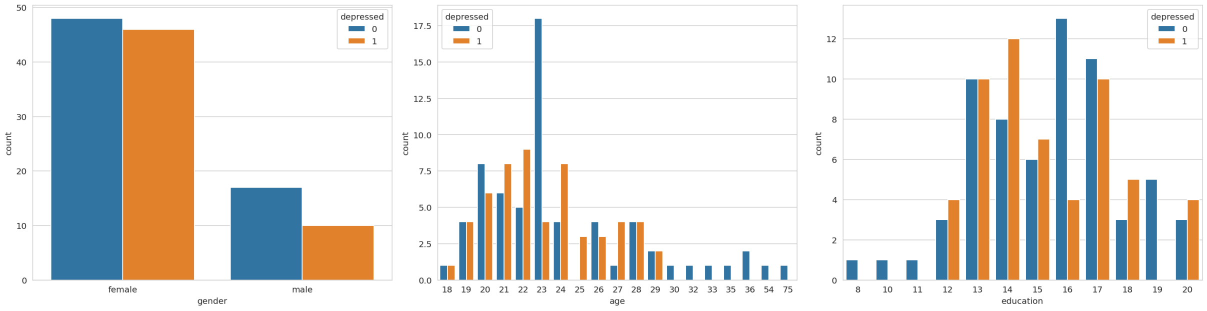


Figure 19: Pair plots.

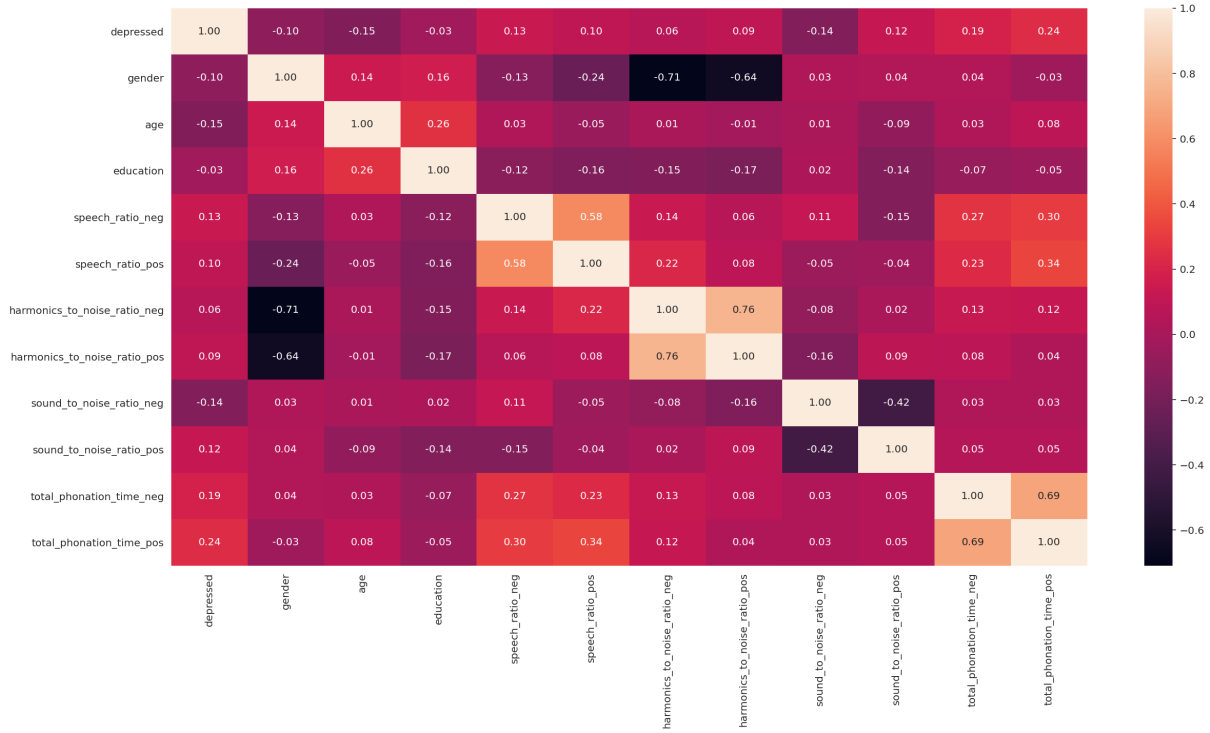


Figure 20: Feature correlation matrix.

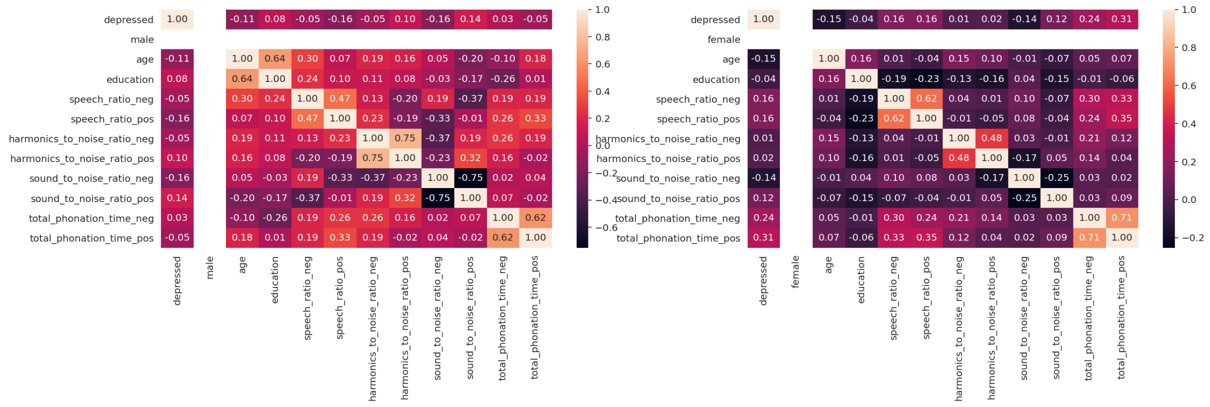


Figure 21: Feature correlation matrix by gender.