# Assignment 3

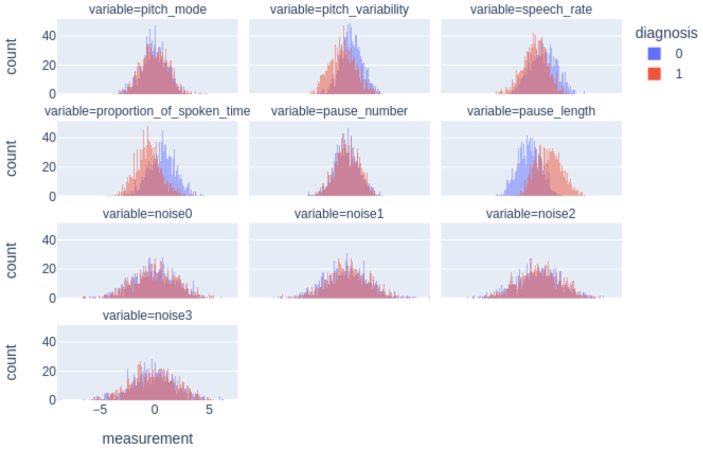
Márton Kardos

October 2022

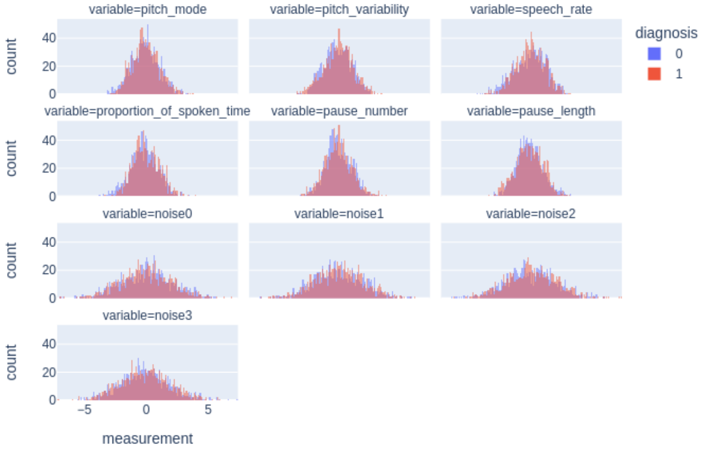
# Simulation

## Simulated data

According to the instructions in the assignments description, I generated two simulated datasets of matched group sizes of 100 with 10 trials, one of them informed by meta analytic research in the field, the other randomly initialized. In order to check whether the simulated data conforms to my expectations, I plotted the informed data set.



as well as the random data set.



Strong effects reported in the literature can be clearly observed on the first graph, while the two groups look the same in the second, the simulation process yielded realistic values.

## Data budgeting

I held out 10% of the participants in order to evaluate model performance after training.

## Classifier model

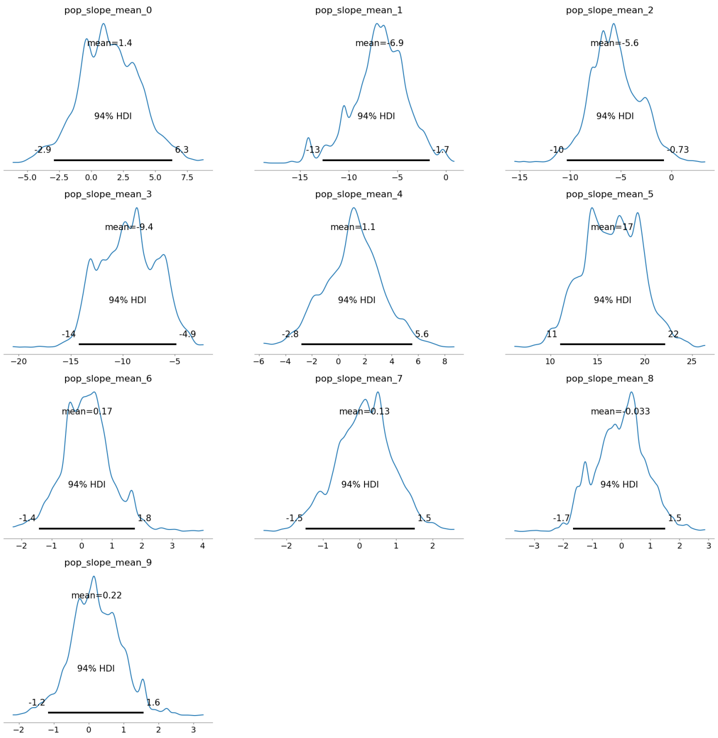
### Model specification

For classification I used a hierarchical Bayesian logistic regression model, that assumes the diagnosis generation process to be the following:

1. Draw a population level mean for intercepts, with prior $\mu_{\beta_0} \sim \mathcal{N}(0, 10)$
2. Draw population level standard deviation for intercepts, with prior $\sigma_{\beta_0} \sim \mathcal{N}(0, 10)$
3. For each participant $j$:
    - (a) Draw an intercept $\beta_{0j} \sim \mathcal{N}(\mu_{\beta_0}, \sigma_{\beta_0})$
4. For each feature $i$:
    - (a) Draw a population level mean slope, with prior $\mu_{\beta_i} \sim \mathcal{N}(0, 10)$
    - (b) Draw population level standard deviation for intercepts, with prior $\sigma_{\beta_i} \sim \mathcal{N}(0, 10)$
    - (c) For each participant $j$:
        - i. Draw a slope $\beta_{ij} \sim \mathcal{N}(\mu_{\beta_i}, \sigma_{\beta_i})$
5. For each observation $k$:
    - (a) Compute probability $logit(p_k) = \beta_{0j} + \sum_{i=1}^{N}(\beta_{ij} \cdot X_{ki})$,
      where $N$ is the number of features and $X$ is the feature matrix.
    - (b) Draw outcome $y \sim Bernoulli(p_k)$

### Informed sample

I took a posterior sample from the model using the informed data set, and investigated feature importances by plotting densities the population level mean posterior slope sample.



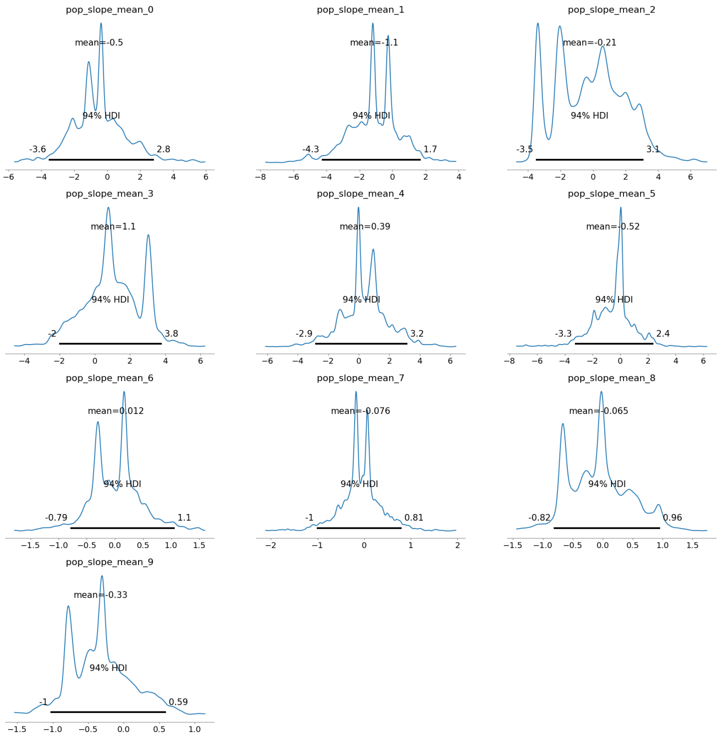
Sampling has only yielded significant slopes for pitch variability, speech rate, proportion of spoken time and pause length. Which is in line with our expectations.

Larger simulated effect sizes also yield larger posterior slopes. None of the noise variables had significant slopes, which all goes to show that the model captured the relations well in the data.

To see how much predictive power the model has I tested it on the held out part of the data set, and found that it performed way better than chance level at classification. The model's accuracy was 0.875, with recall of 0.93, precision of 0.86, and F1 score of 0.98.

### Uninformed sample

To test that there was no data leakage in my pipeline, and that model doesn't capture non-existent effects I fitted the model on the uninformed samples as well.
To assess feature importances I plotted the densities of the population level estimates once again.



It can clearly be observed that none of the features has significant slope values attached to them, meaning that the model did not capture spurious effects.
Testing on the held out data also reveals near-chance-level performance (*Accuracy*=0.485, *Recall*=0.55, *Precision*=0.49, *F1*=0.49)

## Alternative model

Since I had suspicions about whether hierarchical Bayesian modelling actually gives us a benefit here, I decided to also train and test a frequentist Logistic regression model with Ridge penalty, as it has a closed form solver which trains orders of magnitudes faster than the time it takes to explore the posterior even for state-of-the-art samplers.

The model performed almost identically with accuracy of 0.92 (*Recall*=0.93, *Precision*=0.93, *F1*=0.93). Based on this I conclude that reason the model performed well is either due to the fact that this problem is well-suited for logistic regression, or because of the regularizing power of priors.

Since I can have all these benefits with models that can be trained much more efficiently, I decided to proceed with the frequentist model when analyzing empirical data.

# Analysis

## Empirical data

Before any cleaning the dataset had 394 features and 1889 observations. The data is monolingual, all participants spoke the same language. Genders were roughly matched, with 42.8% of observations coming from female, and 57.2% coming from male participants. Class labels were also fairly balanced with 47.6% of observations coming from Schizophrenic patients, and 52.4% from controls.

I Decided to not use gender and language as predictors, which left me with 391 predictors. I decided to split the data set into a training set (90% of participants) that I can use for model selection, and a holdout set (10% of participants), with which the selected model can be evaluated.

## Model selection

I chose 5 candidate models for selection, these being:

1. Forest of 100 randomized trees.

2. Ridge Penalized Logistic Regression with penalty term 1.0 *(identical to Bayesian model with normal priors)*

3. Linear Kernel Support Vector Machine

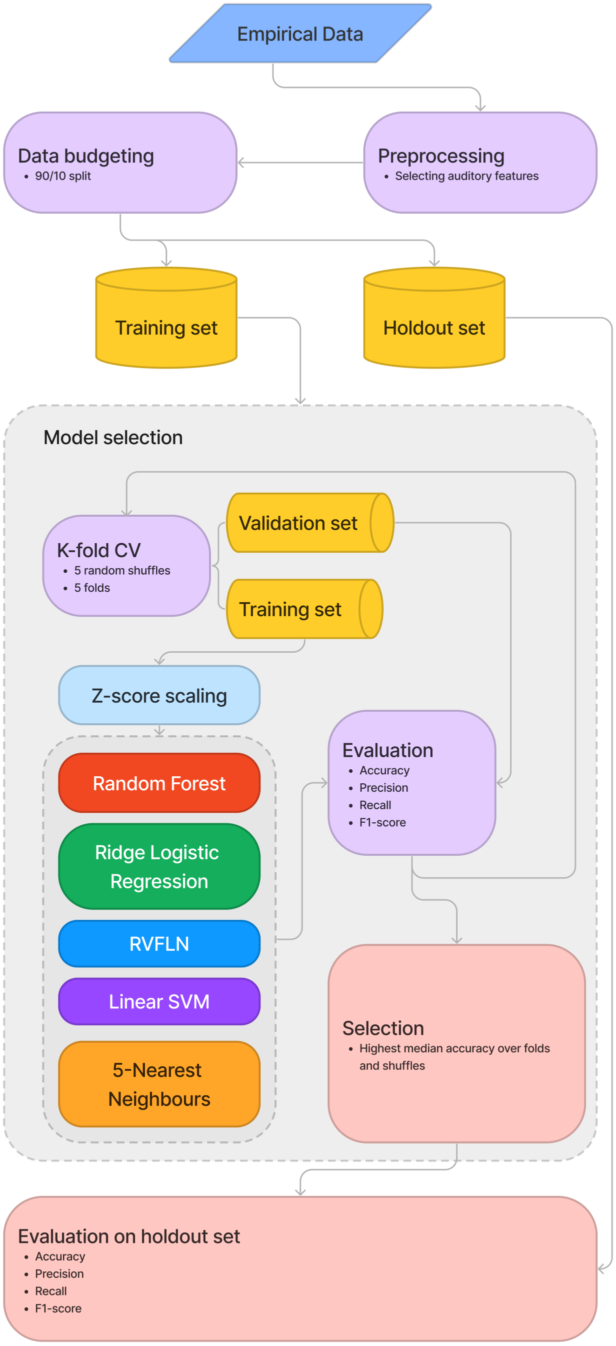
4. 5-nearest Neighbours Classifier

5. Random Vector Functional Link Network with 15 enhancement nodes, ridge penalty term of 1.0 and leaky rectified linear unit as its activation function.

For most models I used their implementation in the scikit-learn Python package.
Since RVFLN does not have a canonical sklearn implementation, I used mine (https://github.com/x-tabdeveloping/rvfln). I decided to add RVFLN into the mix, as research shows that in certain tasks they can achieve close to state-of-the-art deep neural network performance, while needing less tuning, and being generally less prone to overfitting. They might perform better than traditional models as they introduce non-linearities, but due to the weights being random the model does not overfit nearly as much as neural networks would.

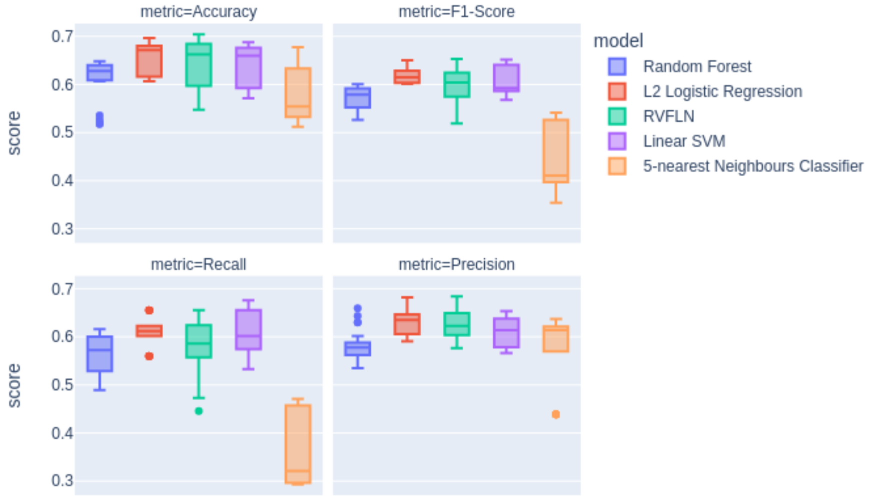
Since the data is quite feature-rich, I considered feature selection, but since most models are either penalized or have feature selection built in (Random Forest), I decided to skip that step. After quick experimentation I also found that it does not give much benefit in model performance.

I added a Z-score scaler to the pipeline, as it grants better convergence for models like logistic regression or SVM.
To select the best model I took the model with the highest median accuracy across 5 folds over 5 random shuffles of the training set.


Empirical Data
Data budgeting
90/10 split
Preprocessing
Selecting auditory features
Training set
Holdout set
Model selection
K-fold CV
5 random shuffles
5 folds
Validation set
Training set
Z-score scaling
Random Forest
Ridge Logistic Regression
RVFLN
Linear SVM
5-Nearest Neighbours
Evaluation
Accuracy
Precision
Recall
F1-score
Selection
Highest median accuracy over folds and shuffles
Evaluation on holdout set
Accuracy
Precision
Recall
F1-score

## Results

I plotted the performance of the candidate models against each other on multiple metrics over all folds and shuffles.



Logistic regression with ridge penalty was chosen as the best model, due to it having the highest median accuracy on the validation set.
All models were similar in performance on all metrics except for the 5-nearest Neighbours model. This is to be expected, as a higher-dimensional feature set significantly inflates euclidean distances. Nearest neighbours vote would probably perform better, if feature selection was part of the pipeline.

The selected model had 60.6% accuracy on the holdout set, which is above chance level (*Accuracy*=0.5), but is not particularly great. The model also has low recall (0.488), indicating that it has a hard time recognizing schizophrenic patients.
In conclusion, the model should not be trusted with clinical assessment.