

# The Genre Factor

## Project Presentation - ML Seminar 2023

---

Henry Krämerkämper

Christopher Breitfeld

13.07.2023

Technische Universität Dortmund

## Genre Classification

- Based on a variety of the tracks' features
- Dataset: Thousands of songs with diverse genres
- Challenge: Developing an accurate classification model

Feature	Value
Artist	Gorillaz
Url_spotify	<a href="https://open.spotify...">https://open.spotify...</a>
Track	Feel Good Inc.
Album	Demon Days
Album_type	album
Uri	spotify:track:0d28khcov6AiegS...
Danceability	0.818
Energy	0.705
Key	6.0
Loudness	-6.679
Speechiness	0.177
Acousticness	0.00836
Instrumentalness	0.00233
Liveness	0.613
Valence	0.772
Tempo	138.559
Duration_ms	222640.0
Url_youtube	<a href="https://www.youtube...">https://www.youtube...</a>
Title	Gorillaz - Feel Good Inc. (Official...
Channel	Gorillaz
Views	693555221.0
Likes	6220896.0
Comments	169907.0
Description	Official HD Video for Gorillaz'...
Licensed	True
official_video	True
Stream	1040234854.0

# Description of the Data Set

## Dataset from Kaggle: Spotify and YouTube

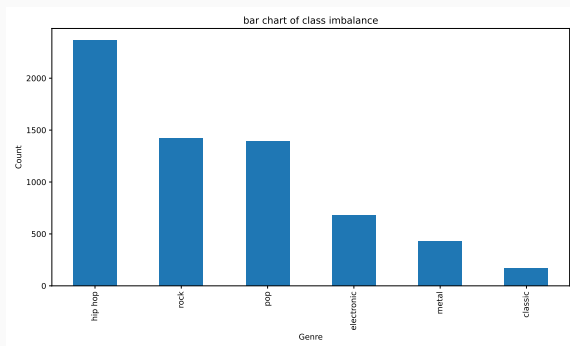
- Contains statistics of 20.7 k songs on Spotify and YouTube
- Does **NOT** include genre information.

## Wikidata Query for the Top-Genre of the Artist

- Query artist's Wikidata page for genres
- Assign artists/album genre to song

## Selection

- Group Subgenres into Supercategories
- Select sample of 6 Genres
- Remaining Songs: 6446

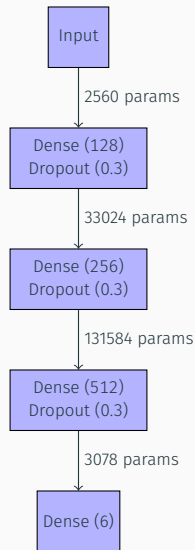


## Model

- 4 Dense Layers with Dropout
- Trainable parameters: 170 246
- Loss function: categorical crossentropy
- Optimizer: **adam**

## Training

- Transform data to a normal distribution with **QuantileTransformer**
- Scaling on  $[-1, 1]$  with **MinMaxScaler**
- Early stopping: Stops training when the validation loss function no longer improves
- Reduce learning rate: Decreases learning rate if validation loss function stagnates  
→ better convergence
- Train the model using the training data with the defined set of hyperparameters.



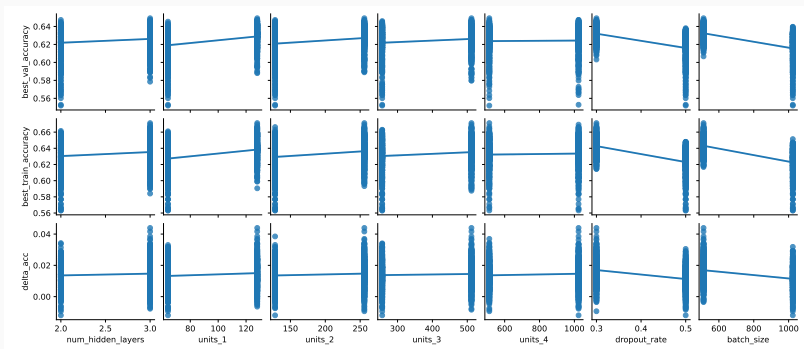
# Hyperparameter Optimization

## Method

- Grid Search: Train models with all combinations of hyperparameters

## Validation

- $k = 3$  Cross Validation
- Save train/validation Accuracy and Loss

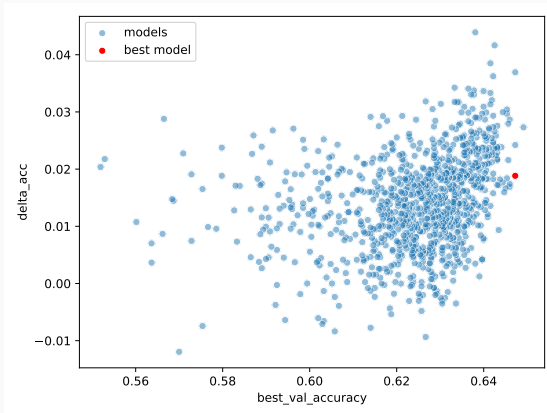


# Overtraining Checks

## Methods to prevent Overtraining

- Dropout
- Early stopping
- Minimize (Training Acc. - Validation Acc.) **but** maximize Validation Acc.

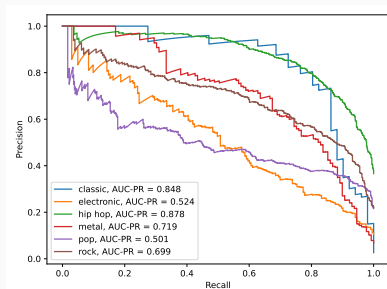
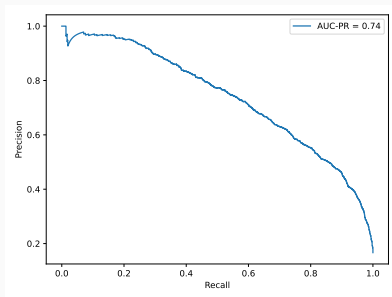
Try different values and decide after Hyperparameter Optimization



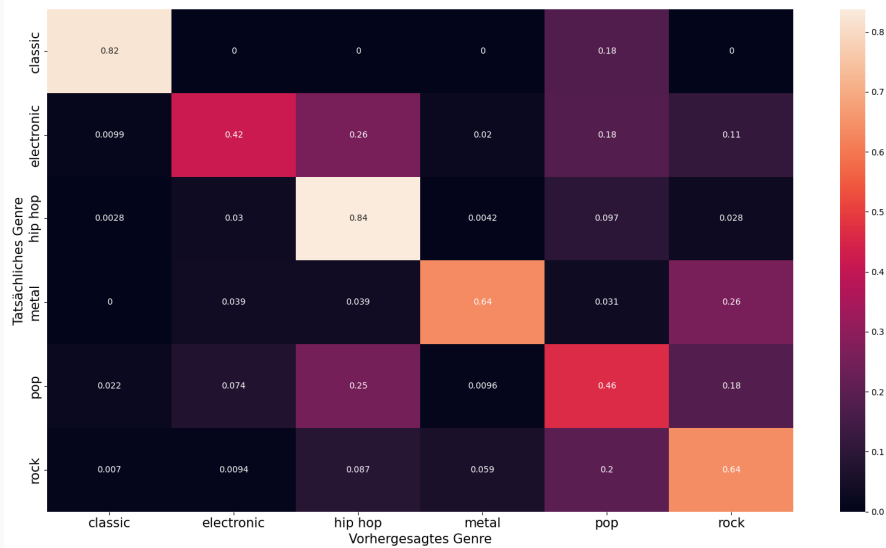
# Results of our Neural Network

## Accuracy and AUC-PR

- Results in an accuracy of 65.56 % on test data
- As well as an AUC-PR score of 0.738



# Results of our Neural Network





## K-nearest-neighbors

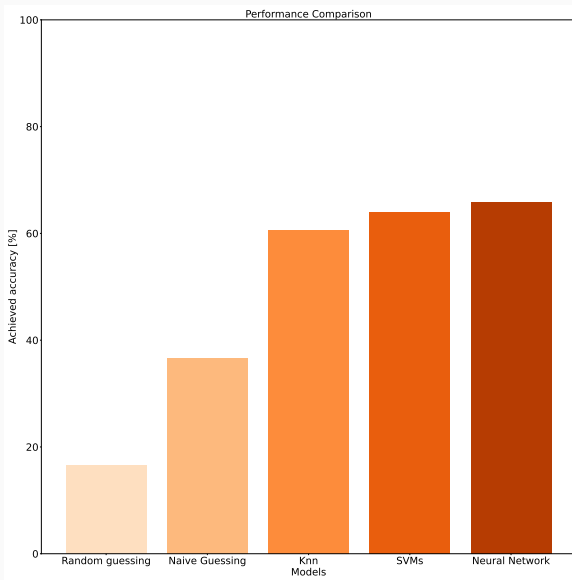
- Use  $k = 12$  as it achieves the highest performance
- Results in an accuracy of 60.62 %

## Support vector machines

- Model that classifies data by finding the hyperplane that maximally separates different categories in a multidimensional space
- Use an One-vs-One approach to be able to do Multiclass-Classification:
  - A separate model is trained for each pair of classes, and a given data point is classified by majority voting among the classifiers
- The used kernel function is the radial basis function (RBF)
- Results in an accuracy of 63.88 %

# Conclusions

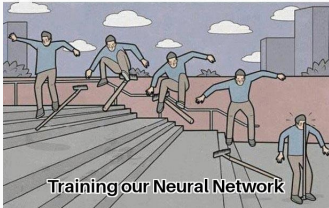
- NN achieves an **accuracy** of 65.56 % on test data.
- Diminishing returns for more complex models, we are constrained by the dataset



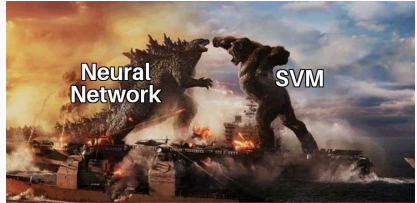
# Conclusions



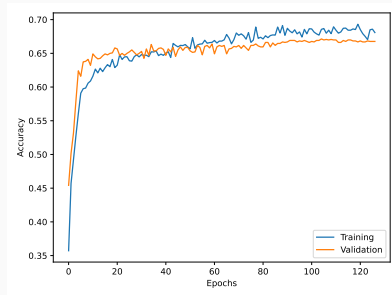
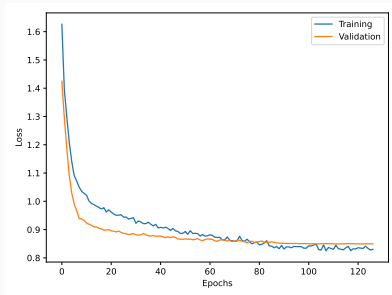
Training our KNN



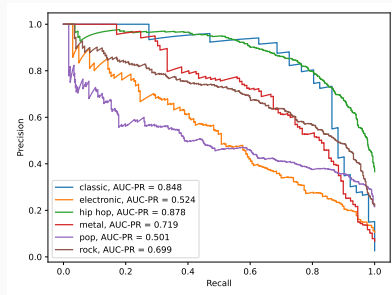
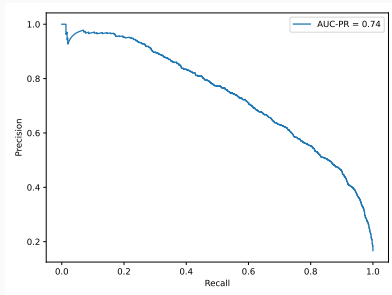
Training our Neural Network



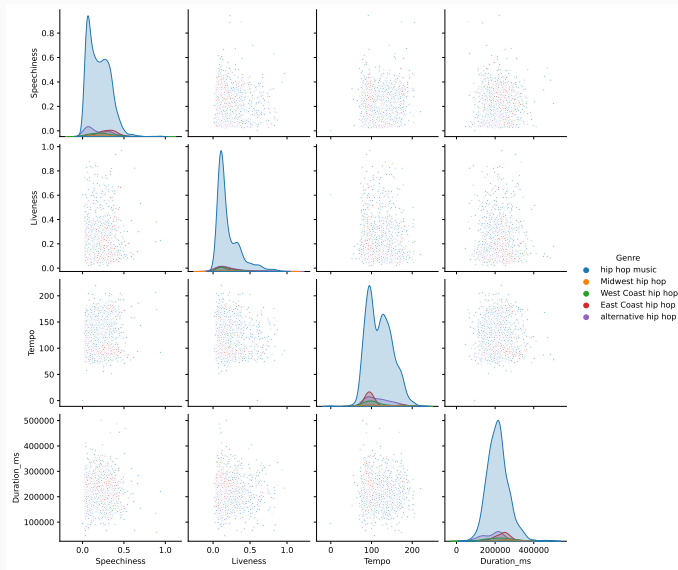
## Appendix: Accuracy and Loss



## Appendix: Precision-Recall Curve



# Appendix: Substructure of Hip Hop



```
param_space = {
    'num_hidden_layers': [2, 3],
    'units_1': [64, 128],
    'units_2': [128, 256],
    'units_3': [256, 512],
    'units_4': [512, 1024],
    'dropout_rate': [0.3, 0.5],
    'activation_function': ['LeakyReLU', "relu"],
    'batch_size' : [128, 256, 512, 1024],
    'early_stopping_patience': [15, 20],
    'reduce_lr_factor': [0.1, 0.5],
    'reduce_lr_patience': [5, 7],
    'reduce_lr_min_lr': [1e-7, 1e-5],
}
```