Spotify®

IS 424 G2 Group 3
Sean Chai
Jazreel Tho
Ngoh Yi Long
Siew Wei Lun
Lee Cheng Leng

SEQUENTIAL SKIP PREDICTION USING SUPERVISED LEARNING TECHNIQUES

A RECAP...

Spotify is looking to make song recommendations in a more innovative way by...

- Finding patterns in user's sequential interactions with recommended songs
- Making predictions of user's skip behavior based on patterns

OUR MOTIVATION

Not many recommendation systems have explored the possibility of recommending music based on the skip behaviour of users

Interesting

Challenging

Opportunity to innovate

LITERATURE REVIEW

Deep Learning

1) LSTM Neural Network
- Popular due to its consideration of the sequential nature of data
- Single model used to predict all track positions
- Two bidirectional LSTM layers used

1) Sequence Learning with Attention
- Modelled after text encoders in Text-to-Speech systems
- Uses dilated convolution layers and Gated Linear Units (GLUs)
- Attention modules allow the model to focus on important parts of the sequence, improving its accuracy

LITERATURE REVIEW

Gradient Boosted Trees

1) Multiple independent GBTs
- Using majority decision to prevent overfitting by a single GBT
- Importance of feature is related to the number of times it is selected by the GBTs

1) XGBoost3 to predict skips of songs
- Train multiple position-dependent models that predict a skip at a particular track position in a session

ORIGINAL DATASETS

Track Features dataset:

- 1.2 GB

Competition Training Dataset:

- 56GB across 16 days

Competition Test Dataset:

- 14GB across 16 days

Due to computational limits, we only managed to run our models on **1 day**'s worth of data

DATASETS

Track Features dataset

- **31 features** for each track, including acoustic analysis

Training dataset

- shows **first half** of user's listening session

Test dataset

- shows **second half** of user's listening session

DATASETS

<u>Summary of data</u>

No. of rows of data: **2,990,609**

No. of unique sessions: **178,342**

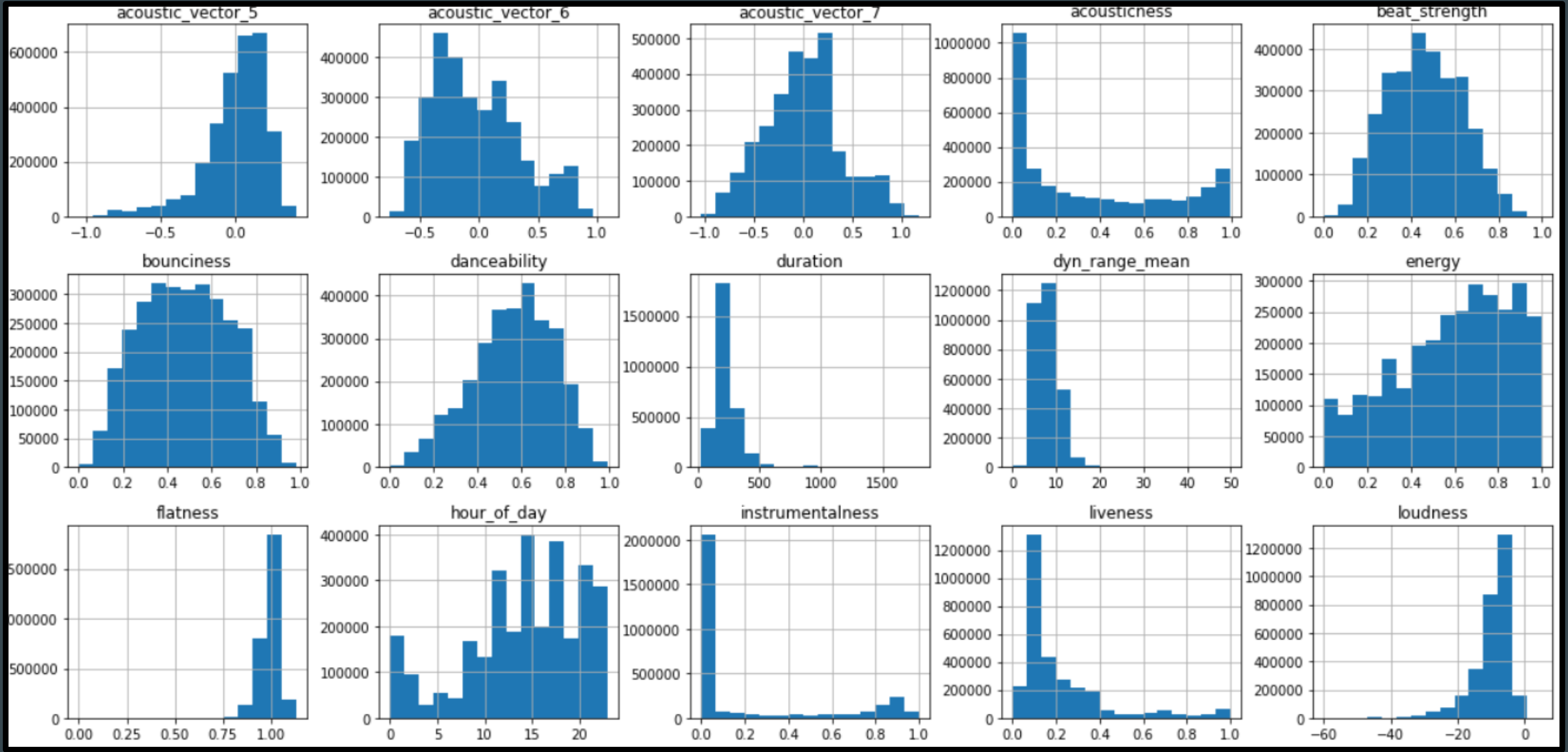
<u>Evaluation metric</u>

Accuracy

EXPLORATORY DATA ANALYSIS

- ☑ Check for missing values
- ☑ Check for categorical data
- ☑ Identify variables with different scales
- ☑ Identify correlation between variables

DISTRIBUTIONS FOR CONTINUOUS VARIABLES

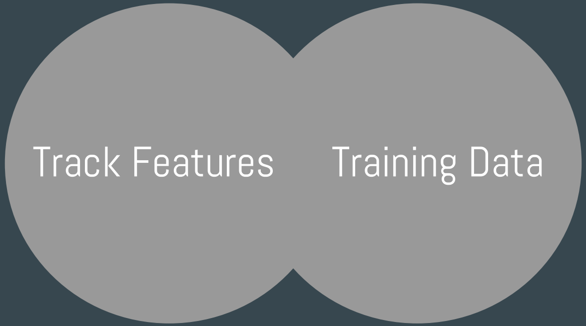


DATA PREPROCESSING

- ☑ Data Integration
- ☑ Data Cleaning
- ☑ Data Selection/ Reduction
- ☑ Data Transformation

DATA INTEGRATION

Merged the **track features** and **training dataset** to obtain a more complete representation of the songs



session_id	session_position	session_length	track_id_clean	skip_1	skip_2	skip_3	not_skipped	track_id	duration	release_year	us_popularity_estimate	acousticness	beat_strength	bounciness

DATA CLEANING:

CATEGORICAL FEATURES

track_id_clean
t_0479f24c-27d2-46d6-a00c-7ec928f2b539
t_9099cd7b-c238-47b7-9381-f23f2c1d1043
t_fc5df5ba-5396-49a7-8b29-35d0d28249e0
t_23cff8d6-d874-4b20-83dc-94e450e8aa20
t_64f3743c-f624-46bb-a579-0f3f9a07a123

track_id_enc
5621
180239
314424
44689
125848

1

Categorical data needs to be transformed

2

Models can only understand numerical data

3

Data smoothing was conducted, which involved encoding the categorical variables

DATA CLEANING:

FEATURES WITH DIFFERENT SCALES

`sklearn.preprocessing.MinMaxScaler`

1

Difference in data range among features

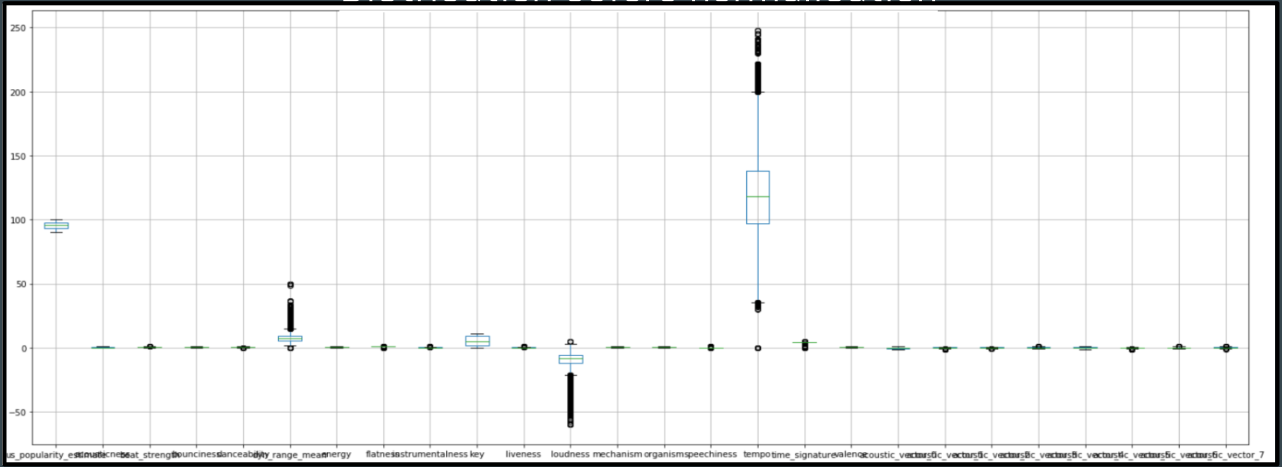
2

Data rescaling for it to be in the same range

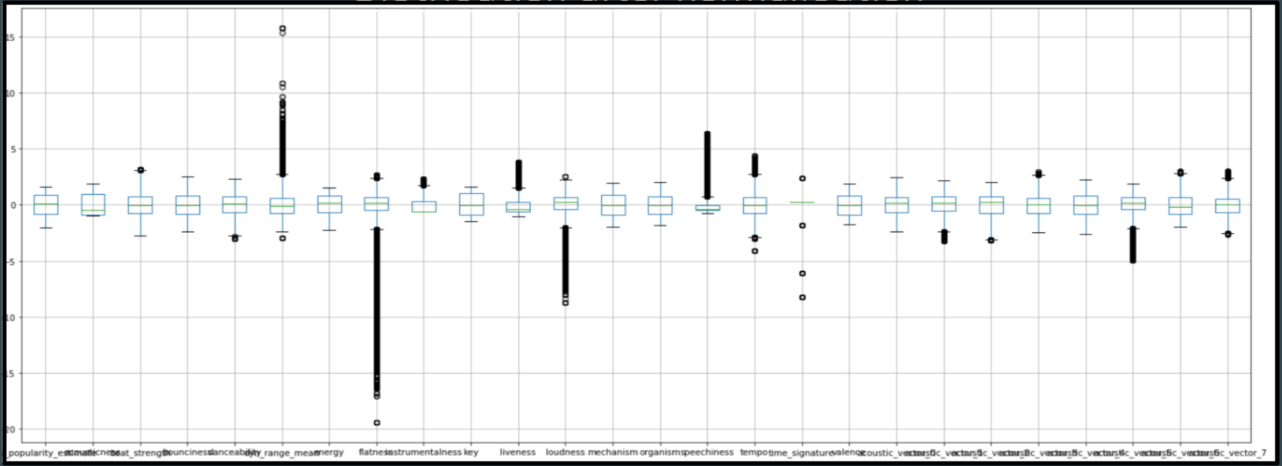
3

Conducted **Normalisation** to change values to **range of** [0, 1]

Distribution before normalisation



Distribution after normalisation



FEATURE REDUCTION:

CURSE OF DIMENSIONALITY

1

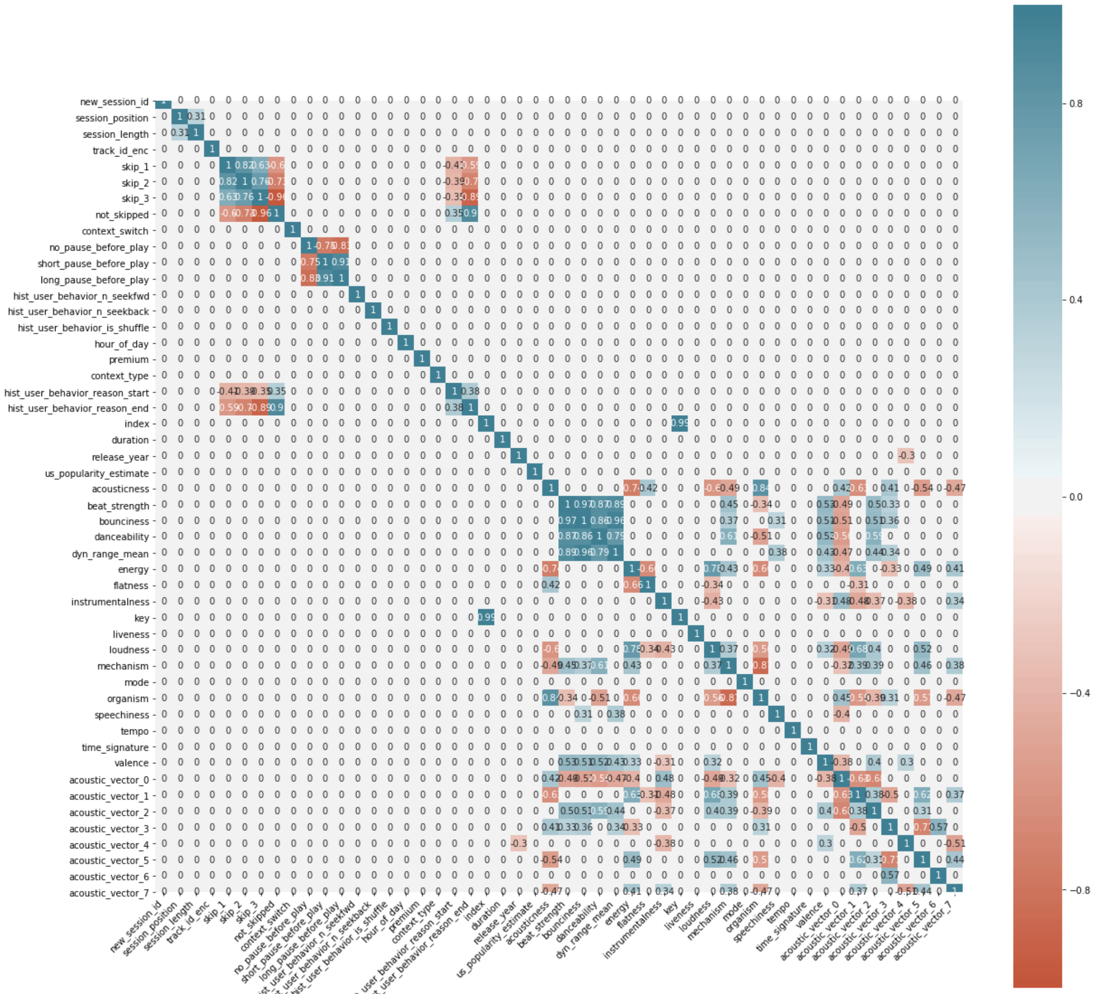
High dimensionality of 46 features in dataset makes it difficult for model to generalise

2

Plot a **correlation matrix** to discover relationships

3

Eliminate independent variables which are correlated with each other



new_session_id
session_position
session_length
track_id_enc
skip_1
skip_2
skip_3
not_skipped
context_switch
no_pause_before_play
short_pause_before_play
long_pause_before_play
hist_user_behavior_n_seekfwd
hist_user_behavior_n_seekback
hist_user_behavior_is_shuffle
hour_of_day
premium
context_type
hist_user_behavior_reason_start
hist_user_behavior_reason_end
index
duration
release_year
us_popularity_estimate
acousticness
beat_strength
bounciness
danceability
dyn_range_mean
energy
flatness
instrumentalness
key
liveness
loudness
mechanism
mode
organism
speechiness
tempo
time_signature
valence
acoustic_vector_0
acoustic_vector_1
acoustic_vector_2
acoustic_vector_3
acoustic_vector_4
acoustic_vector_5
acoustic_vector_6
acoustic_vector_7
0.8
0.4
0.0
−0.4
−0.8

FEATURE REDUCTION:

CURSE OF DIMENSIONALITY

1

High dimensionality of 46 features in dataset makes it difficult for model to generalise

2

Conducted **Principal-Component Analysis (PCA)**

3

Derive **low-dimensionality** projections of the data that explains the variance in the dataset

FEATURE REDUCTION

- **Dropped columns for skip variables other than skip_2**

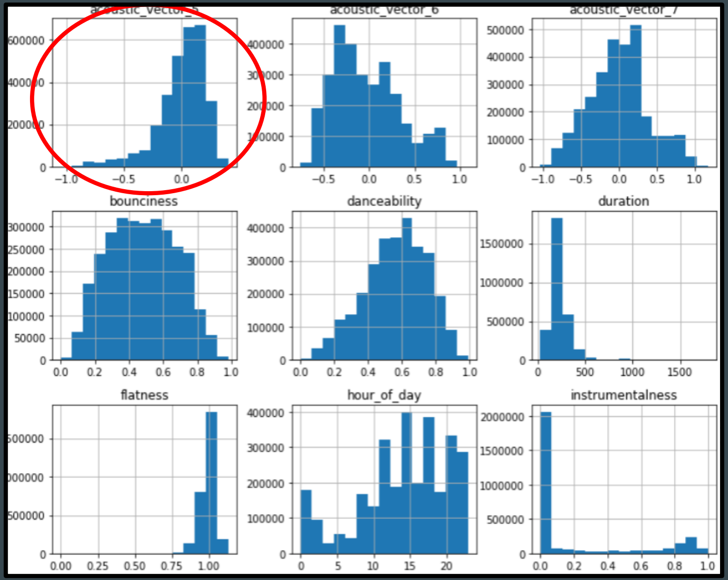
Rationale: Skip variables are highly correlated and redundant in predicting the ground truth variable

- **Decided not to use PCA components**

Rationale: The explained variance with the top 5 PCA components was 57.98%.
However, when a logistic regression model was run on PCA components which explained 95% variance, the accuracy was only 55.4%
Eventually, we did not use the PCA components

DATA TRANSFORMATION

- ☑ Transforming skewed data distribution
- ☑ Exponential on left-skewed data distribution
- ☑ Logarithm on right-skewed data distribution
- ☑ Helps us find patterns after transforming



TRAINING METHODOLOGY

1. Split the training dataset into train and test set by session position, where **first half = training** and **second half = test**

1. Train model on training set and calculate accuracy on test set using sklearn.metrics

1. Based on the results, conduct model-specific hyperparameter tuning if applicable

1. Run the model on the tuned parameters iteratively to find the optimal model parameters

MODEL RESULTS & EVALUATION

NAIVE BAYES THEOREM

- Linear Classifier
- Supervised machine learning method
- Works as a probabilistic classifier
- Calculates probability of features
- Assumes features are independent and equal

GAUSSIAN CLASSIFIER

TP	FP
57308	234360
FN	**TN**
39736	266718

Accuracy: 54.17%

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

LIMITATIONS

Not designed to simultaneously support both categorical and continuous features.

LOGISTIC REGRESSION

- **Explain relationship between one dependent binary variable and one independent variable**

- **Dependent variable is binary:** Skip /Not Skip

- **Values for the parameters:** maximum likelihood estimation (MLE)

LOGISTIC REGRESSION

Library used: statsmodels.api

Selected: 39 Variables

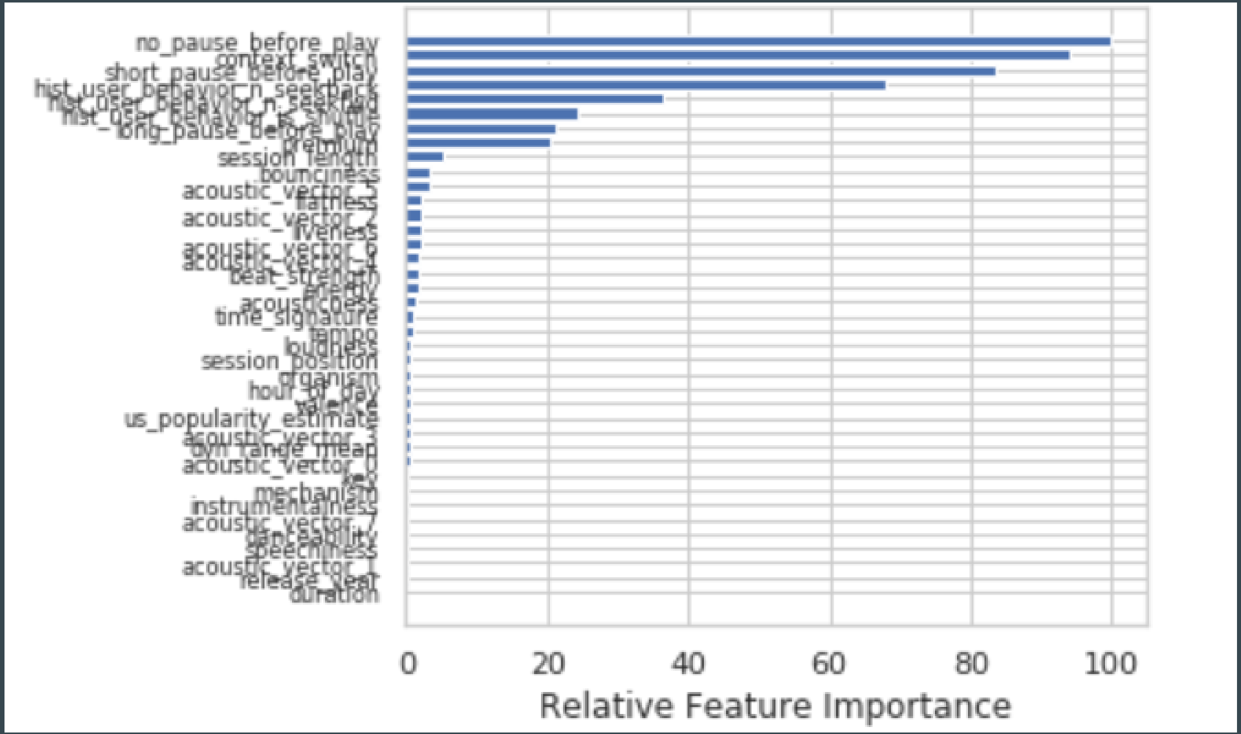
Dropped:

1. no_pause_before_play
2. mode
3. hist_user_behavior_reason_end
4. hist_user_behavior_reason_start
5. dyn_range_mean
6. date

Accuracy: 57. 44%

```
         Current function value: 0.674648
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:                 skip_2   No. Observations:              2093426
Model:                          Logit   Df Residuals:                  2093386
Method:                           MLE   Df Model:                           39
Date:                Wed, 06 Nov 2019   Pseudo R-squ.:                 0.02622
Time:                        14:22:33   Log-Likelihood:            -1.4123e+06
converged:                       True   LL-Null:                   -1.4504e+06
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================================
                                   coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                           -1.4534      0.306     -4.744      0.000      -2.054      -0.853
session_position                -0.0041      0.000    -13.904      0.000      -0.005      -0.004
session_length                   0.0460      0.000    104.901      0.000       0.045       0.047
context_switch                  -0.7484      0.007   -101.380      0.000      -0.763      -0.734
no_pause_before_play             0.7424      0.007    110.715      0.000       0.729       0.756
short_pause_before_play          1.1796      0.011    105.816      0.000       1.158       1.201
long_pause_before_play          -0.6919      0.012    -56.042      0.000      -0.716      -0.668
hist_user_behavior_n_seekfwd     0.2644      0.006     46.407      0.000       0.253       0.276
hist_user_behavior_n_seekback   -0.4200      0.007    -64.124      0.000      -0.433      -0.407
hist_user_behavior_is_shuffle    0.2122      0.003     68.684      0.000       0.206       0.218
hour_of_day                      0.0057      0.000     24.484      0.000       0.005       0.006
premium                         -0.1456      0.004    -39.951      0.000      -0.153      -0.138
duration                      2.803e-05   1.34e-05      2.090      0.037    1.75e-06    5.43e-05
release_year                  5.422e-05      0.000      0.356      0.722      -0.000       0.000
us_popularity_estimate          -0.0035      0.001     -2.454      0.014      -0.006      -0.001
acousticness                    -0.0155      0.006     -2.457      0.014      -0.028      -0.003
beat_strength                    0.0457      0.009      5.238      0.000       0.029       0.063
bounciness                      -0.0783      0.013     -5.972      0.000      -0.104      -0.053
danceability                     0.0031      0.004      0.748      0.455      -0.005       0.011
dyn_range_mean                   0.0197      0.006      3.089      0.002       0.007       0.032
energy                          -0.0094      0.004     -2.368      0.018      -0.017      -0.002
flatness                        -0.0177      0.003     -6.485      0.000      -0.023      -0.012
instrumentalness                -0.0042      0.002     -1.748      0.080      -0.009       0.001
key                             -0.0016      0.001     -1.147      0.251      -0.004       0.001
liveness                        -0.0144      0.001    -10.207      0.000      -0.017      -0.012
loudness                         0.0044      0.003      1.497      0.134      -0.001       0.010
mechanism                        0.0034      0.007      0.490      0.624      -0.010       0.017
organism                         0.0150      0.010      1.451      0.147      -0.005       0.035
speechiness                      0.0003      0.002      0.121      0.904      -0.004       0.005
tempo                            0.0105      0.002      5.404      0.000       0.007       0.014
time_signature                  -0.0082      0.002     -5.423      0.000      -0.011      -0.005
valence                          0.0007      0.002      0.346      0.729      -0.003       0.005
acoustic_vector_0            -2.418e-05      0.007     -0.004      0.997      -0.013       0.013
acoustic_vector_1               -0.0073      0.007     -1.089      0.276      -0.021       0.006
acoustic_vector_2                0.0197      0.003      5.879      0.000       0.013       0.026
acoustic_vector_3               -0.0089      0.008     -1.054      0.292      -0.026       0.008
acoustic_vector_4               -0.0106      0.003     -4.050      0.000      -0.016      -0.005
acoustic_vector_5               -0.0202      0.004     -4.904      0.000      -0.028      -0.012
acoustic_vector_6               -0.0130      0.004     -3.275      0.001      -0.021      -0.005
acoustic_vector_7                0.0048      0.003      1.629      0.103      -0.001       0.011
==============================================================================================
```

LOGISTIC REGRESSION



0 Importance: ['duration','release_year', 'acoustic_vector_1' ,'speechiness','danceability','acoustic_vector_7','instrumentalness','mechanism','key']

Accuracy: 57.32%

LOGISTIC REGRESSION

Gridsearch Hypertuning

- Determine the optimal values for a given model
- A hyperparameter -> value set before the learning process begins

LOGISTIC REGRESSION - HYPERTUNING

Parameters:

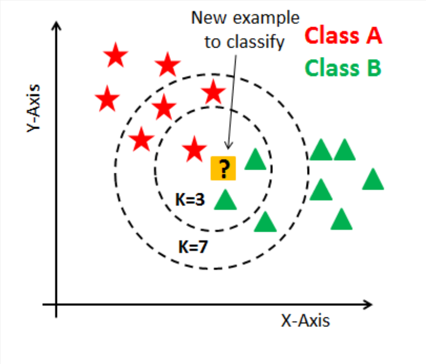
- C = [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]
- penalty = ['l2']
- cv = 5
- verbose=0

Best Parameters: {'C': 0.001, 'penalty': 'l2'}

Accuracy: 57. 46%

LIMITATIONS

Vulnerable to overfitting



K NEAREST NEIGHBOURS

- Distance calculated by a few metrics
- Usually euclidean distance
- Based on closest (k) similar points
- Classify based on <u>odd number</u> of closest points

K NEAREST NEIGHBOURS

Base KNN Model (46 Features)

	k = 5	k = 7	k = 9
Result (Accuracy)	50.941%	50.906%	50.888%

KNN Model (20 Features)

- Using feature_selection.SelectKBest from sklearn

	k = 5	k = 7	k = 9
Result (Accuracy)	50.943%	50.901%	50.886%

	Specs	Score
15	hist_user_behavior_reason_end	1.508963e+06
3	track_id_enc	7.285853e+05
14	hist_user_behavior_reason_start	5.468766e+05
9	hist_user_behavior_n_seekback	2.658098e+04
1	session_position	2.109660e+04
13	context_type	1.982957e+04
0	new_session_id	1.633599e+04
4	context_switch	1.554118e+04
2	session_length	1.308398e+04
7	long_pause_before_play	1.247439e+04
5	no_pause_before_play	7.688899e+03
10	hist_user_behavior_is_shuffle	5.234450e+03
6	short_pause_before_play	2.498477e+03
11	hour_of_day	1.466178e+03
8	hist_user_behavior_n_seekfwd	1.104502e+03
16	duration	8.952575e+02
28	liveness	5.248749e+01
12	premium	4.561756e+01
19	acousticness	3.157311e+01
43	acoustic_vector_6	1.444756e+01

LIMITATIONS

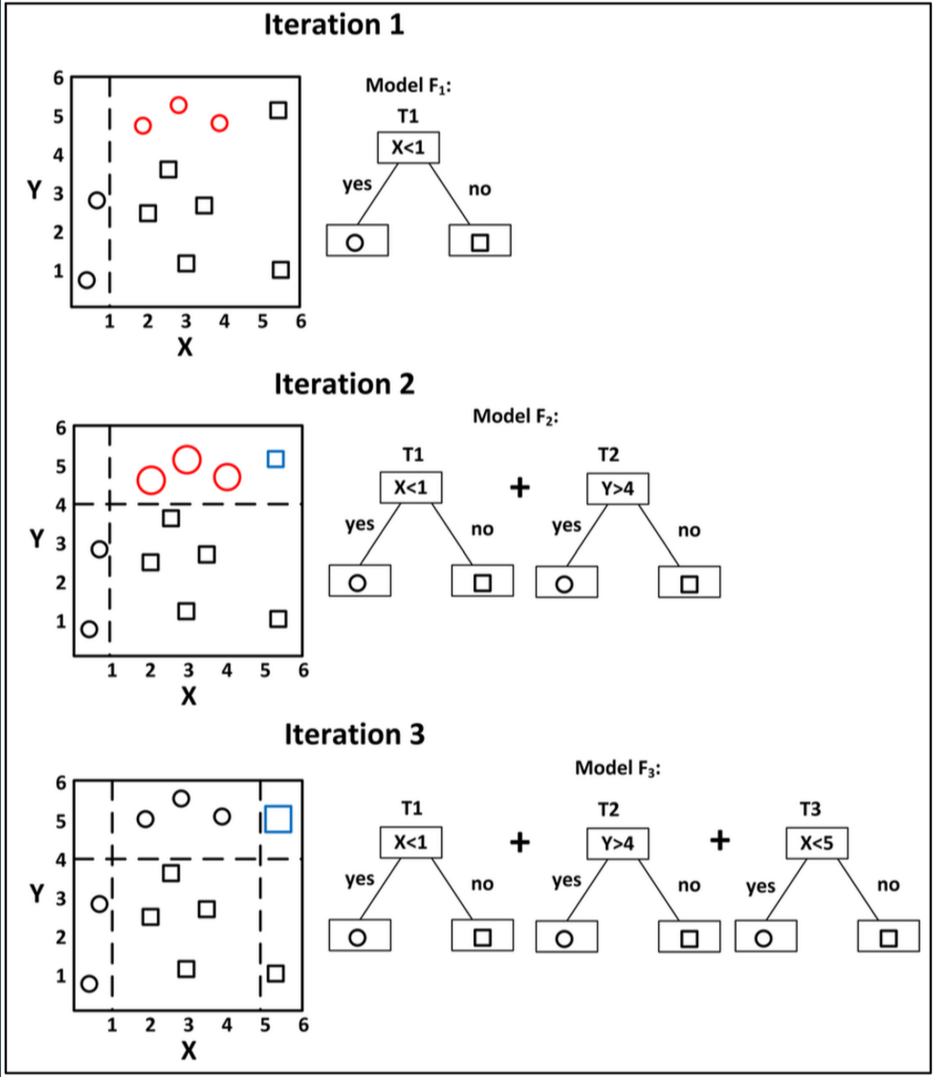
1. Calculation of the euclidean distance of ALL data points
2. High computational cost
3. Hard to determine optimal value of k value
4. Very sensitive to large magnitudes of features since distance measure is used

GRADIENT BOOSTED DECISION TREES

XGBOOST

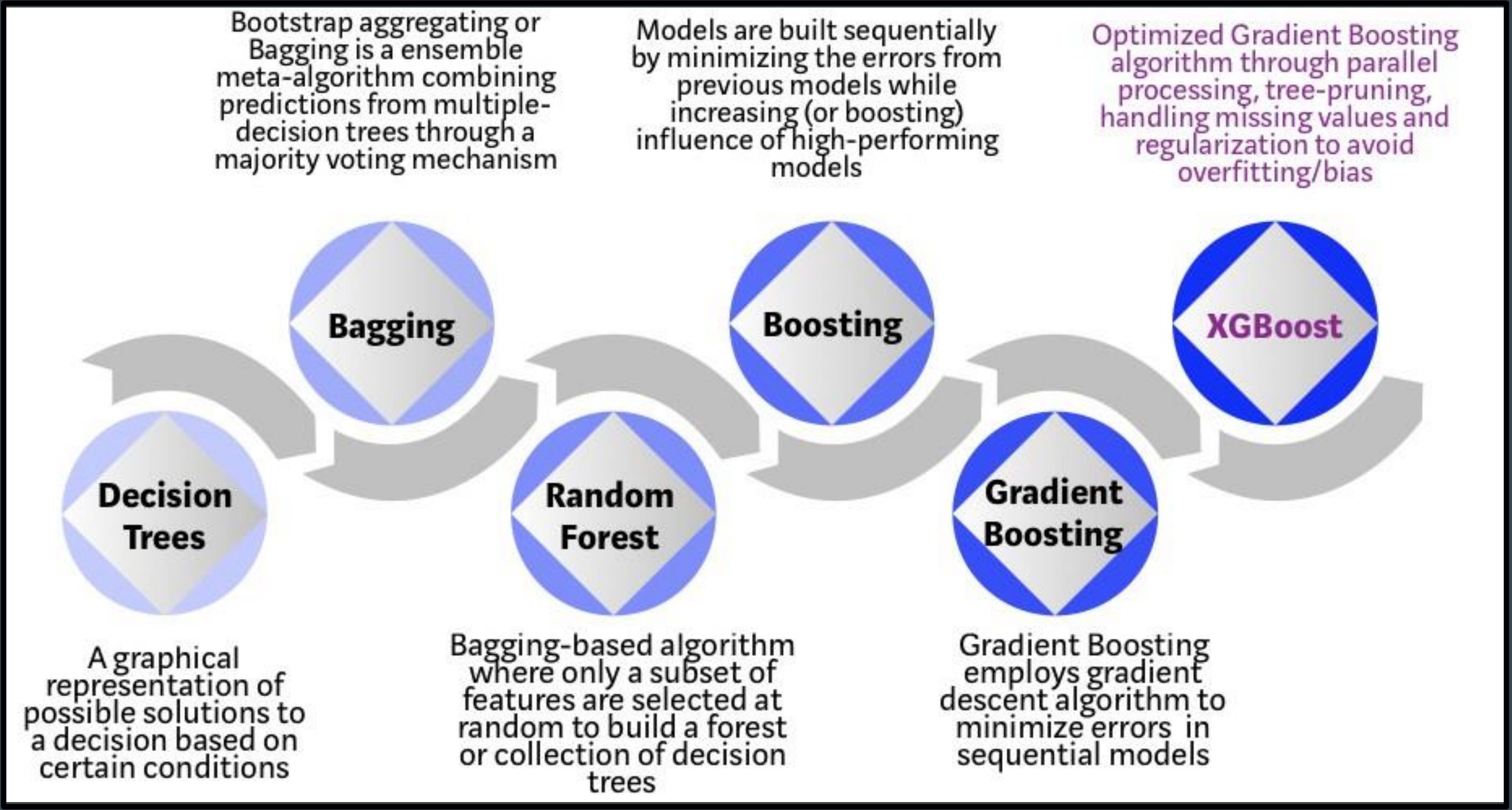
- Gradient boosted decision tree
- Optimised for speed and performance
- Prediction based on residuals/errors
- Gradient descent algorithm used to minimize errors

XGBoost



Iteration 1
Model F1:
T1
X<1
yes
no
Y
X
Iteration 2
Model F2:
T1
X<1
+
T2
Y>4
yes
no
Y
X
Iteration 3
Model F3:
T1
X<1
+
T2
Y>4
+
T3
X<5
yes
no
Y
X

XGBoost


Decision Trees
A graphical representation of possible solutions to a decision based on certain conditions
Bagging
Bootstrap aggregating or Bagging is a ensemble meta-algorithm combining predictions from multiple-decision trees through a majority voting mechanism
Random Forest
Bagging-based algorithm where only a subset of features are selected at random to build a forest or collection of decision trees
Boosting
Models are built sequentially by minimizing the errors from previous models while increasing (or boosting) influence of high-performing models
Gradient Boosting
Gradient Boosting employs gradient descent algorithm to minimize errors in sequential models
XGBoost
Optimized Gradient Boosting algorithm through parallel processing, tree-pruning, handling missing values and regularization to avoid overfitting/bias

STEPS IN OPTIMISING XGBOOST RESULTS

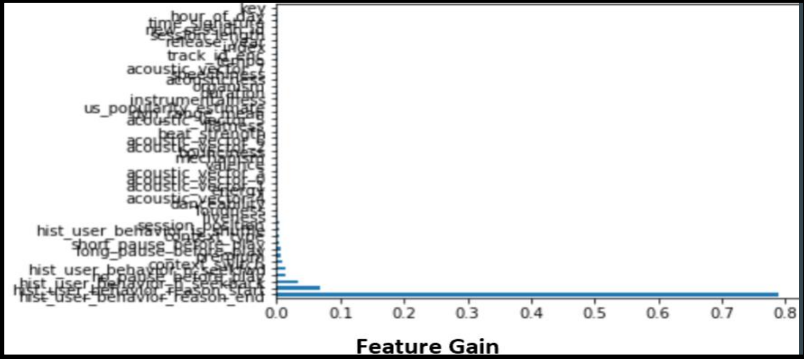
1. Run XGBoost Algorithm to get initial model
2. Select features based on feature importance
3. Run XGBoost Algorithm with selected features
4. Run XGBoost Algorithm with 3-Fold Validation & GridSearch

XGBOOST RESULTS

Results for initial model

```
Time taken: 0 hours 19 minutes and 41.62 seconds.
Accuracy score for initial model: 88.67865665361171%
```

Feature importance by Gain



hist_user_behavior_reason_end	0.807438
hist_user_behavior_reason_start	0.084447
hist_user_behavior_n_seekback	0.029578
no_pause_before_play	0.020297
hist_user_behavior_n_seekfwd	0.012267
context_switch	0.007550

Results for model after important features selected

```
Time taken: 0 hours 9 minutes and 43.97 seconds.
Accuracy score: 88.67212697390899%
```

GridSearch Results after 3-Fold Cross Validation

```
[Parallel(n_jobs=4)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=4)]: Done  24 tasks      | elapsed: 73.5min
[Parallel(n_jobs=4)]: Done 108 out of 108 | elapsed: 357.9min finished
```

```
Time taken: 6 hours 7 minutes and 2.91 seconds.
```

```
Accuracy score with hyperparameters tuned: 88.72067085533527%
```

```
Best estimator:
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0.0,
              learning_rate=0.1, max_delta_step=0, max_depth=7,
              min_child_weight=3, missing=None, n_estimators=500, n_jobs=3,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=0, scale_pos_weight=1, seed=1,
              silent=True, subsample=1, tree_method='exact', verbosity=1)
```

XGBoost

LIMITATIONS

1. Too many hyperparameters makes optimal tuning of hyperparameters time consuming
2. XGBoost unable to accept categorical features, encoding is required
3. Prone to overfitting

LSTM

RNN

- Suffers from short-term memory
- Difficult to pass information from earlier time steps to later ones.
- Vanishing gradient problem during backward propagation
- Small gradient value doesn't contribute too much learning

VS

LSTM

- Similar to RNN
- Have internal mechanisms to regulate flow of information
- Able to retain important information across time steps

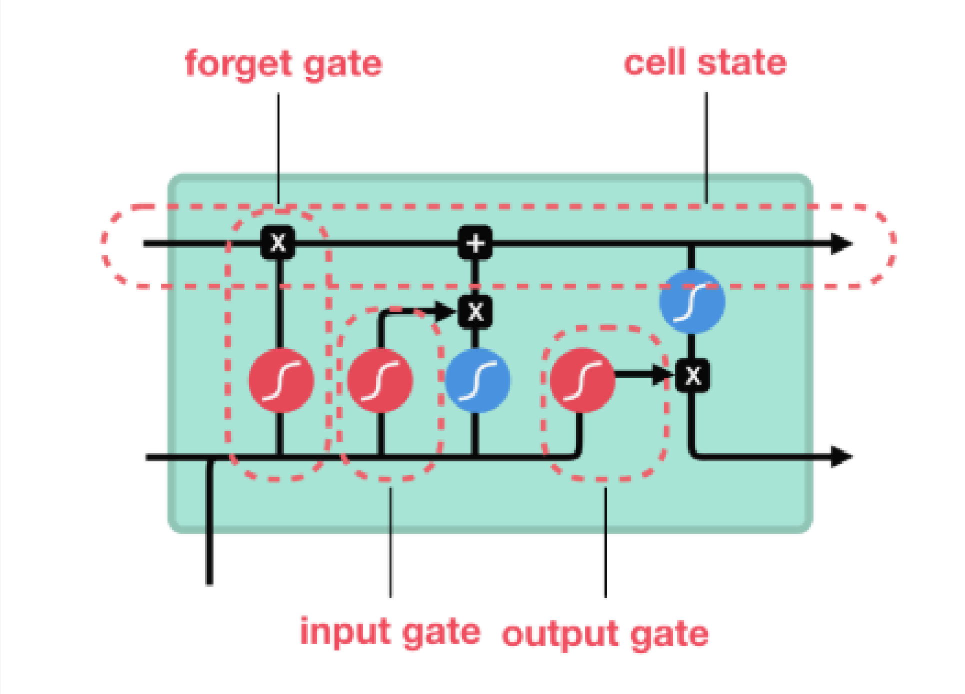
RELEVANCE OF LSTM

Aim to utilize first half of the listening session to predict the probability of skipping a track in the second half of the session

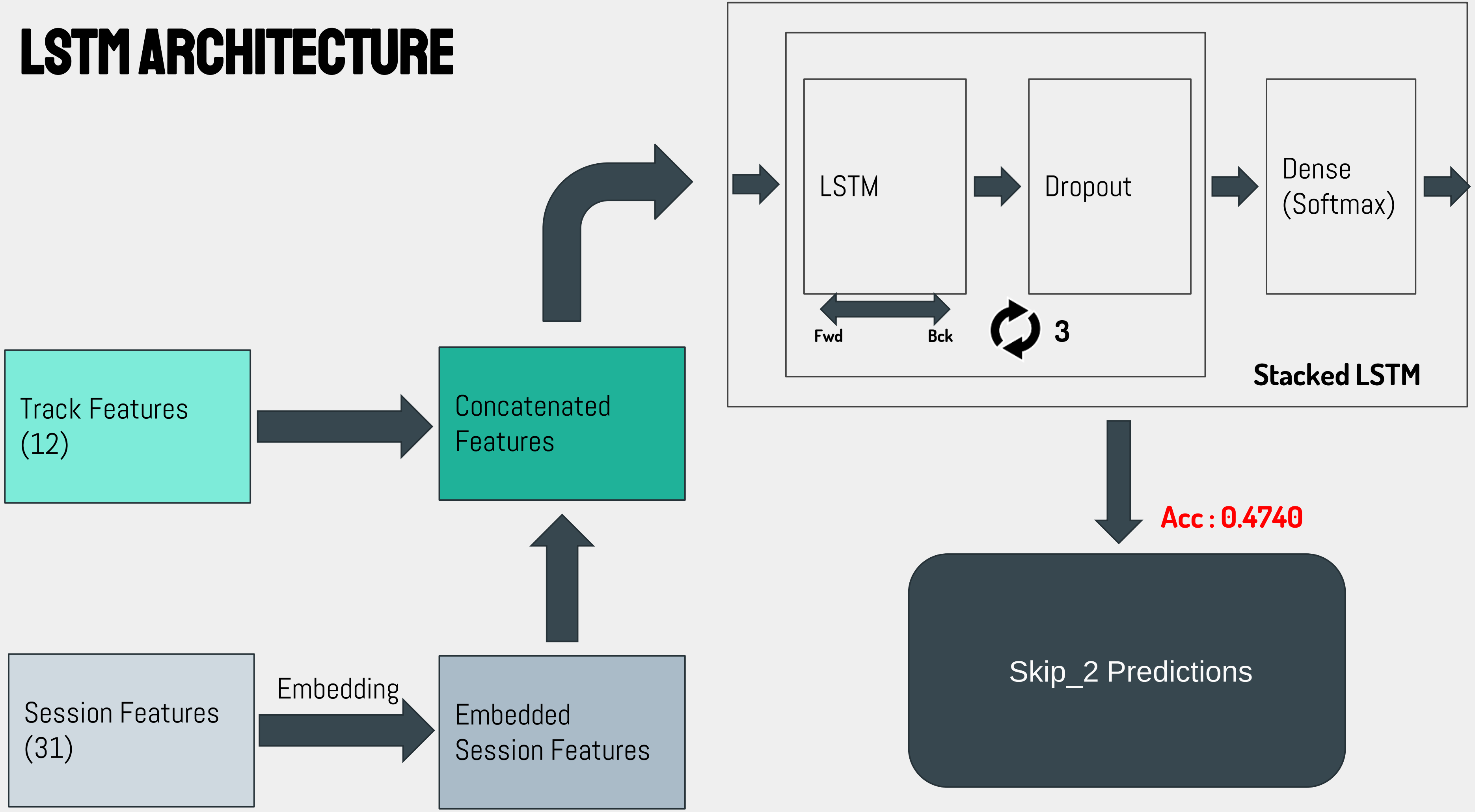
Understand the temporal relationships for particular note and modeling the joint distribution of notes in the particular timestep.

Provides us with the flexibility to alter the hyperparameters to suit our specific needs.

LSTM DIAGRAM



LSTM ARCHITECTURE



LSTM Model Summary

Layer (type)	Output Shape	Param #	Connected to
cat1_input (InputLayer)	(None, 89672, 10)	0	
cat2_input (InputLayer)	(None, 89672, 10)	0	
cat3_input (InputLayer)	(None, 89672, 10)	0	
cat4_input (InputLayer)	(None, 89672, 10)	0	
cat5_input (InputLayer)	(None, 89672, 10)	0	
cat6_input (InputLayer)	(None, 89672, 10)	0	
cat7_input (InputLayer)	(None, 89672, 10)	0	
cat8_input (InputLayer)	(None, 89672, 10)	0	
cat9_input (InputLayer)	(None, 89672, 10)	0	
cat10_input (InputLayer)	(None, 89672, 10)	0	
cat11_input (InputLayer)	(None, 89672, 10)	0	
cat12_input (InputLayer)	(None, 89672, 10)	0	
embedding_26 (Embedding)	(None, 89672, 10, 2)	2	cat1_input[0][0]
embedding_28 (Embedding)	(None, 89672, 10, 2)	2	cat2_input[0][0]
embedding_30 (Embedding)	(None, 89672, 10, 2)	2	cat3_input[0][0]
embedding_32 (Embedding)	(None, 89672, 10, 2)	2	cat4_input[0][0]
embedding_34 (Embedding)	(None, 89672, 10, 41	861	cat5_input[0][0]
embedding_36 (Embedding)	(None, 89672, 10, 47	1128	cat6_input[0][0]
embedding_38 (Embedding)	(None, 89672, 10, 2)	2	cat7_input[0][0]
embedding_40 (Embedding)	(None, 89672, 10, 24	288	cat8_input[0][0]
embedding_42 (Embedding)	(None, 89672, 10, 2)	2	cat9_input[0][0]
embedding_44 (Embedding)	(None, 89672, 10, 6)	18	cat10_input[0][0]
embedding_46 (Embedding)	(None, 89672, 10, 10	50	cat11_input[0][0]
embedding_48 (Embedding)	(None, 89672, 10, 9)	45	cat12_input[0][0]
concatenate_3 (Concatenate)	(None, 89672, 10, 14	0	embedding_26[0][0] embedding_28[0][0] embedding_30[0][0] embedding_32[0][0] embedding_34[0][0] embedding_36[0][0] embedding_38[0][0] embedding_40[0][0] embedding_42[0][0] embedding_44[0][0] embedding_46[0][0] embedding_48[0][0]
numeric_input (InputLayer)	(None, 89672, 310)	0	
reshape_2 (Reshape)	(None, 89672, 1490)	0	concatenate_3[0][0]
concatenate_4 (Concatenate)	(None, 89672, 1800)	0	numeric_input[0][0] reshape_2[0][0]
lstm_4 (LSTM)	(None, 89672, 64)	477440	concatenate_4[0][0]
dropout_3 (Dropout)	(None, 89672, 64)	0	lstm_4[0][0]
lstm_5 (LSTM)	(None, 89672, 64)	33024	dropout_3[0][0]
dropout_4 (Dropout)	(None, 89672, 64)	0	lstm_5[0][0]
lstm_6 (LSTM)	(None, 89672, 64)	33024	dropout_4[0][0]
dense_2 (Dense)	(None, 89672, 10)	650	lstm_6[0][0]

Total params: 546,540
Trainable params: 546,540
Non-trainable params: 0

LIMITATIONS

1. Difficult to train because they require memory-bandwidth-bound computation
2. Vanishing gradient problem still exists
3. High risk of overfitting
4. LSTMs are sensitive to random weight initializations

COMPARISON OF MODELS

COMPARISON OF MODELS

	Naive Bayes	Logistic Regression	k-Nearest Neighbours	LSTM	XGBoost
Time Complexity	✓	✓			✓
Flexibility				✓	✓
Goodness of Fit				✓	
Accuracy					✓

CONCLUSION

- We have used a combination of classification and prediction models to study the best supervised learning method for this task
- The models we have developed showed a marked improvement over the baseline model GaussianNB, from **54.17%** to **88.7%** from the XGBoost implementation
- hist_user_behaviour_reason_end and hist_user_behaviour_reason_start are the two most important features in predicting the skip behaviour of the user
- Concatenation of track and session features provide a more comprehensive understanding of the probability of skipping a track

FUTURE WORK

- ➢ Increase dataset size as larger dataset may improve results of the models utilized
- ➢ Introduce Multi-RNNs in the LSTM model and evaluate its accuracy against our LSTM Model
- ➢ Fine-tune hyperparameters in LSTM Model
- ➢ Creating more GBTs and ensembling the models to find the average predictions on the models we trained
- ➢ Perform validation test to gain an unbiased estimate of the performance of the final tuned model
- ➢ Instead of just focusing on the accuracy, we should look at the lost metrics such as negative log-likelihood and residual sum of squares.

REFERENCES

https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e

https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

https://arxiv.org/ftp/arxiv/papers/1804/1804.07300.pdf

https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

https://machinelearningmastery.com/models-sequence-prediction-recurrent-neural-networks/

http://data-mining.philippe-fournier-viger.com/an-introduction-to-sequence-prediction/

https://www.cddt.nz/projects/spotify.html

https://people.eng.unimelb.edu.au/jianzhongq/wsdm19-cup-reports/reports/report16.pdf

https://arxiv.org/pdf/1901.08203.pdf

https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/

Spotify®

IS 424 G2 Group 3
Sean Chai
Jazreel Tho
Ngoh Yi Long
Siew Wei Lun
Lee Cheng Leng

THANK YOU!