

Work Report Technocolabs Software

Name: **Mudit Vyas**

Date of Internship: **1st Dec 2021 - 15th Jan 2022**

Work and Position: Data Scientist Internship

Aim: The goal of the project is to analyze Spotify Data and find relationship between Track Feature and Consumer Behavior and to build the best classification ML model in terms of accuracy and performance for predicting the Consumer Skipping Behavior.

➤ **Steps involved in project:**

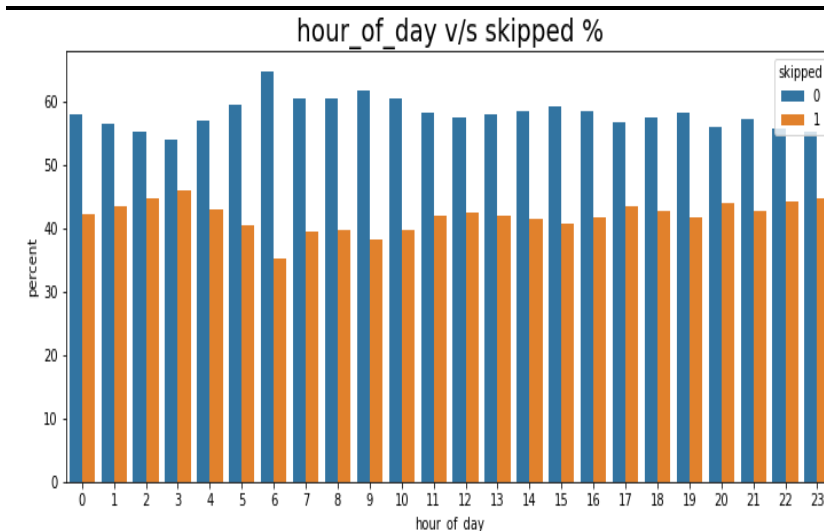
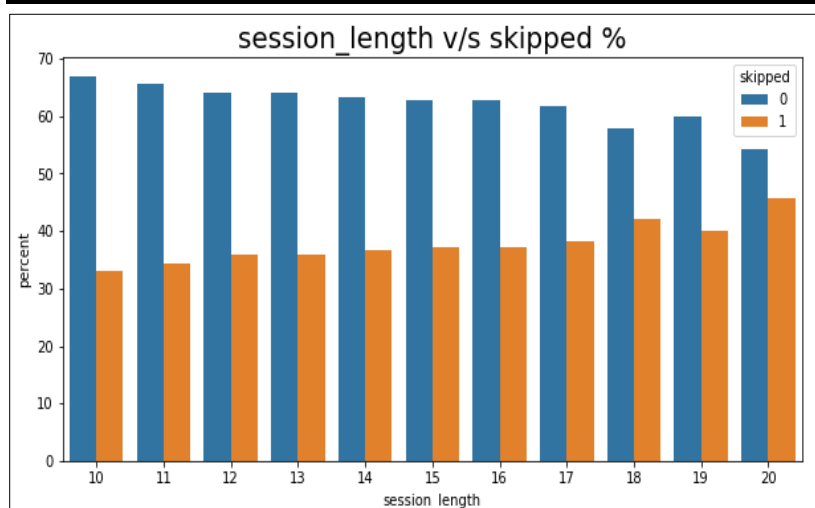
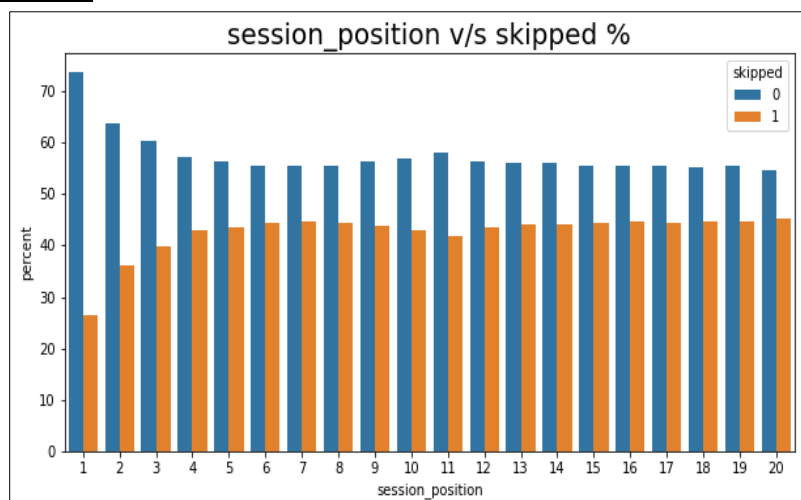
- EDA
- Feature Engineering
- Model Development
- Model Deployment

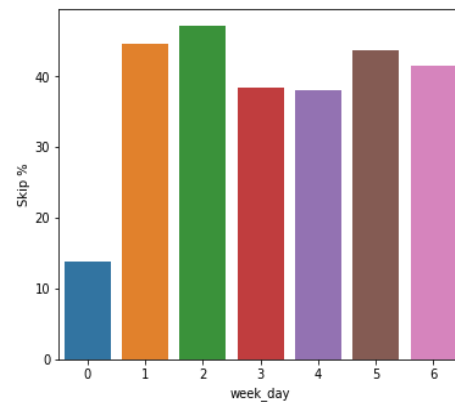
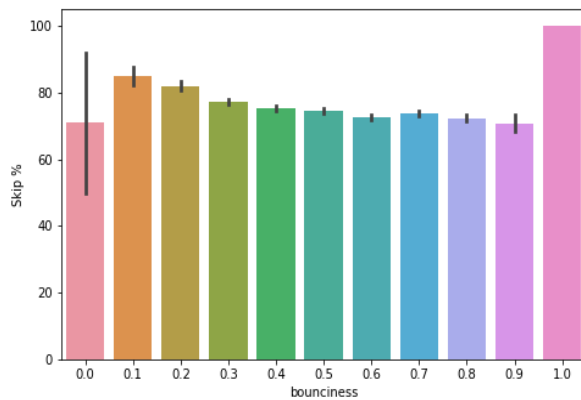
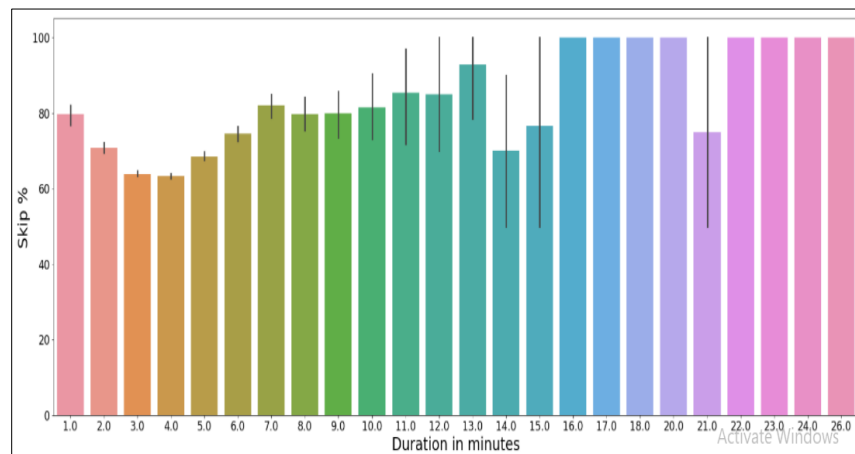
➤ **EDA**

- Since, the data for track feature we got is about 2 GB.
- I divided EDA task in three different profiles:
 - **Track Features:** Knowing the track features like bounciness, loudness, beat, Mode, acoustic Vector etc and their distribution plots. From this we can draw a conclusion that this data require standardization and conversion of categorical objects to numeric (int) data type.
 - **Log Session:** In this data, we have Target Feature: Skip1, Skip2, Skip3 and Skip4 as well as Features such as Session Position, Session Length, Premium, End / Start Button etc. I analyzed different session features with Skip Operations.
 - **Track Feature and Log Session Merged Data:** I merged both data using Left Joint. Also, I combined skip{1,2,3,4} then replaced it by new skipped column which has net result of

skip{1,2,3,4}. Then I performed EDA and did analysis between skip% and features.

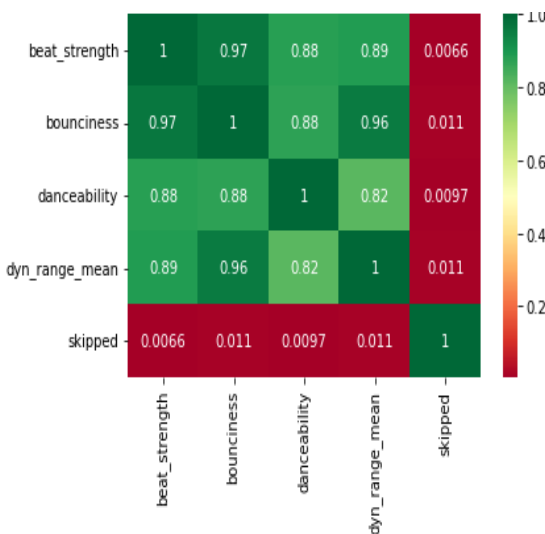
>>EDA Analysis:



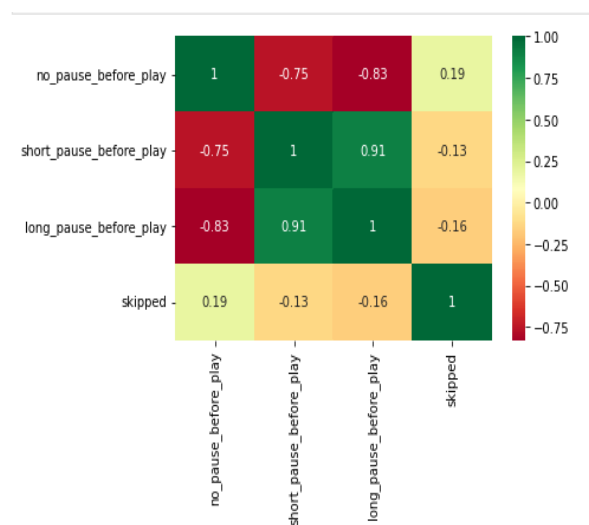


➤ Feature Engineering

- From Correlation Heatmap, I aimed to remove most correlated features among 48-50 features heatmap



Heatmap 1



Heatmap 2

- **Heatmap 1:** Features: Bounciness, Beat Strength, Danceability, dyn_range_mean
 - Out of 4 Features in heatmap 1>> Bounciness and dyn_range_mean is highly correlated features hence are removed.
- **Heatmap 2:** Features: no_pause_before_play, short_pause_before_play, long_pause_before_play
 - Out of 3 Features in heatmap 2>> short_pause_before_play and long_pause_before_play are highly correlated features hence are removed.

➤ ***Model Development***

- Main Model Worked and tried for development process:
 - Logistic Regression
 - XGBoost Classifier
 - Light Gradient Boosting Machine
 - Decision Tree Classifier
- Model Development Process starts by following order:
 - Boolean objects to integer
 - One Hot Label Encoding for
 - Standardization of Session_Track_merged Data
 - Train_Test Splitting of Data
 - Model Training and Evaluation
 - Hyper parameter Tuning
 - If Imbalanced Data then, switch to Undersampling or Oversampling Classification Techniques.

➤ **Summary of Model Training with Imbalanced Data**

- Observed Accuracy Score greater than 80.
- Poor Precision and Recall since Training and Test Data is Imbalanced (Ratio of Skip to Not Skip ~0.667)
- Hence, For Model Training I decided to go for UnderSampling Route.

	Model	Accuracy	Precision	Recall	F1 Score
0	Logistic Regression	83.968708	77.808312	86.692732	82.010606
1	LGBM	86.643237	81.018994	89.212869	84.918731
2	XGBoost Classifier	85.160035	79.753407	86.838758	83.145409
3	Decision Tree Classifier	84.270511	78.177275	86.956522	82.333527

➤ **Good Classification Model has F1score closer to 100.**

➤ **Summary of Model Training with Under-sampled Data**

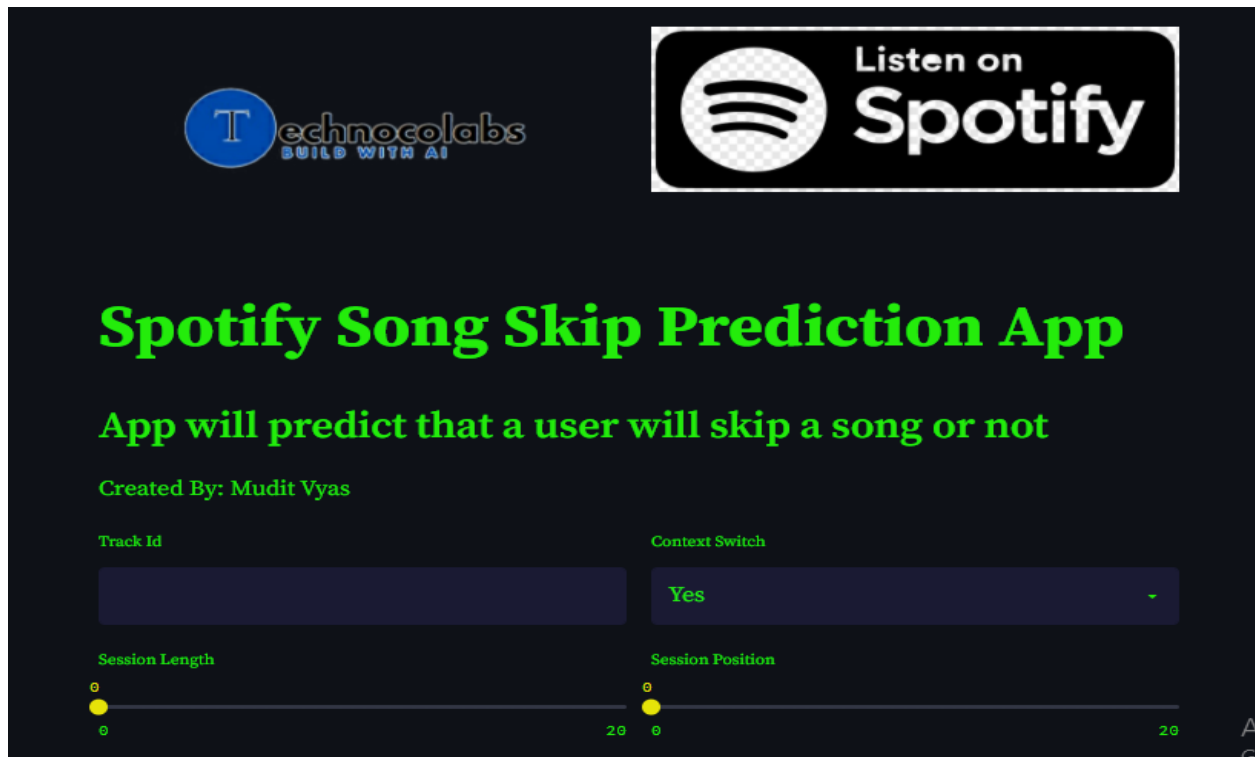
- Random Undersampling method is used for Undersampling of session_track_merged_data.
- Undersampling is **a technique to balance uneven datasets by keeping all of the data in the minority class and decreasing the size of the majority class.**
- Improvement in Accuracy score and classification evaluation metrics.
- DTC, LGBM and XGBoost are top 3 best models.
- **Models performance is assessed on the basis of F Score and Confusion Metrics.**

	Model	Accuracy	Precision	Recall	F1 Score
0	Logistic Regression	87.378115	84.179014	92.058034	87.942401
1	LGBM	88.800697	86.592625	91.817796	89.128695
2	XGBoost Classifier	87.964577	85.783421	91.012295	88.320534
3	Decision Tree Classifier	88.605210	86.098753	92.076876	88.987526

➤ DEPLOYMENT OF ML MODEL

- Model: Decision Tree Classifier
- Webapp built using: Streamlit
- Language: Python
- Application Deployed on: Heroku
- Website/App URL: <https://spotify-skip-prediction-app.herokuapp.com/>

Web App Frontend



The screenshot displays the web interface of the 'Spotify Song Skip Prediction App'. At the top left is the 'Technocolabs BUILD WITH AI' logo. At the top right is a 'Listen on Spotify' button with the Spotify logo. The main heading is 'Spotify Song Skip Prediction App' in large green text, followed by the subtitle 'App will predict that a user will skip a song or not' in smaller green text. Below this, it says 'Created By: Mudit Vyas'. The interface features four input fields: 'Track Id' (a text box), 'Context Switch' (a dropdown menu currently showing 'Yes'), 'Session Length' (a slider from 0 to 20), and 'Session Position' (a slider from 0 to 20). The background is dark blue.

No Pause Before Play: <input checked="" type="radio"/> Yes <input type="radio"/> No	Shuffle: <input checked="" type="radio"/> Yes <input type="radio"/> No
Seek Forward 0 - +	Seek Backward 0 - +
Hours of the Day 0 - +	Date 2022/01/17
Context Type editorial_playlist ▾	Premium Yes ▾
Start Reason trackdone ▾	End Reason trackdone ▾
<div>Predict</div>	

