**Udacity Sparkify Project Report**

Git: <https://github.com/GinnCheng/Cloud-computing-classifier-breed>

**High-level Overview**

The Udacity Sparkify project aims to analyse user behaviour on the Sparkify music streaming platform to identify patterns and predict churn. Churn, in this context, refers to users who stop using the service. Addressing churn is crucial for maintaining a stable user base and enhancing business growth. This project uses data from Sparkify's user activity logs to develop a predictive model that can help the company proactively identify users at risk of churning and implement retention strategies.

**Description of Input Data**

The dataset used for this project comprises user activity logs from the Sparkify music streaming platform. The logs are in .json format, capturing detailed interactions such as song plays, thumbs up, thumbs down, friend additions, and cancellation requests. The variables are significant as they provide insights into user engagement, preferences, and potential indicators of churn.

**Strategy for Solving the Problem**

The overall approach involves several stages: data exploration, pre-processing, feature engineering, modelling, and evaluation. The features are discussed via exploring the data, and machine learning techniques are employed to build a classification model that predicts churn. Algorithms such as decision trees, and random forest are considered due to their interpretability and performance in classification tasks. The rationale behind these choices is their ability to handle large datasets and complex relationships effectively and robustly.

**Discussion of the Expected Solution**

The proposed solution involves creating a pipeline that includes data pre-processing, feature engineering, model training, and evaluation. The workflow starts with data cleaning and transformation, followed by extracting meaningful features such as session duration, length of songs played. These features are then used to train a classification model. The components work together to predict whether a user is likely to churn based on their historical behaviour.

Detail analysis can be referred to the notebook script “Sparkify”.

**Metrics with Justification**

The evaluation metrics used are precision, recall, F1 score, The F1 score balances precision and recall, making it suitable for imbalanced datasets.

**Exploratory Data Analysis (EDA)**

The EDA details can be found in the notebook script “Sparkify”.

**Data Pre-processing**

Data pre-processing steps include:

* **Data Cleaning**: Handling missing values and correcting inconsistencies.
* **Transformation**: Converting timestamps to datetime objects and standardising data.
* **Feature Engineering**: Extracting features such as length of songs played, average session length, and user engagement metrics, and label encoding the text values.

**Modelling**

The chosen models include decision trees, and random forest. Decision Trees offer insights into feature importance, while Random Forest improve predictive accuracy through ensemble learning. The models are implemented using PySpark's MLlib.

**Hyperparameter Tuning**

Hyperparameter tuning is performed using grid search and cross-validation to find the optimal parameters for each model. In the model training process, parameters such as the number of features, model impurity and maximum depth are tuned:

**Results**

The model evaluation results are as follows:

* **Decision Trees (two features)**: F1 Score: 0.597
* **Decision Trees (four features)**: F1 Score: 0.683
* **Decision Trees (more depth)**: F1 Score: 0.683
* **Random Forest**: F1 Score: 0.635

**Comparison Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Precision** | **Recall** | **F1 Score** |
| **Decision Trees (two features)** | 0.784 | 0.656 | 0.597 |
| **Decision Trees (four features)** | 0.781 | 0.708 | 0.683 |
| **Decision Trees (more depth)** | 0.781 | 0.708 | 0.683 |
| **Random Forest** | 0.781 | 0.688 | 0.635 |

**Conclusion**

The project successfully developed a predictive model for user churn on the Sparkify platform. The decision trees model provided the best performance, achieving an F1 score of 0.683.

Due to the limited time, the model still has a capacity to improve. The EDA shows that the churn users and non-churn users are significantly imbalanced. Further feature engineering can be applied includes, for instance, generating new feature of user activity and number of song played.

**Improvements**

Future improvements could include

* incorporating additional user demographic data to enrich the feature set
* exploring deep learning models for potentially better performance
* further EDA and feature engineering to add additional features such as users’ activity frequency, users’ location, songs that users most frequently listen to
* continuously updating the model with new user data to maintain its relevance and accuracy

**Acknowledgment**

We acknowledge Udacity for providing the dataset and project framework. Special thanks to the Sparkify team for their support and the open-source community for the tools and libraries that facilitated this project.