Customer Churn Prediction pada Streaming Musics Platform menggunakan Ensemble Learning

Iqbal Saviola Syah bill haq[1], Tjokorda Agung Budi Wirayuda[2]

[1,2]Fakultas Informatika, Universitas Telkom, Bandung
[1]iqbalsaviola@students.telkomuniversity.ac.id,
[2]cokagung@telkomuniversity.ac.id

Abstrak

***Churn prediction* sangat penting bagi layanan berbasis *subscriptions* seperti KKBOX, yang mana merupakan sebuah *streaming music platform* terkenal di Asia. Meskipun terkenal, KKBOX menghadapi tantangan signifikan dengan *churn customer*, di mana ketika pelanggan membatalkan *subscriptions* mereka, yang berdampak langsung pada pendapatan dan pertumbuhan perusahaan. Penelitian ini mengeksplorasi pengembangan model *churn prediction* menggunakan *ensemble machine learning*.**

***Churn prediction* membantu mengidentifikasi pelanggan yang kemungkinan akan membatalkan *subscriptions* mereka, memungkinkan perusahaan untuk menerapkan *retention strategies*. Pentingnya topik ini terletak pada implikasi finansial dan pertumbuhan jangka panjang bagi bisnis. *Churn predicition* yang efektif dapat secara signifikan meningkatkan *retention customers*, karena mempertahankan hanya 5% dari pelanggan yang ada dapat meningkatkan keuntungan sebesar 25% hingga 95%.**

Penelitian ini menggunakan dataset dari KKBOX dan mengimplementasikan berbagai model *machine learning*, termasuk logistic regression, SVM, XGBoost, dan LightGBM, untuk memprediksi *churn*. Solusi ini melibatkan data exploration, data preparation, feature engineering, untuk meningkatkan *model accuracy*.

Pada experiment ini LightGBM unggul dibanding model lainnya, dengan mencapai skor log loss terendah. Model-model ini menyediakan *framework* yang kuat untuk *churn prediction*, dapat meningkatkan *retention strategies customers* untuk *subscription-based services* seperti KKBOX. Experiment selanjutnya dapat mengeksplorasi *features* lainnya dan *tuning hyperparameter* untuk lebih meningkatkan *model performances*.

Kata kunci : Churn Prediction, XGBoost, LightGBM, Ensemble learning, SVM, Logistic Regression

Abstract

Churn prediction is crucial for subscription-based services like KKBOX, a leading streaming music platform in Asia. Despite its popularity, KKBOX faces significant challenges with customer churn, where customers cancel their subscriptions, directly impacting the company's revenue and growth. This research explores the development of a churn prediction model using ensemble machine learning.

Churn prediction helps identify customers who are likely to cancel their subscriptions, allowing the company to implement retention strategies. The importance of this topic lies in its financial implications and long-term growth for the business. Effective churn prediction can significantly enhance customer retention, as retaining just 5% of existing customers can increase profits by 25% to 95%.

This research uses a dataset from KKBOX and implements various machine learning models, including logistic regression, SVM, XGBoost, and LightGBM, to predict churn. The solution involves data exploration, data preparation, and feature engineering to improve model accuracy.

In this experiment, LightGBM outperforms other models, achieving the lowest log loss score. These models provide a robust framework for churn prediction and can enhance customer retention strategies for subscription-based services like KKBOX. Future experiments could explore additional features and hyperparameter tuning to further improve model performance.

Keywords: Churn Prediction, XGBoost, LightGBM, Ensemble learning, SVM, Logistic Regression

1. Pendahuluan

KKBOX, didirikan di Taiwan pada tahun 2005, telah menjadi *leading streaming music platform* di Asia, menampilkan lebih dari 40 juta *track* dan mencapai lebih dari 10 juta *customers* di Taiwan, Hong Kong, Jepang, Singapura, dan Malaysia. Meskipun sukses, KKBOX menghadapi tantangan signifikan terkait dengan *"churn" customers*, di mana *users* menghentikan *subscription* mereka, yang berdampak langsung pada *revenue* dan *growth*. Memahami dan memprediksi *churn customers* penting bagi layanan berbasis *subscription* seperti KKBOX. *Paper* ini mengeksplorasi pengembangan *model churn prediction* menggunakan *machine learning*.

Churn, merupakan fenomena penting dalam perusahaan-perusahaan yang bergantung pada customer sebagai *main asset* mereka. Ketika *customers* merasa *dissatisfied* dengan *product* atau *services* yang ditawarkan, mereka cenderung untuk berhenti menggunakan *product* tersebut. *High Churn rate* dapat mengakibatkan penurunan *growth* perusahaan secara keseluruhan. Oleh karena itu, penting bagi perusahaan untuk dapat memprediksi *customer churn behavior* dengan *accurate* agar dapat mengambil *strategic decision* yang sesuai untuk mempertahankan *customer*.

Churn prediction melibatkan identifikasi pengguna yang berpotensi untuk membatalkan *subscription* mereka, memungkinkan perusahaan untuk menerapkan *rentention strategy*. *Churn prediction* yang efektif dapat secara signifikan meningkatkan kemampuan perusahaan untuk mempertahankan pelanggan [1]. Penelitian ini menggunakan dataset KKBOX yang juga digunakan untuk competition "WSDM - KKBox's Churn Prediction Challenge". Pendekatan ini didukung oleh studi Huang dkk. yang menekankan pentingnya data komprehensif dalam meningkatkan *accuracy prediction* [2]. *Churn prediction* merupakan masalah *binary classification problem* yang bertujuan untuk menentukan apakah seorang *customers* akan *churn* atau tidak. Dalam konteks ini, *churn* dapat didefinisikan sebagai *customers* yang tidak memperbaharui *subscriptions* mereka dalam periode tertentu setelah masa *subscriptions* mereka berakhir.

Metodologi yang digunakan dalam penelitian ini melibatkan *data exploration* dan *data preparation*, diikuti dengan penerapan berbagai model machine learning, termasuk logistic regression, SVM, dan metode ensemble seperti XGBoost dan LightLGM. Mengkomparasikan model yang berbeda dapat memberikan gambaran masing-masing kelemahan dan keunggulan, yang mana menjadi salah satu tujuan riset ini [3]. Selain itu, perlu juga untuk memperhatikan *feature engineering* dan bagaimana mengatasi *data imbalances* untuk meningkatkan *model's accuracy* [4]. Dengan menerapkan teknik-teknik canggih ini, penelitian ini bertujuan untuk membangun *model churn prediction* yang dapat diterapkan secara *practical* dalam *business operation streaming music platform*. Nilai *machine learning* dalam sektor *telecomunication*, yang memiliki karakteristik serupa dengan industri *streaming music* [5].

Secara keseluruhan, paper ini memberikan kontribusi bagi bidang analitik pelanggan dengan menyediakan kerangka kerja untuk *churn prediction* yang dapat membantu layanan berbasis langganan seperti KKBOX meningkatkan *user retentions* dan mendorong *growth*. Temuan ini diharapkan relevan tidak hanya pada case seperti dataset KKBOX tetapi juga bagi sektor lain yang menyediakan *digital content*. Vafeiadis dkk. dan Tsai & Lu lebih lanjut menekankan *potential impact* dari model *prediction* semacam ini di berbagai industri [6,7].

1.1 Latar belakang

Masalah *churn* bukan hanya masalah *finances* saja, tetapi juga memiliki *impact* besar pada *growth* jangka panjang sebuah perusahaan. Penelitian dari Harvard Business School menunjukkan bahwa menjaga hanya 5% *customers retentions* yang ada dapat menghasilkan peningkatan *profitability* yang signifikan, hingga 25% hingga 95% [8]. Dengan demikian, mengurangi tingkat *churn* dapat meningkatkan *profitability* yang substansial bagi perusahaan. Pemilihan topik ini sangat relevan dalam konteks saat ini karena semakin banyak perusahaan yang berusaha memanfaatkan *data* untuk meningkatkan *strategy customer retentions* mereka. Namun, masih terdapat kesenjangan antara pemahaman konseptual tentang *churn* dan implementasi praktisnya. Oleh karena itu, *experiment* ini akan mengisi celah ini dengan mengembangkan model *churn prediction* yang dapat membantu perusahaan dalam memutuskan langkah-langkah strategis berdasarkan data pelanggan mereka.

Studi-studi terdahulu telah menunjukkan berbagai pendekatan untuk memodelkan dan memprediksi *churn.* Sebagai contoh, pada studi oleh Liao & Chen, mereka menggunakan pendekatan *machine learning* untuk mengidentifikasi faktor-faktor yang mempengaruhi *churn customers* [9]. Begitu juga, penelitian oleh Verbeke menunjukkan bahwa *predictive model* dapat secara signifikan meningkatkan akurasi dalam mengidentifikasi pelanggan yang berpotensi melakukan *churn* [10].

Dengan menggunakan algoritma *machine learning* dan *data analytics* pada *customers historical data*, ekperimen ini akan mengembangkan model *churn prediction* yang dapat membantu perusahaan untuk lebih efektif dalam menjaga dan meningkatkan basis *customers*. Pemilihan metode ini didasarkan pada kemampuan algoritma *machine learning* untuk menangani *volume* data yang besar dan kompleksitas dari masalah *classification churn*. Contohnya metede seperti *Ensemble methods*, seperti *bagging* dan *boosting*, dapat mengurangi bias dan variance, dan mengasilkan performances yang lebih baik pada *cases imbalanced data*. *Bagging* misalnya, dapat mengurangi *variance* dengan melakukan *averaging* dari beberapa *models*, yang dapat mencegah *overfitting* [11]. Boosting mengurangi *bias* dengan berfokus kepada data yang sulit di klasifikasikan, yang mana sering melibatkan *minority class samples* [12]. Model dari ensemble learning dipilih untuk jadi acuan pada *experiment* ini.

1.2 Topik dan Batasannya

Competition yang semakin ketat dalam *industry streaming music* pada kasus ini KKBOX, telah menuntut platform untuk secara efektif mempertahankan customer mereka. Pada Data train dan test yang ada, terdapat class imbalances dimana *class churn* sangat rendah. Untuk mengatasi masalah ini, sangat penting untuk mengembangkan model churn prediction yang secara akurat melakukan prediction terhadap customers behavior – khususnya pada dataset yang mengalami *class imbalances* pada class target– agar dapat mengidentifikasi customers yang kemungkinan akan akan melakukan churn.

Paper ini bertujuan untuk membangun *model churn prediction* menggunakan pendekatan ensemble learning seperti XGBoost,LightBGM dan metode *traditional machine learning* lainnya pada *data customers* dari KKBOX. Dataset yang digunakan dalam penelitian ini mencakup table members, transaction, dan user logs. Input utama untuk model ini mencakup *feature* yang behubungan dengan *customers* seperti registration_date, membership_expired_date, music_activity_per_users, dan auto_renew. Output dari sistem ini akan berupa *prediction probability of churn* (berupa angka antara 0 dan 1). *Performance model* akan dievaluasi menggunakan log loss function, untuk memastikan keandalan dan efektivitasnya.

Beberapa *constraints* harus dipertimbangkan dalam penelitian ini. Dataset yang digunakan pada data train dan test yang tersedia di WSDM *competition* hanya terbatas pada periode April dan Mei 2017. Selain itu, karena keterbatasan *hardware*, hanya 1-5 juta *record* dari tabel user_logs yang digunakan dari total 300 juta *record* yang ada. Untuk memastikan perbandingan performa yang lebih adil dan objektif, 3 nilai hyperparameter digunakan per model.

1.3 Tujuan

Dalam penelitian ini, penulis membangun model churn prediction menggunakan *ensemble learning* dan metode lainnya untuk case *streaming music platform* KKBOX. Model ini bertujuan untuk memprediksi *customer behaviors*, apakah mereka akan *churn* atau tidak. Model XGBoost dan LightGBM diimplementasikan sebagai metode utama karena kemampuannya dalam menangani dataset yang besar dan kompleks serta memberikan hasil yang baik. Selain itu, kami juga akan mengevaluasi beberapa metode lain seperti Logistic Regression, SVM, dan LightLBM sebagai pembanding. *Performance* model-model ini akan dievaluasi menggunakan metric log loss, yang mana baik untuk *case imbalance data.*

Table 1: Keterkaitan tujuan riset paper, metrics, hipotesis

No	Tujuan	Metrics	Hipotesis
1	Membangun dan mengimplementasikan model *churn prediction* menggunakan Ensemble Learning dan metode lainnya untuk KKBOX.	Log Loss: Mengukur seberapa dekat prediciton dengan actual value, (dengan *higher penalties* untuk prediction yang salah) metode lainnya.	Model Ensemble Learning diharapkan menunjukkan kinerja terbaik dalam memprediksi churn dengan akurasi tertinggi dan log loss terendah dibandingkan dengan metode lain.
2	Mengidentifikasi fitur-fitur paling signifikan yang mempengaruhi churn pada model Ensemble Learning.	Importance scores dari features didalam model Ensemble Learning	Fitur yang digunakan dalam prediksi model diharapkan memiliki perilaku tertentu sesuai dengan metode (XGBoost dan LightGBM).

1.4 Organisasi Tulisan

Bagian awal dari penelitian ini akan membahas terkait Preprocessing dataset KKBOX, dan framework model. Pada bagian kedua, hasil eksperimen dari berbagai model yang digunakan akan dianalisis dan dibandingkan. Kesimpulan dari penelitian ini akan merangkum temuan model terbaik untuk kasus ini dan *future works* dalam konteks *streaming music platforms churn prediction*.

2. Studi Terkait

Metode *ensemble learning*, seperti XGBoost dan LightGBM, telah terbukti lebih unggul dibandingkan dengan algoritma *machine learning* tradisional *prediction task*. Metode *ensemble* menggabungkan beberapa *machine learning* untuk meningkatkan *generalizability* dan *robustness* model. Mereka mengurangi *trade-off bias-variance* dengan meratakan atau menggabungkan prediction dari model yang berbeda, sehingga mengurangi *overfitting* dan meningkatkan *prediction performaces* [13]. Di konteks *Churn Prediction* untuk *subscription-based services* seperti KKBOX, metode *ensemble* sangat baik dalam menangani hubungan yang kompleks dan *nonlinearities* yang ada dalam dataset berskala besar, yang krusial untuk mengidentifikasi dengan akurat *customers* yang berisiko untuk *churn*.

XGBoost dan LightGBM telah menjadi model yang baik untuk *churn prediction* karena efisiensinya dalam menangani dataset berskala besar dan kemampuannya untuk menangkap *dependencies* yang rumit antar *features*. XGBoost mengoptimalkan *computatition* melalui *parallel* dan *distributed computing*, menjadikannya *scalable* bahkan untuk dataset yang sangat besar [14]. LightGBM, di sisi lain, menggunakan *gradient-based approach*

untuk *decision tree splitting* dan *training* yang lebih cepat serta penggunaan *memory* yang lebih rendah dibandingkan dengan *traditional gradient boosting methods* [15]. Hal ini membuat kedua model tersebut sangat cocok untuk *churn prediction tasks* di mana ukuran dataset, *computational efficiency* dan *model interpretability* menjadi faktor yang penting.

Log loss banyak digunakan dalam *model churn prediction* karena mengukur *accuracy* dengan *probabilistic predictions.* Berbeda dengan *accuracy* yang hanya mempertimbangkan *correctness* dari *class labels*, *log loss* mengevaluasi c*onfidence* dari sebuah *predictions* dengan *penalizing models* yang untuk *predictions* yang salah. *Metric* ini sejalan dengan *business objective* untuk memaksimalkan *revenue* melalui *identification* yang akurat terhadap potensial *churners*, sehingga dapat menghasilkan *retention efforts* yang *personlized* [16].

2.1 Log Loss

Log loss, atau logarithmic loss, merupakan performance metric yang umumnya digunakan untuk mengevaluasi keakuratan *probabilistic* dari sebuah *classifiers*. *Log loss* mengukur *prediction* yang dibuat oleh model dengan menghukum *classifications* yang salah dengan nilai *loss* yang tinggi. Semakin kecil nilai *log loss,* maka semakain baik *model* dalam melakukan prediksi *probabilities* dari setiap *records*. Menurut Chen et al. (2019), Log loss memberikan gambaran evaluasi lebih baik, dimana *metric* ini mengukur seberapa jauh *prediction* dari *actual value*, bukan hanya *correctness* dari model. Log Loss didefinisikan dengan *equation* (1) berikut.

$$Log\ Loss\ (y, \hat{y})\ =\ -\frac{1}{N}\sum_{i=1}^{N}[y_i\ log\ (\hat{y})\ +\ (1-y_i)\ log\ (1\ -\hat{y}_i)] \quad (1)$$

2.2 Accuracy

Accuracy adalah salah satu *metrics* yang paling sederhana untuk mengevaluasi *model classification*. Accuracy didefinisikan sebagai rasio antara *instance* yang diprediksi dengan benar dan jumlah total *instance*. Meskipun Accuracy adalah *metrics* yang populer, walaupun *metrics* ini mungkin bukan pilihan terbaik untuk dataset yang sedang menghadapi *class imbalances*. Sebuah studi oleh Kelleher et al. (2020) menekankan bahwa Accuracy bisa menyesatkan ketika dataset memiliki *class imbalances* yang signifikan, karena dapat mengabaikan minority class. Accuracy didefinisikan dengan *equation* (2) berikut.

$$Accuracy\ =\ \frac{TP}{TP+TN+FP+FN} \quad (2)$$

2.3 Precision

Precision adalah rasio antara *true positive* dan jumlah total *positive prediction*, baik yang benar *(true positive*) maupun yang salah *(false positive)*. *Metrics* ini mencerminkan akurasi dari *positive prediction* yang dibuat oleh model. Precision yang tinggi berarti model membuat sedikit kesalahan *(false positive)*. Chawla et al. (2018) berpendapat bahwa Precision sangat penting dalam implementasi dimana *false positives* sangat tinggi. Precision didefinisikan dengan *equation* (3) berikut.

$$Precision\ =\ \frac{TP}{TP+FP} \quad (3)$$

2.4 Recall

Recall, yang juga dikenal sebagai sensitivitas atau *true positive rate*, mengukur rasio antara *predicted positive observations* yang diprediksi dengan benar dan semua *actual positives*. Recall yang tinggi menunjukkan bahwa model berhasil menangkap sebagian besar *positive case*. Menurut López et al. (2019), recall adalah *metrics* penting dalam skenario di mana kehilangan *positive cases* dapat berdampak serius, seperti dalam *fraud detection.* Recall didefinisikan dengan *equation* (4) berikut.

$$Recall\ =\ \frac{TP}{TP=FN} \quad (4)$$

2.5 F1 Score

F1-score adalah rata-rata *harmonic* dari precision dan recall, yang memberikan keseimbangan antara keduanya. *Metrics* ini sangat berguna ketika berhadapan dengan *imbalances* dataset, karena mempertimbangkan baik *false positive* maupun *false negative*. Powers (2020) menjelaskan bahwa F1-score adalah *metrics* yang lebih informatif ketika diperlukan satu angka untuk menyampaikan keseimbangan antara precision dan recall. F1 Score didefinisikan dengan *equation* (5) berikut.

$$F1\ Score\ =\ 2\cdot\frac{Precision\cdot Recall}{Precision\ +\ Recall} \quad (5)$$

3. Sistem yang Dibangun

Pada experiement ini, system terbagi menjadi beberapa bagian, yaitu Preprocessing dan Modelling. Pada tahap

preprocessing, feature engineering dilakukan untuk memersiapkan data train dan test model. Setelah itu, kami melakukan model fitting pada 4 model dan mengukur log loss dari masing-masing model, untuk mendapatkan model terbaik.

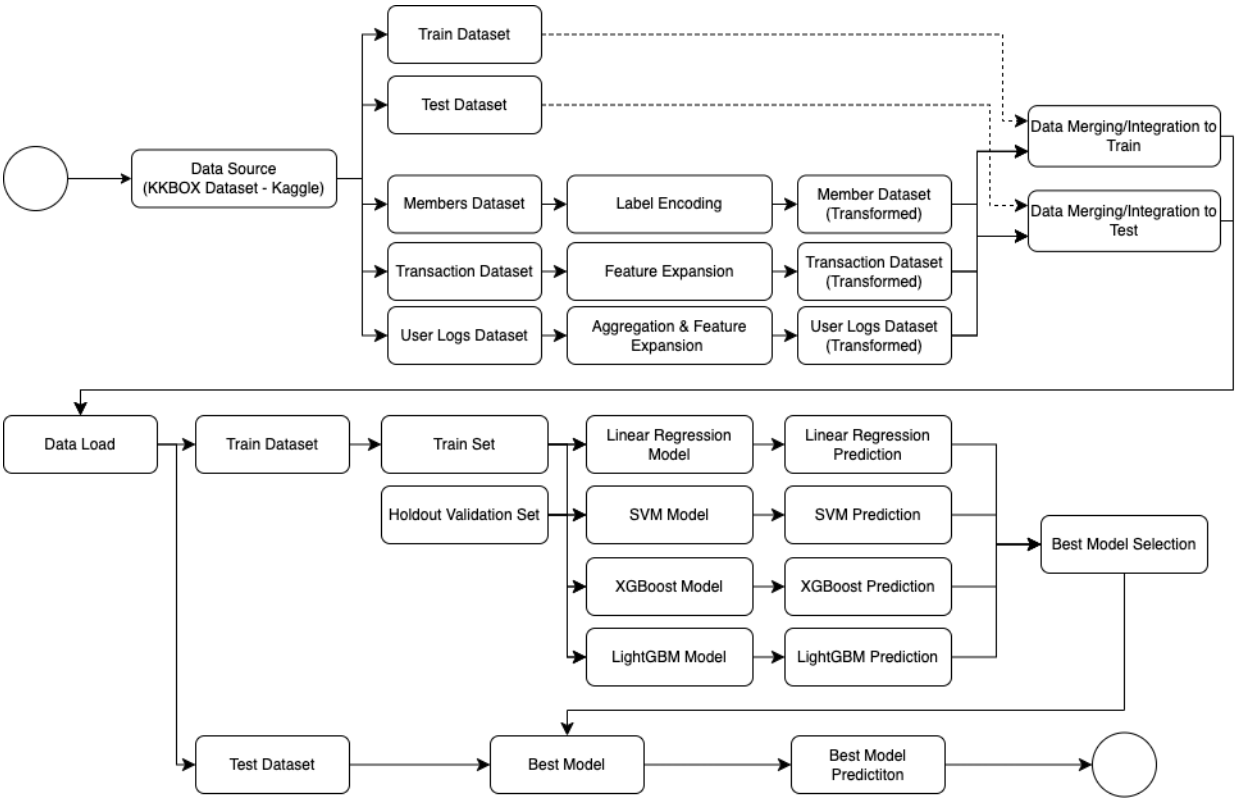

Figure 1: System Flow Preprocessing and Modelling

3.1 *Data Sources / Acquisition*

Dataset yang tersedia pada Kaggle terdiri dari 5 bagian, yaitu Train, Test, Members, Transaction, dan User Logs. Data ini berasal dari dataset KKBOX yang tersedia di Kaggle, digunakan untuk memprediksi *churn* dalam *streaming music platform*. Pada bagian ini, kami melakukan Exploratory Data Analysis (EDA) untuk memahami struktur fitur setiap tabel dataset dan mengevaluasi presentase *churn* dari setiap dataset yang akan menjadi input model.

3.2 *Data Preprocessing*

Pada tahap ini, kami melakukan feature engineering untuk pada 3 table, yaitu members, Transactions, dan user logs.

3.2.1 User Logs Transformation

Pada dataset members dilakukan *Label encoding*, untuk mengubah *feature categoric*al seperti *gender (male, female)* menjadi *numerical value* (1,2) agar *model machine learning* dapat memprosesnya.

Table 2: Members dataset sample, and label encoding gender

msno	city	bd	gender	Transformed gender	registered_via	registration_init_time
Rb9UwLQTrxzBV	1	0	NaN	0	11	20110911
tJonkh+O1CA796	1	0	NaN	0	7	20110914
cV358ssn7a0f7jZO	1	0	NaN	0	11	20110915
9bzDeJP6sQodK73	1	0	NaN	0	11	20110915
WFLY3s7z4EZsie	6	32	female	2	9	20110915

Pada table 1, dapat dilihat transformed gender diubah dari “female” menjadi “2”.

3.2.2 Transaction Transformation

Pada dataset *transactions* dilakukan *Feature Expansion* adalah proses yang melibatkan pembuatan *features* baru (*is_discount* dan *membership_duration*) dari *feature* yang sudah ada untuk memberikan lebih banyak informasi dan berpotensi meningkatkan *performance* model machine learning.

Table 3: Transactions dataset sample, and feature expansions

msno	payment_ method_id	payment_ plan_days	plan_list _price	actual_amount_ paid	is_auto_ renew	transaction_ date	membership_expire _date	is_discont	membershp_ duration
YyO+tlZ tAX	41	30	129	129	1	20150930	20151101	0	171
AZtu6Wl 0gPo	41	30	149	149	1	20150930	20151031	0	101
UkDFI97 Qb6	41	30	129	129	1	20150930	20160427	0	9497
M1C56ij xozN	39	30	149	149	1	20150930	20151128	0	198
yvj6zyB Uaqdb	39	30	149	149	1	20150930	20151121	0	191

Seperti dapat dilihat pada table 3, terdapat feature “is_discount” yang didapat dari selisih feature “actual_amount_paid” dan “plan_list_price”, dan feature “membership_duration” yang didapat dari selisih feature “membership_expire_date” dan “transaction_date”.

3.2.3 User Logs Transformation

Pada dataset user logs, dilakukan *aggregating* dan pembuatan *features* hasil *aggregation.* Hal in dilakukan dengan *aggregate function* pada *record user logs*, dan menghitung beberapa *statistic* per pengguna (seperti sum, count, std, mean, min, dan max) untuk berbagai kolom (num_25, num_50, num_75, num_985, num_100, num_unq, total_sec). Proses ini sangat penting untuk mengubah data mentah menjadi format terstruktur, yang bagus untuk *model machine learning*. Pada *experiment* ini, digunakan chunking untuk 5 juta data dari 300 juta data. Table 3 dibawah ini menujukkan kondisi awal dataset user logs, dan table 5 adalah hasil *aggregation*.

Table 4: User Logs dataset sample

msno	date	num_25	num_50	num_75	num_985	num_100	num_unq	total_secs
u9E91QDTvH	20170331	8	4	0	1	21	18	6309.273
nTeWW/eOZA	20170330	2	2	1	0	9	11	2390.699
2UqkWXwZbI	20170331	52	3	5	3	84	110	23203.337
ycwLc+m2O0	20170331	176	4	2	2	19	191	7100.454
EGcbTofOSOk	20170331	2	1	0	1	112	93	28401.558

Table 5: Aggregating and feature expansion sample

msno	date_min	date_max	num_25_sum	num_25_count	num_25_mean	num_50_sum	num_50_mean	num_75_sum
rxIP2f2aN0r	20150326	20150716	75	13	5.769231	11	0.846154	6
yxiEWwE9V	20150105	20170111	315	78	4.038462	119	1.525641	82
PNxIsSLWOJ	20151201	20170125	18	18	1	9	0.5	10
KXF9c/T66LZ IzFq+xS64i	20150803	20170201	89	24	3.708333	28	1.166667	14
oy2721XIrBu	20150205	20160727	167	60	2.783333	90	1.5	83

3.3 *Developing Predictive Models*

Pada tahap ini, kami melakukan fitting model menggunakan dataset yang telah diproses sebelumnya. Kami menggunakan empat algoritma *supervised machine learning* untuk memprediksi variabel *is_churn*, yaitu XGBoost, Linear Regression, SVM, dan LightGBM. Proses ini meliputi membangun split holdout validation, mendefinisikan *parameters default* untuk masing-masing *model*, dan melakukan training setiap model menggunakan *data* yang telah *preprocessing*. 4 Model ini akan dikomparasikan satu sama lain menggunakan *metrics Log Loss* untuk mengukur *probabilistic prediciton, accuracy, precision, recall,* dan *F1-Score*. Hasil terbaik dari semua *metric* akan dipilih untuk melakukan *prediction dataset test*.

4. Evaluasi

Dalam studi ini, kami bereksperimen dengan empat *machine learning algorithms*: XGBoost, LightGBM, logistic regression, and Support Vector Machine (SVM) (i.e. pada kasus ini menggunakan Linear SVC, untuk mempersingkat computation).

4.1 Hyperparameter

Dalam konteks logistic regression digunakan parameter *alpha* (0.0001, 0.001, 1), dimana biasanya merujuk pada *regularization* parameter. Parameter ini mengontrol jumlah *regularization* yang diterapkan untuk mencegah *overfitting*. *Regularization* membantu menjaga model tetap sederhana dan lebih dapat digeneralisasi dengan memberikan penalti pada koefisien yang besar. Pada experiment ini, skor log loss terendah didapat ketika menggunakan *regularization* yang sangat kecil ($\alpha = 0.0001$).

Pada model SVM Linear digunakan parameter C (0.01, 1, 10). Parameter ini mengontrol *trade-off* antara *margin* yang besar dan *accuracy classification* pada data *training*. C yang lebih kecil mendorong margin yang lebih besar (model yang lebih sederhana), sementara C yang lebih besar bertujuan untuk mengklasifikasikan semua contoh pelatihan dengan benar. Skor log loss rendah dengan menggunakan C = 10, yang menunjukkan bahwa nilai C yang lebih tinggi, yang memungkinkan model lebih fokus pada mengklasifikasikan *instances* dengan benar.

Pada model XGBoost, parameter yang digunakan yaitu *Learning Rate* (3, 7, 11), *Max Depth* (0.01, 0.1, 0.3), dan *Min Child Weight* (1, 5, 7). *Learning Rate*, mengontrol seberapa besar *learning step* pada setiap iterasi saat bergerak menuju minimum dari *loss fuction*. Nilai yang lebih kecil membuat model lebih robust tetapi memerlukan lebih banyak pohon. *Max Depth*, yaitu maksimum kedalaman *tree*. Menaikkan nilai ini membuat model lebih kompleks dan mampu mempelajari pola yang lebih detail, tetapi juga lebih rentan terhadap *overfitting*. *Min Child Weight*, merupakan jumlah minimum dari bobot setiap *instance (hessian)* yang dibutuhkan di sebuah *child*. Nilai yang lebih tinggi mencegah model dari mempelajari *pattern* yang mungkin sangat spesifik untuk sampel *instances* tertentu yang dipilih untuk sebuah *tree*. Log loss terbaik dicapai oleh kombinasi Learning Rate: 0.1, Max Depth: 7, dan Min Child Weight: 1, dimana parameter ini membantu dalam menangkap pola yang diperlukan tanpa mengalami *overfitting*.

Pada model LightGBM, digunakan parameter *Learning Rate* (0.01, 0.1, 0.2), *Max Depth* (3, 7, 15), *Min Child Samples* (5, 10, 20). *Learning Rate*, mirip dengan XGBoost, parameter ini mengontrol seberapa besar *learning step* langkah pada setiap iterasi. Nilai yang lebih kecil memerlukan lebih banyak iterasi. *Max Depth*, Kedalaman maksimum dari *tree*. Nilai yang lebih tinggi membuat model lebih kompleks. *Min Child Samples*, Jumlah minimum *data point* yang dibutuhkan di setiap *leaf*. Parameter ini digunakan untuk mengontrol *overfitting*. Log loss terbaik dicapai oleh kombinasi Learning Rate: 0.1, Max Depth: 15, Min Child Samples: 20. Kombinasi ini mencapai log loss terendah, yang menunjukkan bahwa ini adalah parameter dengan kinerja terbaik, seimbang antara kompleksitas model dan *generalization.*

4.2 Hasil Komparasi Experiment

Metode *evaluation metric* yang digunakan untuk membandingkan *performances* model-model ini adalah *log loss*, yang mengukur *accuracy* dari *probabilistic predictions*. Nilai log loss yang lebih rendah menunjukkan *performances* model yang lebih baik. Evaluasi dilakukan pada *Train set* dan *holdout validation set* untuk melihat perfomances *prediction* dari setiap model.

Table 6: Algorithm performances on Log Loss scoring

Algorithm Method	**Log loss Scoring**	
	Train set	**Holdout Validation set**
XGBoost	0.041	0.049
LightGBM	0.044	0.047
Logistic Regression	0.320	0.319
SVM	0.367	0.367

Pada *table* 6 diatas, penggunaan *holdout validation* digunakan sebagai set yang memperlihatkan bagaimana *perfomances model* pada data yang memiliki class 0 *(not-churn)* dan 1 *(churn)*. LightGBM menunjukkan kinerja superior dengan nilai log loss yang sangat rendah. Hasil ini menunjukkan bahwa algoritma *gradient boosting* sangat cocok untuk dataset dan masalah yang dihadapi karena kemampuannya untuk memodelkan hubungan non-linear pada kasus KKBOX *churn prediction.* Sebaliknya, logistic regression dan SVM menunjukkan nilai log loss yang lebih tinggi, yang mengindikasikan kinerja yang lebih buruk dibandingkan dengan XGBoost dan LightGBM.

Table 7: Algorithm performances on Precison, Recall, F1-Score, and Accuracy

Algorithm Method	**Holdout Validation set**				
	Class	**Precision**	**Recall**	**F1-Score**	**Overall Accuracy**
XGBoost	0	0.99	0.99	0.99	0.98
	1	0.87	0.89	0.88	
Light GBM	0	0.99	0.99	0.99	0.98
	1	0.88	0.91	0.89	
Logistic Regression	0	0.91	1.00	0.95	0.91
	1	0.00	0.00	0.00	
SVM	0	0.96	0.99	0.98	0.95
	1	0.89	0.57	0.69	

Pada *holdout validation data*, (di mana terdapat kelas 0 dan 1) XGBoost dan LightGBM menunjukkan kinerja yang mirip dan *superior* dengan *precision*, *recall*, dan *F1-score* yang tinggi untuk kedua kelas (0 dan 1), serta *accuracy* sebesar 0.98. Logistic Regression berkinerja baik untuk kelas 0 tetapi gagal untuk memprediksi kelas 1, dengan *F1-score* 0.00. SVM menunjukkan kinerja yang kurang baik untuk kelas 1 dengan *F1-score* yang jauh lebih rendah, yaitu 0.69.

Secara keseluruhan, LightGBM adalah model yang paling andal dalam perbandingan ini, sementara Logistic Regression dan SVM menunjukkan keterbatasan, terutama dalam menangani ketidakseimbangan kelas. Pada test set, LightGBM akan digunakan untuk melakukan *final predictions* karena memberikan Log loss terendah pada *holdout validation set*, dan nilai *Accuracy* paling tinggi.

4.3 Analisis Hasil Test Set

Table 8: LightGBM performances on both Test and Validation set

	Test Dataset			**Holdout Validation Set**		
Class	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
0	1.00	0.96	0.98	0.99	0.99	0.99
1	0.00	0.00	0.00	0.88	0.91	0.89
Accuracy			0.96			0.98
Log loss		0.1068			0.0470	

Pada table 8, LightGBM menunjukkan kinerja yang baik pada Test Dataset untuk Kelas 0 (*Non-Churned Customers*), dengan precision sebesar 1.00, recall sebesar 0.96, dan F1-Score sebesar 0.98. Namun, model ini sepenuhnya gagal memprediksi Kelas 1 (*Churned Customers*), seperti yang ditunjukkan oleh precision, recall, dan F1-Score yang semuanya 0.00. Sebaliknya, pada Holdout Validation Set, yang mengandung kedua kelas (0 dan 1), kinerja model lebih seimbang di kedua kelas, dengan Kelas 1 mendapatkan F1-Score sebesar 0.89, dan Kelas 0 yang mendekati 0.99.

Perbedaan dikarenakan distribusi *class* yang berbeda antara dataset. *Test Dataset* hanya terdiri dari *instance class* 0, yang menyebabkan LightGBM mengoptimalkan secara eksklusif untuk *class* ini, tetapi gagal untuk *class* 1. Sebaliknya, *Holdout Validation Set* mencakup distribusi yang lebih seimbang dari kedua *class*, memungkinkan LightGBM untuk menunjukkan performa dalam memprediksi pelanggan yang *churned* dan yang tidak secara efektif. *Score* 0 untuk *class* 1 pada *Test set* terjadi karena *model* tidak memiliki kesempatan untuk belajar atau memvalidasi contoh-contoh dari kelas tersebut dalam *Test Dataset*.

Pada *Test Dataset*, log loss sebesar 0.1068 relatif tinggi, menunjukkan *probability calibration* yang buruk, terutama untuk *class* 1. Karena model tidak memprediksi *instance class* 1, probabilitas untuk kelas ini kemungkinan sangat rendah, yang menyebabkan log loss lebih tinggi meskipun kinerja untuk *class* 0 sangat baik. Sebaliknya, *Holdout Validation Set* memiliki log loss yang jauh lebih rendah sebesar 0.0470, mencerminkan estimasi probabilitas yang lebih baik di kedua kelas. Kemampuan model untuk memprediksi *class* 1 dengan akurat dalam *validation set* berkontribusi pada log loss yang lebih rendah, karena model dapat mengidentifikasi pelanggan yang berhenti berlangganan dan yang tidak.

5. Kesimpulan

Eksperiment kami menunjukkan bahwa LightGBM adalah algorithms yang paling efektif untuk kasus streaming music platform, terutama KKBOX. Dngan nilai *log loss* yang jauh lebih rendah (0.047) dibandingkan dengan XGBoost (0.049) Logistic Regression (0.319) dan SVM (0.367). *Performance* yang baik dari *gradient boosting algorithms* menunjukan performance yang baik dalam menangani *complex patterns* dan *imbalances class data*.

Future work dapat lebih fokus untuk membangun lebih banyak *features* tambahan untuk lebih meningkatkan *model performances*, karena LightGBM telah menunjukkan *performances* yang sangat baik dalam menangani berbagai *features*, dan model bossting cukup baik dalam mengatasi jumlah *feature* yang banyak. Selain itu, *tuning hyperparameters* dari model-model ini dapat menghasilkan hasil yang lebih baik. LightGBM merupakan metode boosting, yang mana rentan untuk *overfitting*. Maka experiment setelah ini juga bisa dilakukan dengan pendekatan *Ensemble Learning* lain seperti *Bagging* dan *Stacking* yang baik dalam mengatasi *overfitting*, yang mana hal tersebut adalah kekurangan dari model *boosting*.

Daftar Pustaka

[1] Verbeke, W., Martens, D., Mues, C., & Baesens, B. (2012). Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications, 38(3), 2354-2364*. Elsevier. DOI: 10.1016/j.eswa.2010.08.023

[2] Huang, S., Ke, W., Chen, J., & Chen, S. (2021). A comprehensive survey on customer churn prediction with big data. *Artificial Intelligence Review, 54, 2757-2811*. Springer. DOI: 10.1007/s10462-020-09867-1

[3] Nguyen, T., Pham, T., & Cao, T. (2015). Predicting customer churn in subscription-based services using machine learning. *International Journal of Information Management, 35(2), 244-253*. Elsevier. DOI: 10.1007/978-981-99-8438-1_26

[4] Lariviere, B., & Van den Poel, D. (2005). Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications, 29(2), 472-484*. Elsevier. DOI: /10.1016/j.eswa.2005.04.043

[5] Amin, A., Anwar, S., Adnan, A., Nawaz, M., Howard, N., Qadir, J., & Hussain, A. (2016). Customer churn prediction in the telecommunication sector using a rough set approach. *Neurocomputing, 237, 242-254*. Elsevier. DOI: 10.1016/j.neucom.2016.12.009

[6] Vafeiadis, T., Diamantaras, K. I., Sarigiannidis, G., & Chatzisavvas, K. C. (2015). A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory, 55, 1-9*. Elsevier. DOI: 10.1016/j.simpat.2015.03.003

[7] Tsai, C. F., & Lu, Y. H. (2009). Customer churn prediction by hybrid neural networks. *Expert Systems with Applications, 36(10), 12547-12553*. Elsevier. DOI: 10.1016/j.eswa.2009.05.032

[8] Reichheld, F. F., & Schefter, P. (2000). The Economics of E-Loyalty. *Harvard Business School Working Knowledge*. Retrieved from https://hbswk.hbs.edu/archive/the-economics-of-e-loyalty

[9] Liao, S. H., & Chen, Y. C. (2017). Predicting customer churn in the insurance industry using data mining techniques. *Expert Systems with Applications, 83, 89-101*. Elsevier.

[10] Verbeke, W., Dejaeger, K., Martens, D., Hur, J., & Baesens, B. (2014). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research, 218(1), 211-229*. Elsevier. DOI: /10.1016/j.ejor.2011.09.031

[11] Breiman, L. (1996). Bagging predictors. *Machine Learning, 24(2), 123-140*. Springer. DOI: 10.1007/BF00058655

[12] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics, 29(5), 1189-1232*. Institute of Mathematical Statistics.DOI: 10.1214/aos/1013203451

[13] Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research, 15(1), 3133-3181*.

[14] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). ACM. DOI: 10.1145/2939672.2939785

[15] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems* (pp. 3146-3154). DOI: 10.5555/3294996.3295074

[16] Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning* (pp. 625-632). ACM. DOI: 10.1145/1102351.1102430