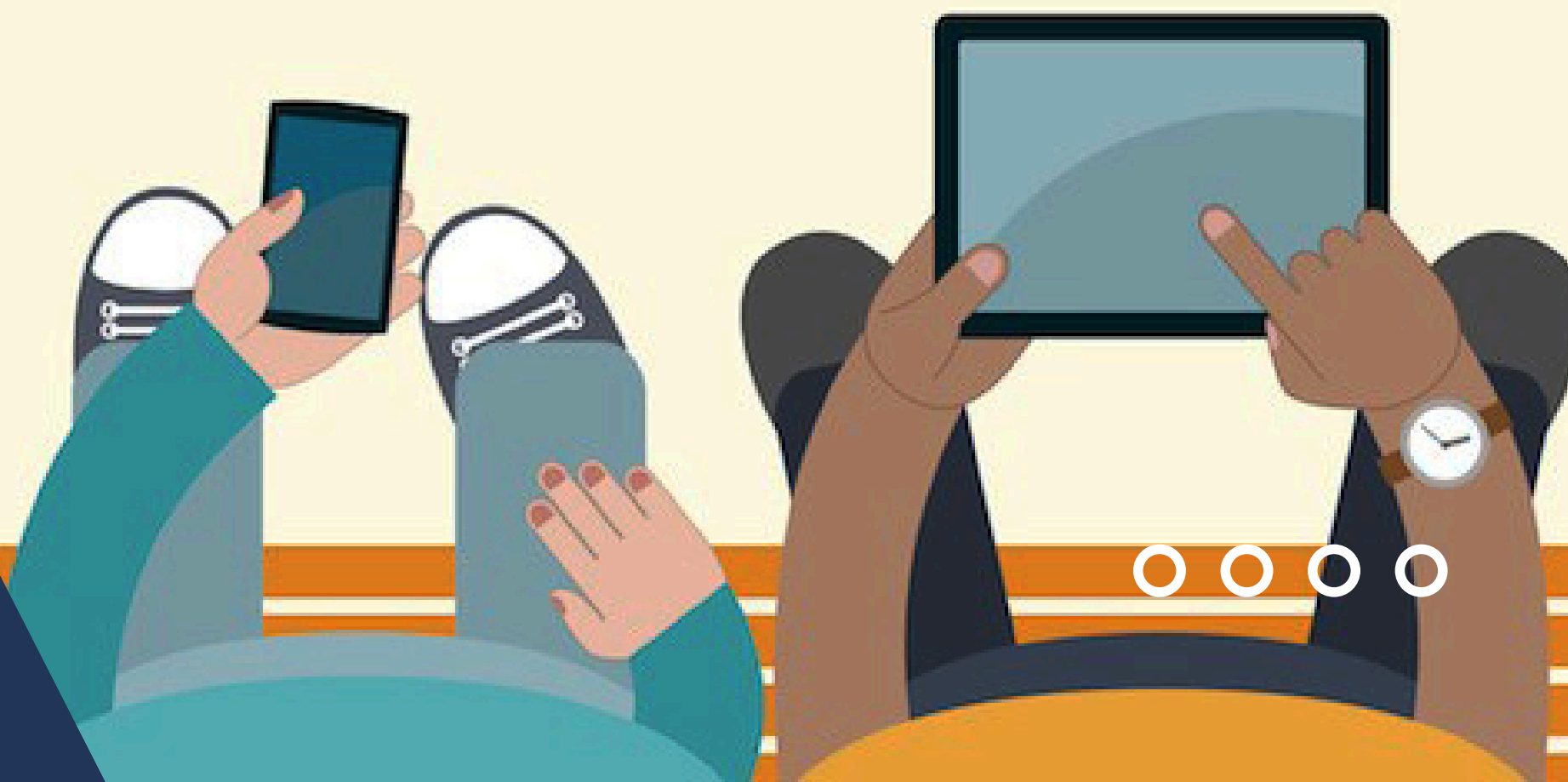


○○○○

TELCO CUSTOMER CHURN

○○○○○



OVERVIEW

Pendahuluan

Analisis Data

**Machine
Learning
Modelling**

**Simulasi Model
prediksi Churn**

**Kesimpulan dan
rekomendasi**



LATAR BELAKANG



Perusahaan telekomunikasi mengalami **tingkat churn pelanggan yang tinggi** dan **kerugian besar** karena strategi promosi yang tidak tepat sasaran. Mereka **mengeluarkan biaya promosi ke semua pelanggan**, tanpa tahu siapa yang berisiko berhenti, sehingga banyak dana promosi yang tidak efisien. Kondisi tersebut mengakibatkan perusahaan mengalami **kerugian sampai \$154,580**

RUMUSAN MASALAH

Perusahaan belum memiliki sistem untuk memprediksi pelanggan yang berpotensi berhenti, sehingga promosi dilakukan secara massal dan menimbulkan biaya besar yang tidak efisien. Dengan **menerapkan machine learning**, perusahaan dapat memprediksi pelanggan berisiko churn dan menargetkan promosi secara lebih tepat, efisien, dan efektif.





METRIC EVALUATION

Cost FP: \$10

Cost FN: \$80

False Positive (FP): Model memprediksi akan churn → beri promosi \$10 Tapi kenyataannya pelanggan tidak akan churn → promosi tidak perlu → rugi \$10

False Negative (FN): Model memprediksi tidak akan churn → tidak beri promosi Tapi kenyataannya pelanggan churn → kehilangan pelanggan → rugi \$80

Karena cost dari FN jauh lebih tinggi dibanding FP, karena itu F2-score dipilih sebagai metrik evaluasi utama



TUJUAN ANALISIS

- Memprediksi Churn Pelanggan
- Mengurangi Kerugian Finansial
- Mengoptimalkan Strategi Promosi
- Menggunakan Metode Evaluasi yang Tepat Memberikan Insight Fitur Penting



DATA AWAL



- Data awal pada file data_telco_customer_churn.csv mempunyai 4930 rows.
- Data memiliki 11 kolom yang terdiri dari 10 kolom feature dan 1 kolom target, kolom target pada data ini adalah Churn.
- Tipe datanya 9 bertipe Kategorikal dan 2 bertipe Numerikal.
- Perbandingan pelanggan yang Churn dan yang tidak Churn adalah 1288 : 3565

KOLOM FEATURE

Dependent

InternetService

OnlineBackup

TechSupport

Tenure

Contract

OnlineSecurity

MonthlyCharges

DeviceProtection

PaperlessBilling



ANALISIS DATA

Perbedaan pelanggan yang Churn **hampir 4 kali** dengan pelanggan yang tidak Churn

Perbandingan persentase pelanggan yang Churn dan yang tidak Churn adalah **27% : 73%**

Dilihat dari Pvalue semua feature mempunyai **kaitan dengan target (Churn)**



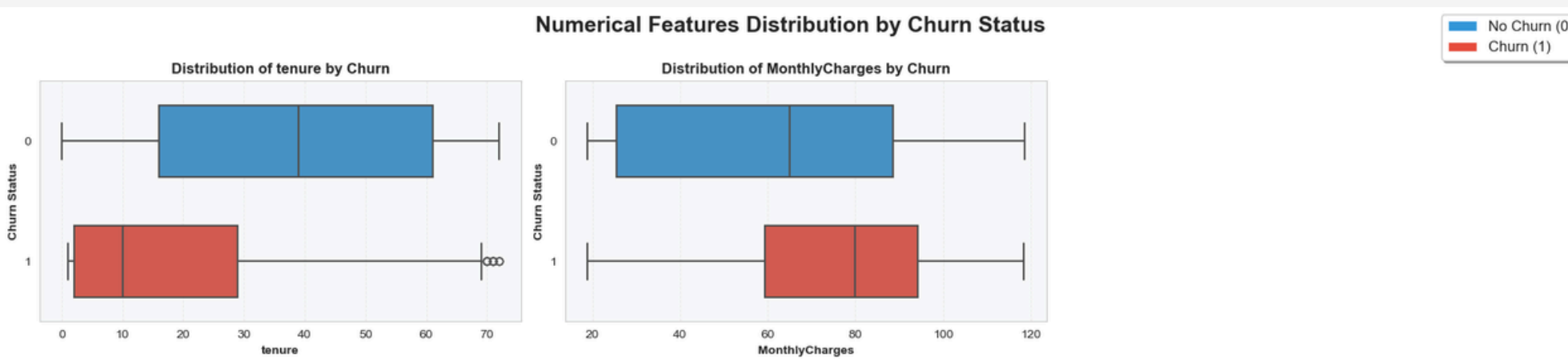


DATA CLEANING

- Tidak ada Missing Value
- Tidak ada Outlier pada kolom tenure dan MonthlyCharges
- Tidak ada value kolom Kategorikal yang dihapus atau digabungkan karena value countnnya sedikit
- Terdapat duplikat data sebanyak 77, maka kita akan menghapusnya karena persentase nya hanya sedikit
- Merubah value kolom target (Churn) dari Yes \rightarrow 1 dan No \rightarrow 0

○ ○ ○ ○

EDA



STATISTICAL COMPARISON: CHURN vs NO CHURN

tenure:

No Churn		Mean:	37.97		Median:	39.00		Std:	23.87
Churn		Mean:	18.03		Median:	10.00		Std:	19.33

MonthlyCharges:

No Churn		Mean:	61.84		Median:	64.95		Std:	30.82
Churn		Mean:	74.95		Median:	80.00		Std:	24.22

- **Tenure:** Pelanggan churn rata-rata hanya 18 bulan, sedangkan pelanggan tetap 38 bulan → risiko churn tinggi pada pelanggan baru.
- **Biaya Bulanan:** Pelanggan churn membayar lebih tinggi ($\pm \$75$) dibanding pelanggan tetap ($\pm \$62$) → sensitif terhadap harga.
- **Strategi:** Fokus retensi pada pelanggan baru & berbiaya tinggi melalui diskon atau program loyalitas.

MACHINE LEARNING

**Define X dan
y**

Data Splitting

Data Preprocess

**Hyper Parameter
Tuning**

Feature Importance

Confusion Matrix

Best Model

**Performance in
Test set**



DEFINE X & Y

Feature (X) : 'Dependents', 'tenure', 'OnlineSecurity', 'OnlineBackup', 'InternetService', 'DeviceProtection', 'TechSupport', 'Contract', 'PaperlessBilling', dan 'MonthlyCharges'

Target (y) : `Churn`

```
X = df.drop(columns="Churn")  
y = df["Churn"]
```



DEFINE X & Y

```
X_train,X_test,y_train,y_test = train_test_split(X,y,  
random_state=0,  
test_size=0.20,  
stratify=y)
```

- **stratify:** Menjaga proporsi distribusi kelas antara training dan testing set agar tetap sama seperti data asli
- **test_size:** Menentukan berapa besar proporsi data yang akan digunakan sebagai test set.
- **random_state:** Menetapkan seed angka acak agar pembagian data selalu konsisten setiap dijalankan (reproducibility).



DATA PREPROCESS

- Encoding : OneHot : 'Dependents', 'OnlineSecurity', 'OnlineBackup', 'InternetService', 'DeviceProtection', 'TechSupport', 'Contract', 'PaperlessBilling'
- Scaling : Robust : 'tenure', 'MonthlyCharges'

Best Model

Model	Mean	Std
logreg	0.708367	0.018856
gbc	0.702178	0.018792
knn	0.636492	0.020035



HYPER PARAMETER TUNING

- **Logreg Best**
- **Best_score: 0.7206181669473259**
- **Best_params: {'resampler': SMOTE(random_state=0), 'model__solver': 'liblinear', 'model__penalty': 'l1', 'model__C': np.float64(0.01)}**

- **GBC Best**
- **Best_score: 0.7295286811244361**
- **Best_params: {'resampler': RandomOverSampler(random_state=0), 'model__subsample': np.float64(0.6), 'model__n_estimators': np.int64(93), 'model__max_features': np.int64(3), 'model__max_depth': np.int64(1), 'model__learning_rate': np.float64(0.34)}**



PERFORMANCE IN TEST SET

Before Tuning

- **Running model: LogisticRegression**
- **F2 Score: 0.7303**

- **Running model: GradientBoostingClassifier**
- **F2 Score: 0.7194**

After Tuning

- **Evaluating: LogisticRegression (Tuned)**
- **F2 Score: 0.7192**

- **Evaluating: GradientBoostingClassifier (Tuned)**
- **F2 Score: 0.7453**

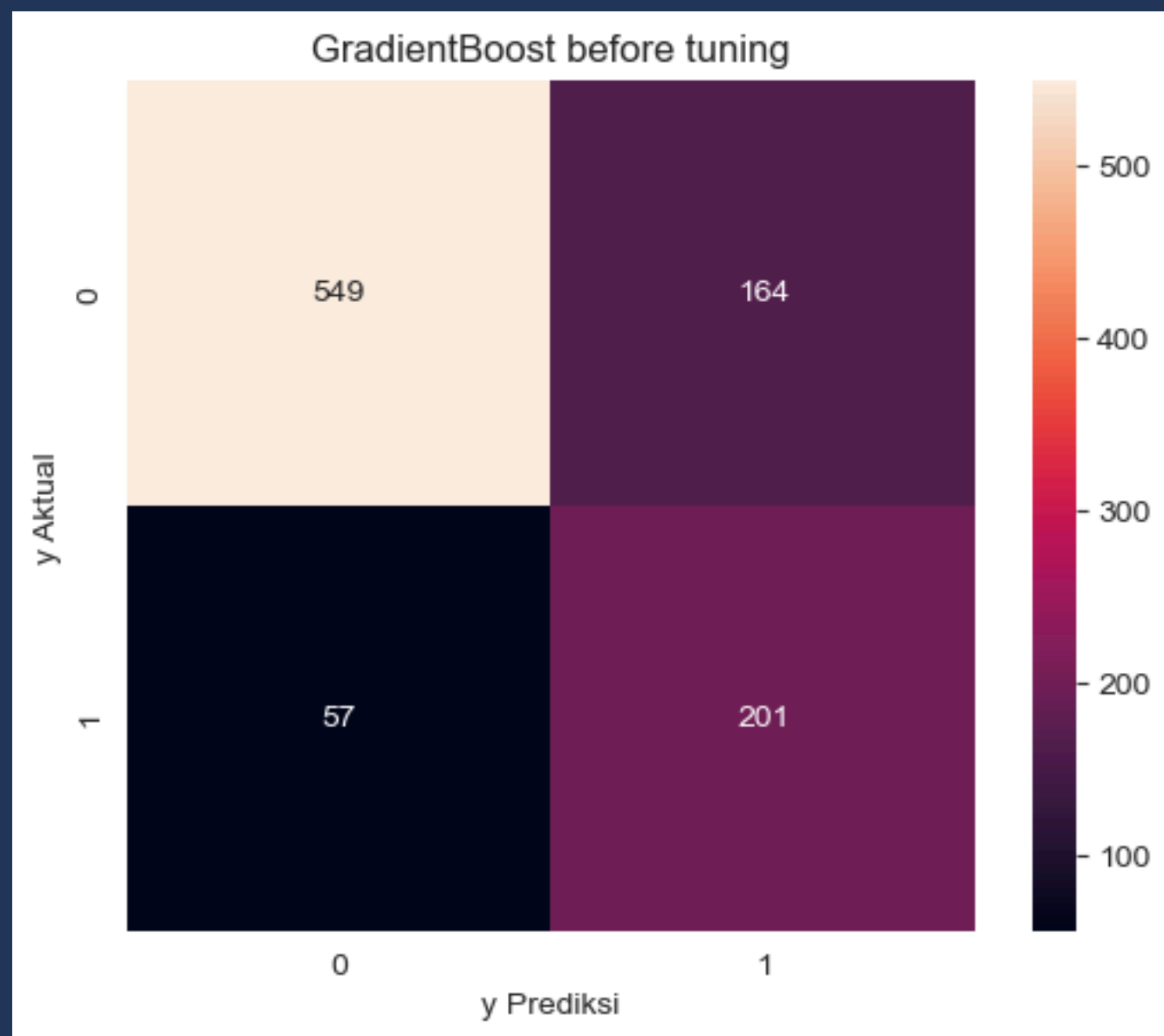
BEST MODEL

**Gradient Boosting setelah di tuning
merupakan best model**

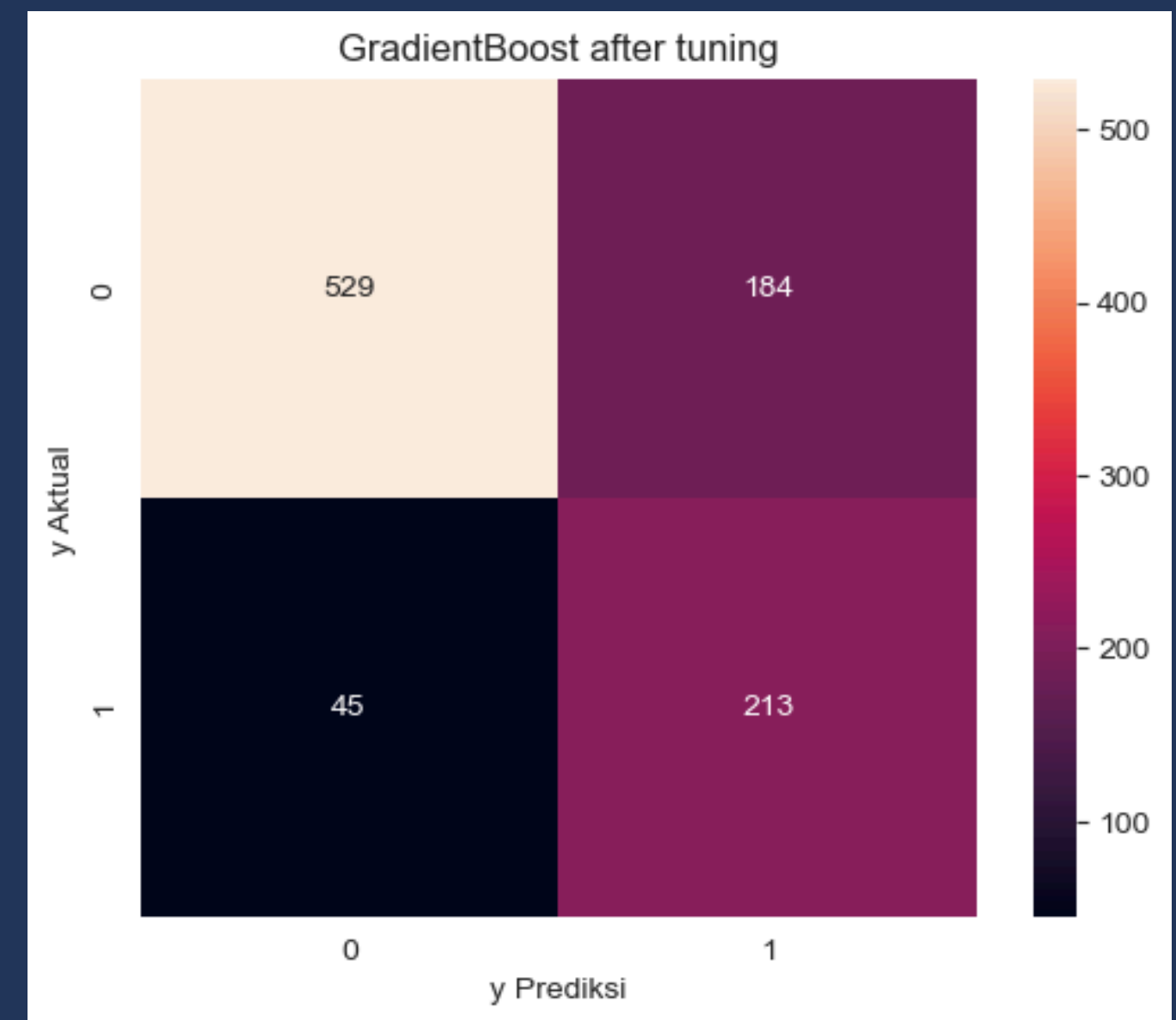
- **Best_score: 0.7295286811244361**
- **Best_params: {'resampler': RandomOverSampler(random_state=0), 'model__subsample': np.float64(0.6), 'model__n_estimators': np.int64(93), 'model__max_features': np.int64(3), 'model__max_depth': np.int64(1), 'model__learning_rate': np.float64(0.34)}**
- **F2 Score : 0.745**
- **Before Tuning : 0.7194**
- **After Tuning : 0.7453**

BEST MODEL

Before



After



MACHINE LEARNING VS NO MACHINE LEARNING

	Predicted (0)	Predicted (1)
Actual (0)	0	713
Actual (1)	0	258

- Total biaya promosi (seluruh pelanggan):

$$971 \times \$10 = \$9.710$$

- Promosi yang tepat sasaran (untuk 258 pelanggan churn):

$$258 \times \$10 = \$2.580$$

- Biaya promosi yang sia-sia ke pelanggan loyal:

$$713 \times \$10 = \$7.130$$

	Predicted (0)	Predicted (1)
Actual (0)	529	184
Actual (1)	45	213

- False Positive (FP):

$$184 \times \$10 = \$1.840 \rightarrow \text{biaya promosi ke pelanggan loyal}$$

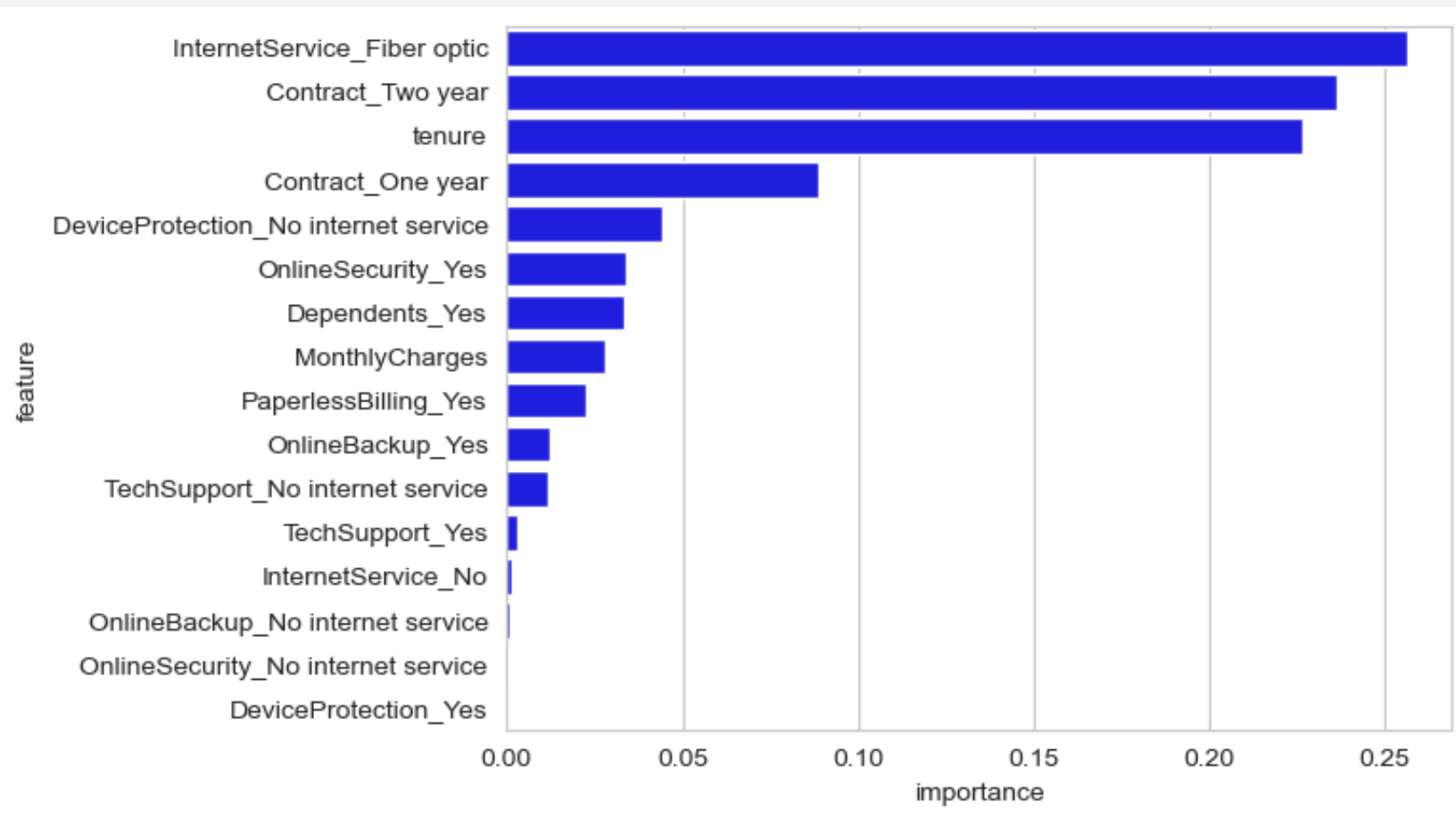
- False Negative (FN):

$$45 \times \$80 = \$3.600 \rightarrow \text{kehilangan customer karena tidak dipromosikan}$$

- Total kerugian:

$$\$1.840 + \$3.600 = \$5.440$$

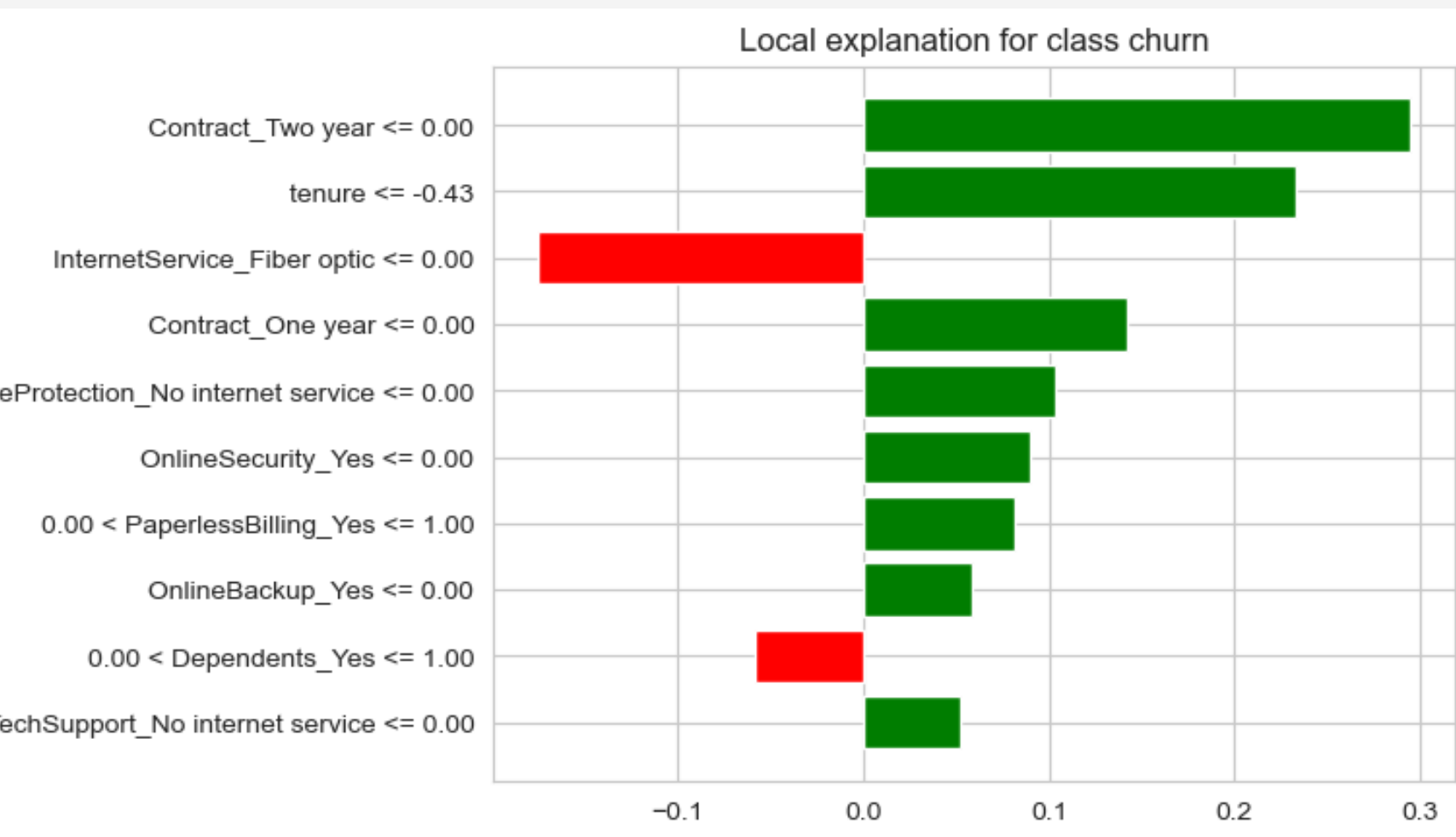
FEATURE IMPORTANCE



Berdasarkan feature importance dari model Gradient Boosting, fitur yang paling berpengaruh terhadap churn adalah:

- InternetSetvice
- Contract
- Tenure

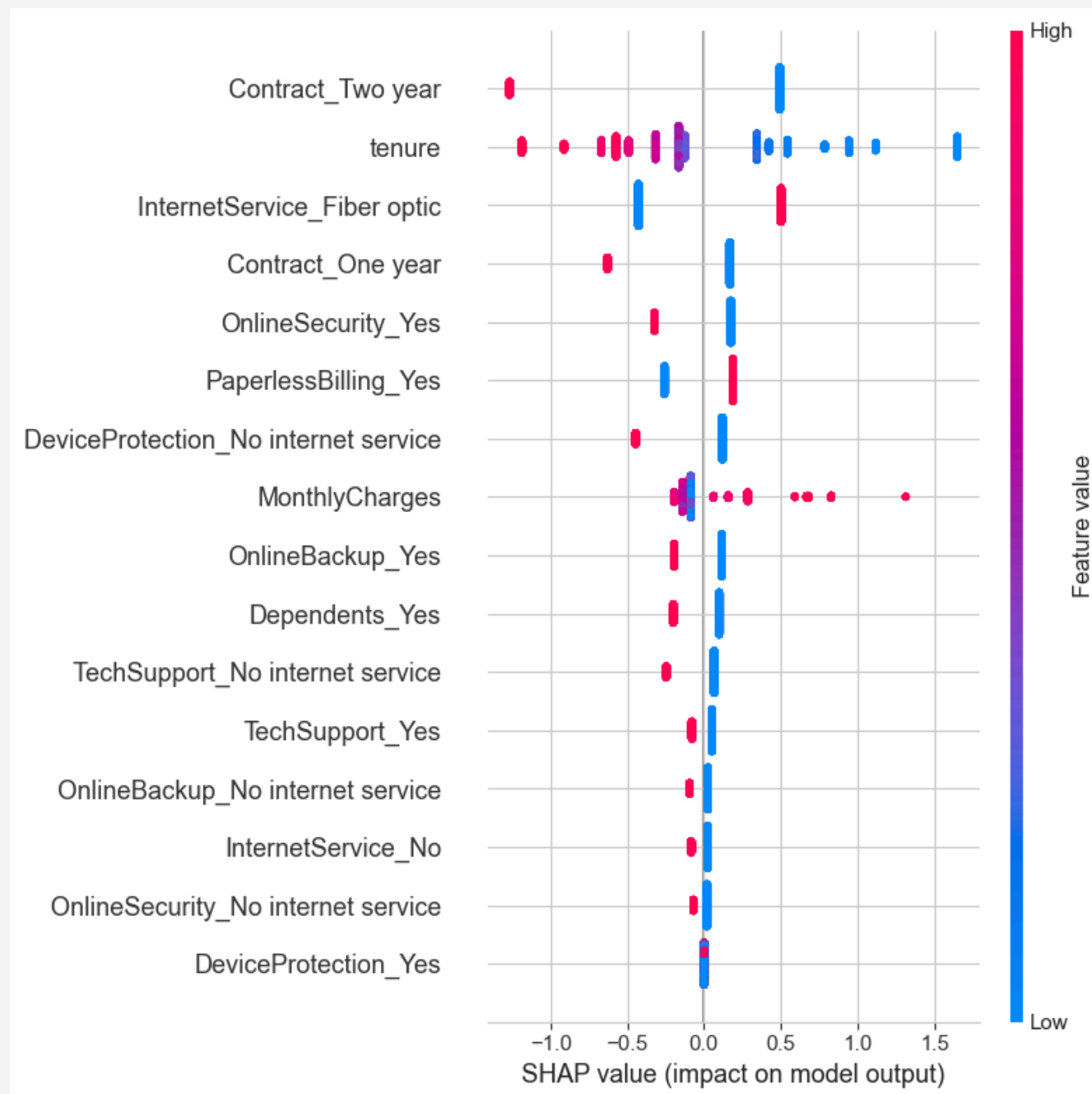
FEATURE IMPORTANCE



Berdasarkan feature importance dari model Gradient Boosting, fitur yang paling berpengaruh terhadap churn adalah:

- Contract
- Tenure
- InternetService

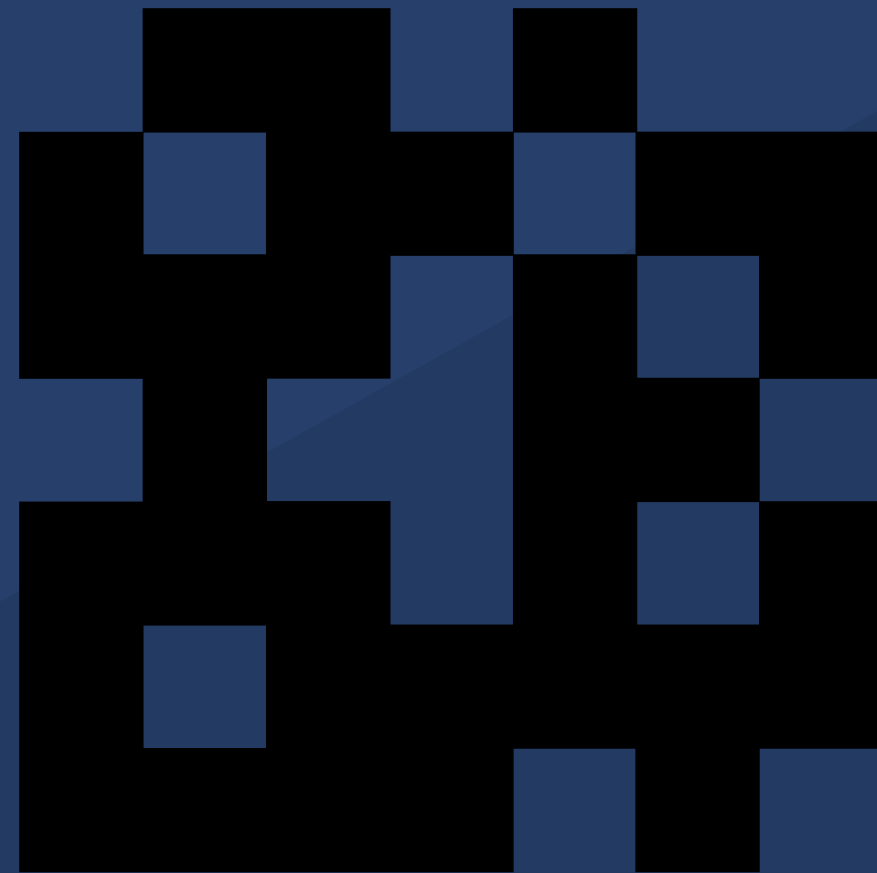
FEATURE IMPORTANCE



Berdasarkan feature importance dari model Gradient Boosting, fitur yang paling berpengaruh terhadap churn adalah:

- Contract
- Tenure
- InternetService

SIMULASI MACHINE LEARNING



KESIMPULAN DAN SARAN



Tingkat Churn Pelanggan masih tergolong tinggi yakni sebesar 27% yang merupakan sinyal manajemen untuk meningkatkan retensi pelanggan

Pengaruh Feature terhadap Churn dimana feature yang paling berpengaruh adalah tenure, Contract, dan MontlyChages

**Model Gradient Boosting
F2 Score (Train Set): 0.7194
F2 Score (Test Set): 0.7453**

Machine Learning berhasil menurunkan kerugian sampai 45% dari total kerugian tanpa Machine Learning





THANK YOU

