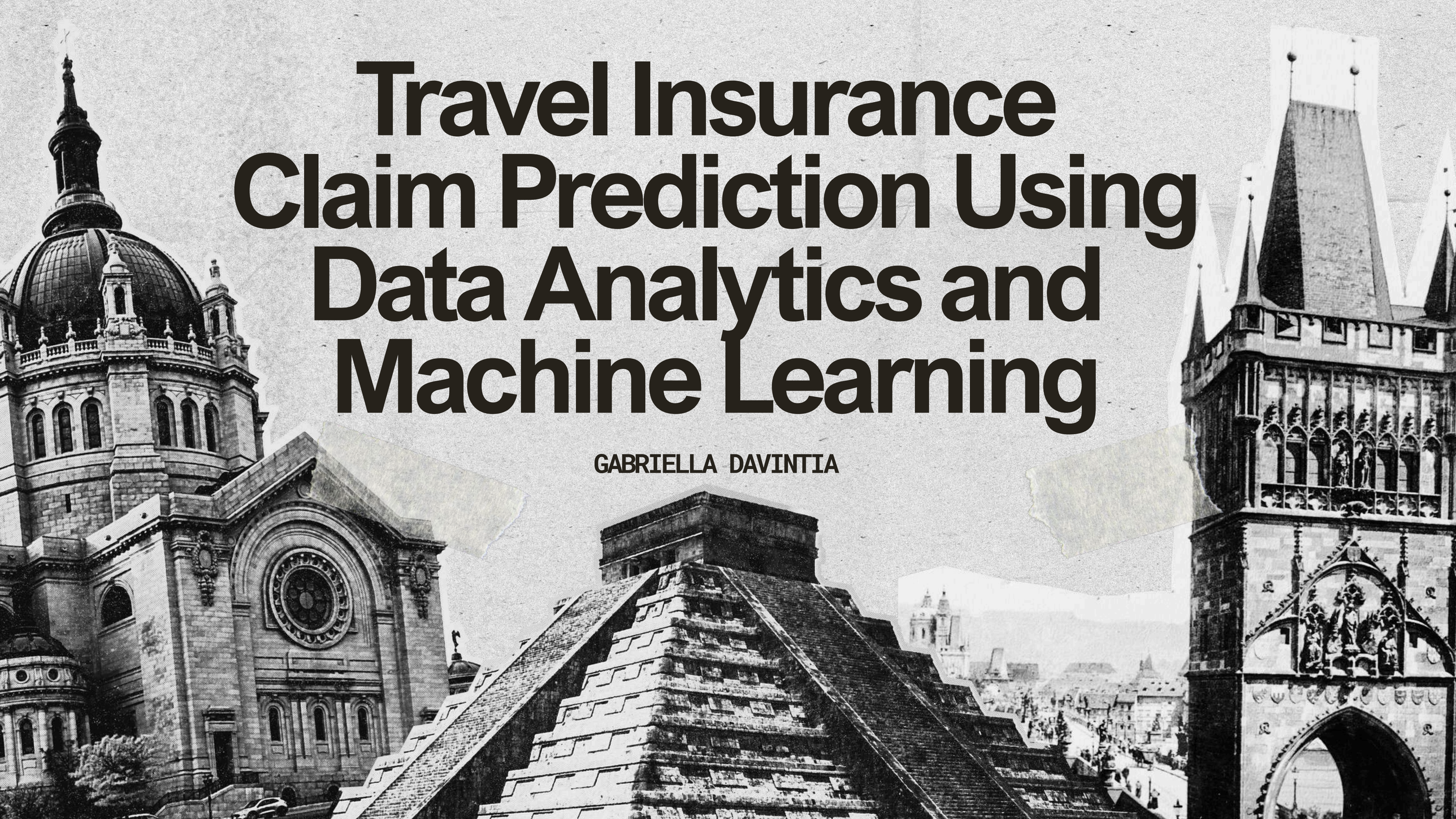
Travel Insurance Claim Prediction Using Data Analytics and Machine Learning

GABRIELLA DAVINTIA



EXECUTIVE SUMMARY

Perusahaan asuransi perjalanan menghadapi tantangan dalam memprediksi polis yang berpotensi mengajukan klaim.

Kesalahan prediksi dapat menyebabkan:

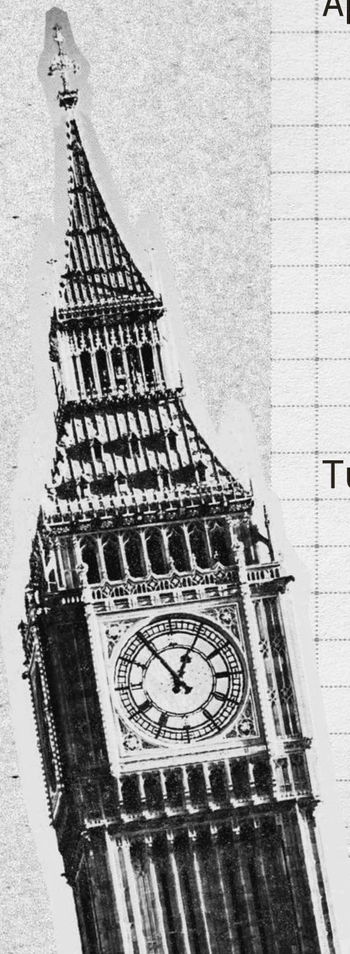
- Risiko klaim yang tidak terantisipasi
- Inefisiensi biaya dan pengelolaan polis

Proyek ini bertujuan membangun model machine learning untuk membantu identifikasi risiko klaim lebih dini

Claim
0: No
1: Yes

Table of Content

Business Problem Understanding



Context

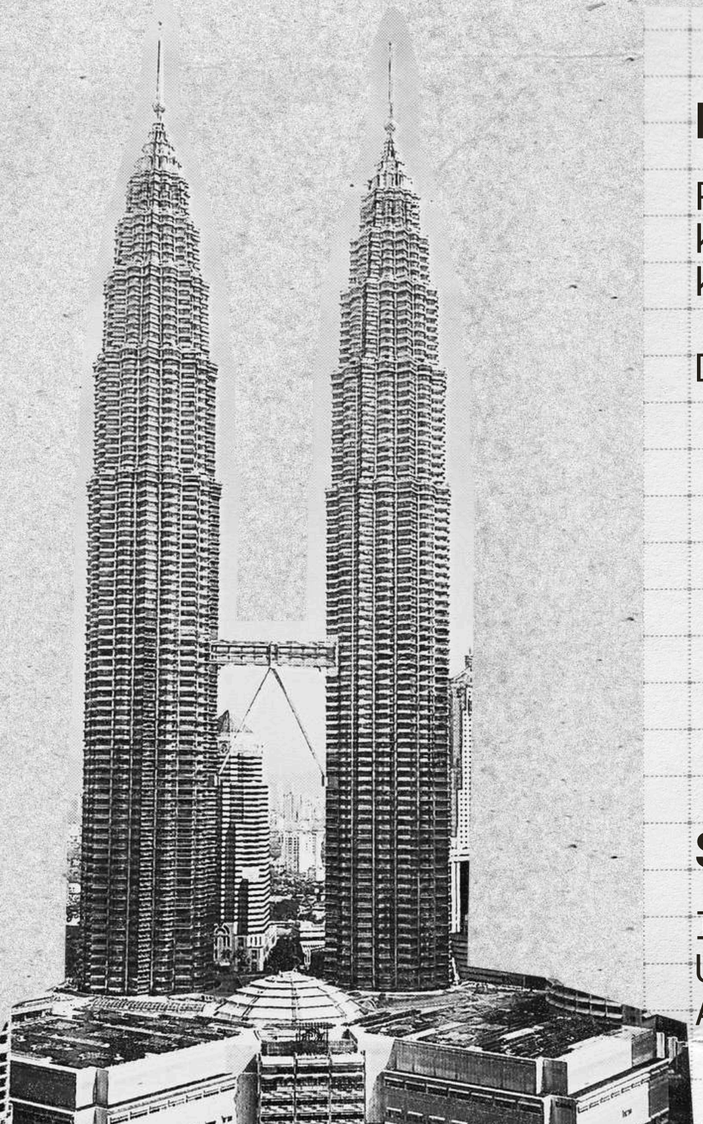
Apa Itu Asuransi Perjalanan?

- Memberikan perlindungan selama melakukan perjalanan, baik domestik maupun internasional.
- Beberapa negara mewajibkan turis memiliki asuransi perjalanan (contoh: negara-negara di Eropa dan Amerika).
- Besarnya premi dipengaruhi oleh:
  - Jenis pertanggungan
  - Lama perjalanan
  - Tujuan perjalanan

Tujuan Analisis

- Perusahaan asuransi ingin memprediksi pemegang polis yang berpotensi mengajukan klaim. Mengurangi Type 2 Error (False Negative)
- Data yang digunakan merupakan data historis pemegang polis.
- Variabel yang dianalisis mencakup:
  - Destinasi perjalanan
  - Produk asuransi
  - Informasi demografis
  - Karakteristik polis dan perjalanan

Business Problem Understanding



Problem Statement

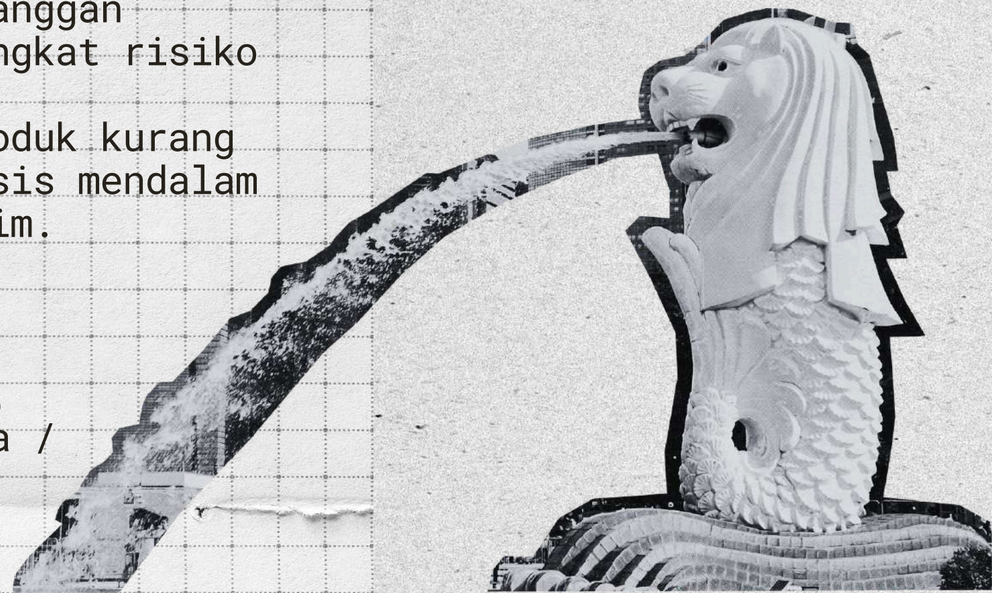
Perusahaan belum memiliki pemahaman jelas mengenai karakteristik dan perilaku pelanggan yang mengajukan klaim asuransi perjalanan.

Dampak yang Ditimbulkan:

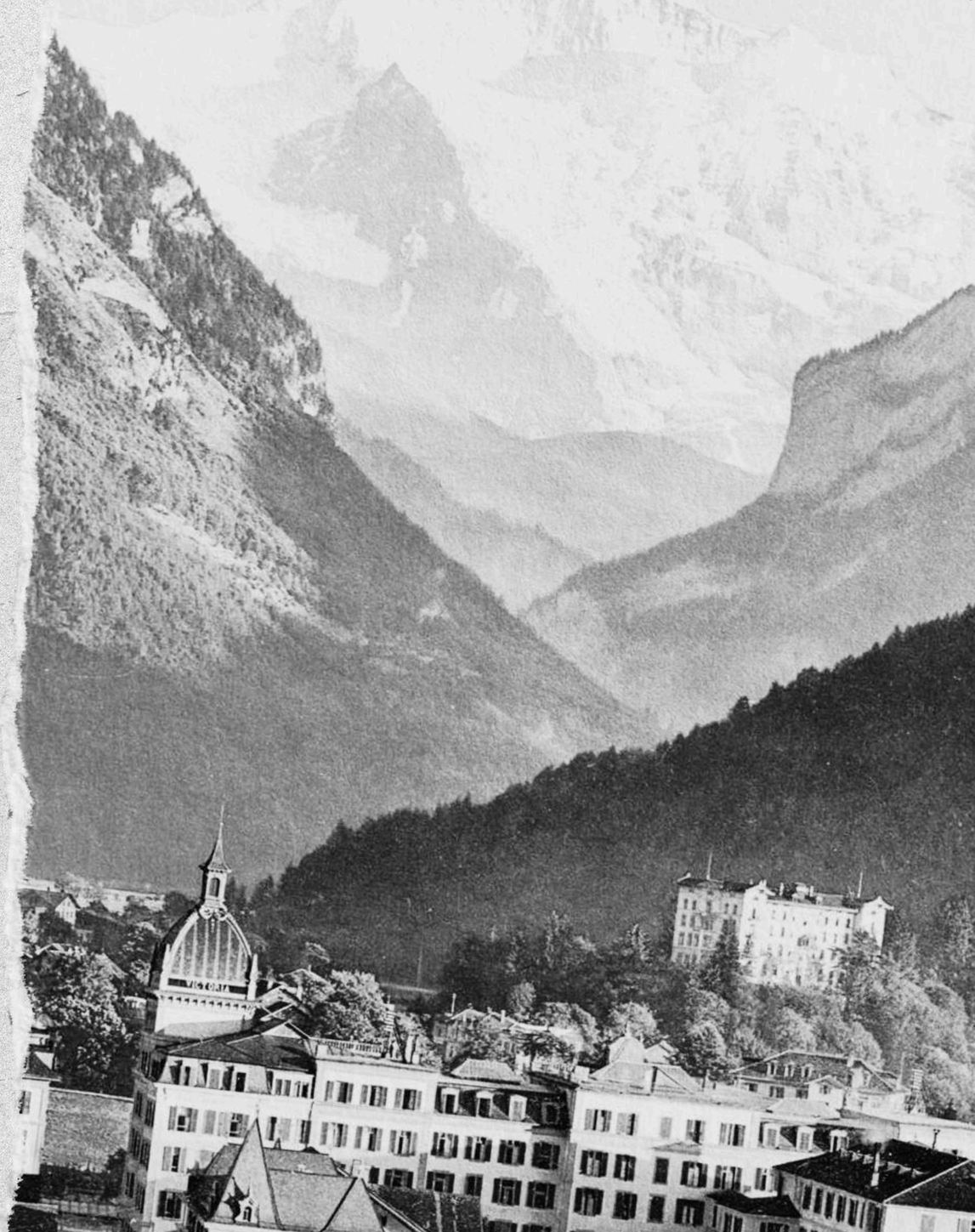
- Kesulitan mengidentifikasi faktor utama yang mempengaruhi kemungkinan klaim → penentuan premi & mitigasi risiko belum optimal.
- Belum dapat melakukan segmentasi pelanggan berdasarkan profil, perilaku, dan tingkat risiko klaim.
- Strategi pemasaran & pengembangan produk kurang tepat sasaran karena belum ada analisis mendalam terkait pola pembelian polis dan klaim.

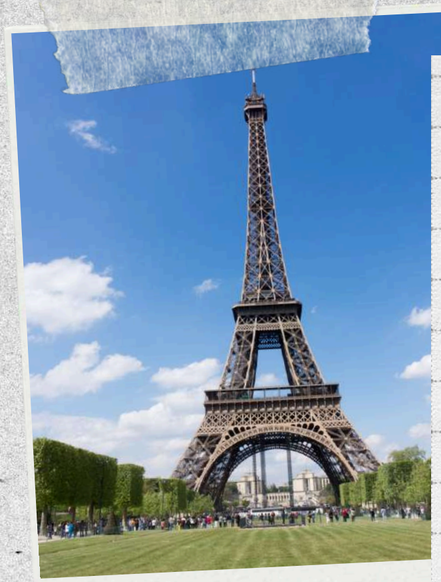
Stakeholder

Investor, Manajemen Perusahaan Asuransi, Underwriting & Risk Management, dan Data / Analytics



Faktor apa saja yang paling memengaruhi kemungkinan seorang nasabah untuk mengajukan klaim asuransi perjalanan?





EIFELL TOWER
(PARIS, FRANCE)

Data Dictionary

44328 rows × 11 columns

Kategorik : 6

Numerik : 4

Boolean : 1

Info agensi

1. Agency: Name of agency.
2. Agency Type: Type of travel insurance agencies.
3. Distribution Channel: Channel of travel insurance agencies.

Info produk

4. Product Name: Name of the travel insurance products.

Info nasabah

5. Gender: Gender of insured.

Info perjalanan

6. Duration: Duration of travel.
7. Destination: Destination of travel.

Info penjualan

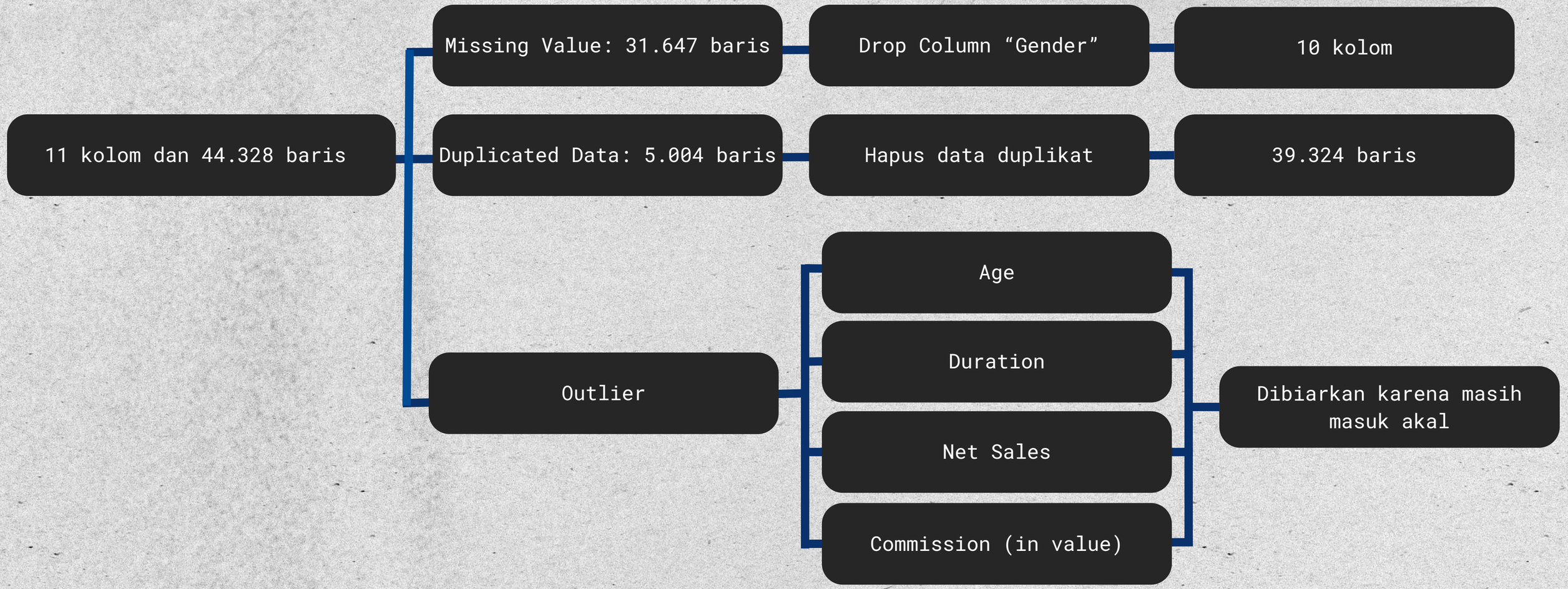
8. Net Sales: Amount of sales of travel insurance policies.
9. Commission (in value): Commission received for travel insurance agency.
10. Age: Age of insured.

Info Klaim

11. Claim: Claim status.

LEANING TOWER OF PISA
(TUSCANY, ITALY)

Data Cleaning



Data Cleaning

	dataFeatures	dataType	null	nullPct	unique	uniqueSample
0	Agency	object	0	0.00	16	[CSR, KML]
1	Agency Type	object	0	0.00	2	[Airlines, Travel Agency]
2	Distribution Channel	object	0	0.00	2	[Online, Offline]
3	Product Name	object	0	0.00	26	[1 way Comprehensive Plan, Annual Travel Prote...
4	Gender	object	31647	71.39	2	[F, M]
5	Duration	int64	0	0.00	437	[44, 384]
6	Destination	object	0	0.00	138	[BOLIVIA, LATVIA]
7	Net Sales	float64	0	0.00	1006	[47.5, 5.27]
8	Commision (in value)	float64	0	0.00	915	[20.64, 12.07]
9	Age	int64	0	0.00	89	[86, 33]
10	Claim	object	0	0.00	2	[Yes, No]

	dataFeatures	dataType	null	unique	uniqueSample
0	Agency	object	0	16	[CCR, ART]
1	Agency Type	object	0	2	[Airlines, Travel Agency]
2	Distribution Channel	object	0	2	[Online, Offline]
3	Product Name	object	0	26	[Single Trip Travel Protect Gold, Ticket Prote...
4	Duration	int64	0	437	[257, 268]
5	Destination	object	0	138	[TRINIDAD AND TOBAGO, ARGENTINA]
6	Net Sales	float64	0	1006	[4.84, -29.5]
7	Commision (in value)	float64	0	915	[82.6, 23.4]
8	Age	int64	0	89	[23, 82]
9	Claim	object	0	2	[Yes, No]

[177]
✓ 0s

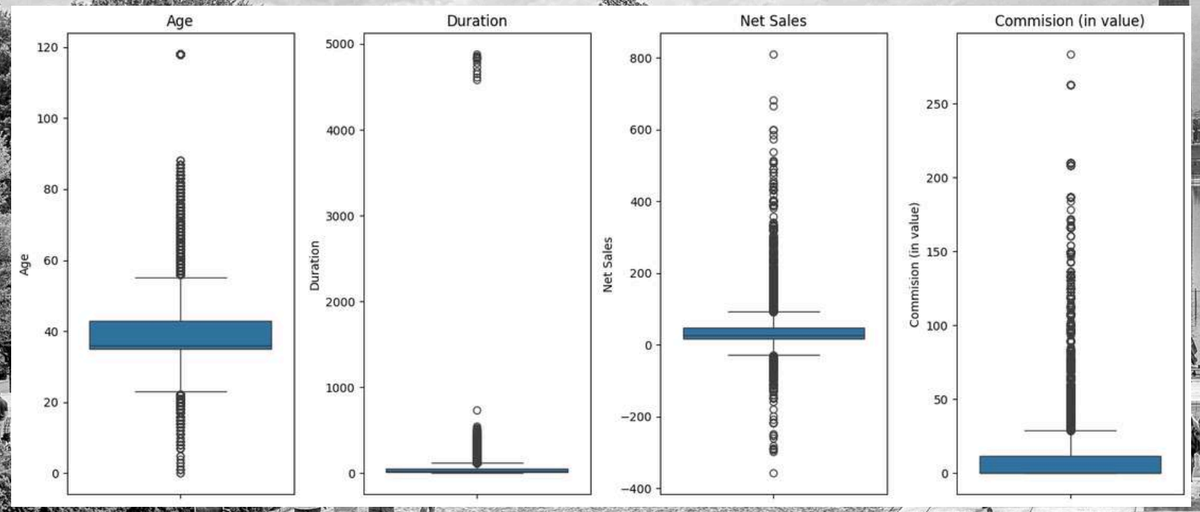
```
print("Jumlah duplikat sebelum dihapus:", df_model.duplicated().sum())

df_clean = df_clean.drop_duplicates(keep='first')

print("Jumlah duplikat sesudah dihapus:", df_clean.duplicated().sum())
print("Total baris setelah dihapus:", df_clean.shape[0])
```

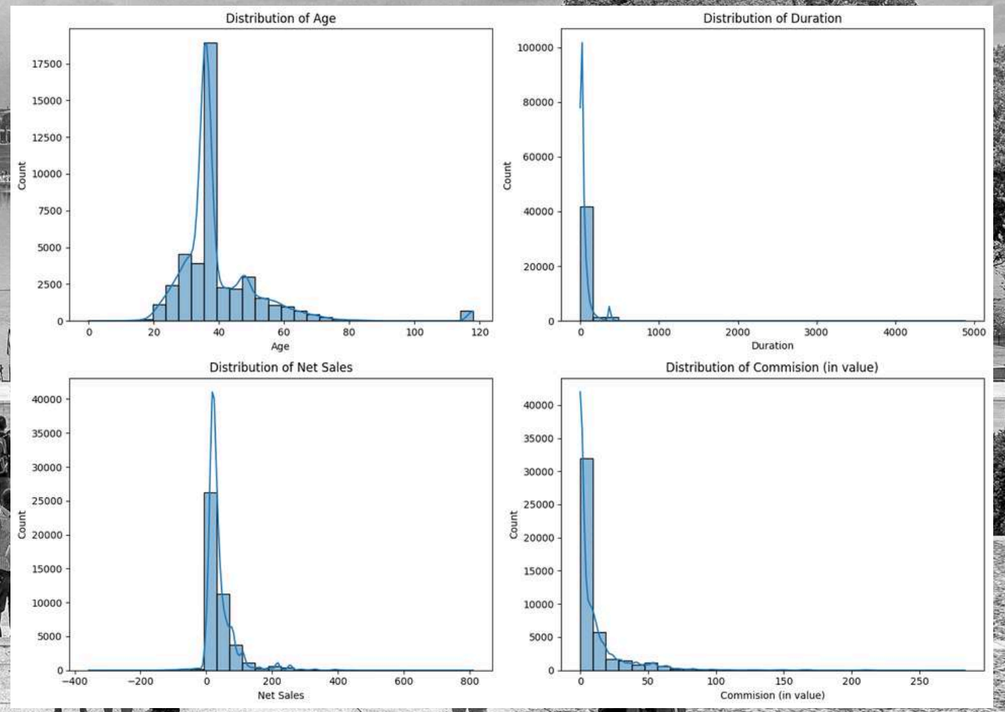
```
Jumlah duplikat sebelum dihapus: 5004
Jumlah duplikat sesudah dihapus: 0
Total baris setelah dihapus: 39324
```

Data Cleaning

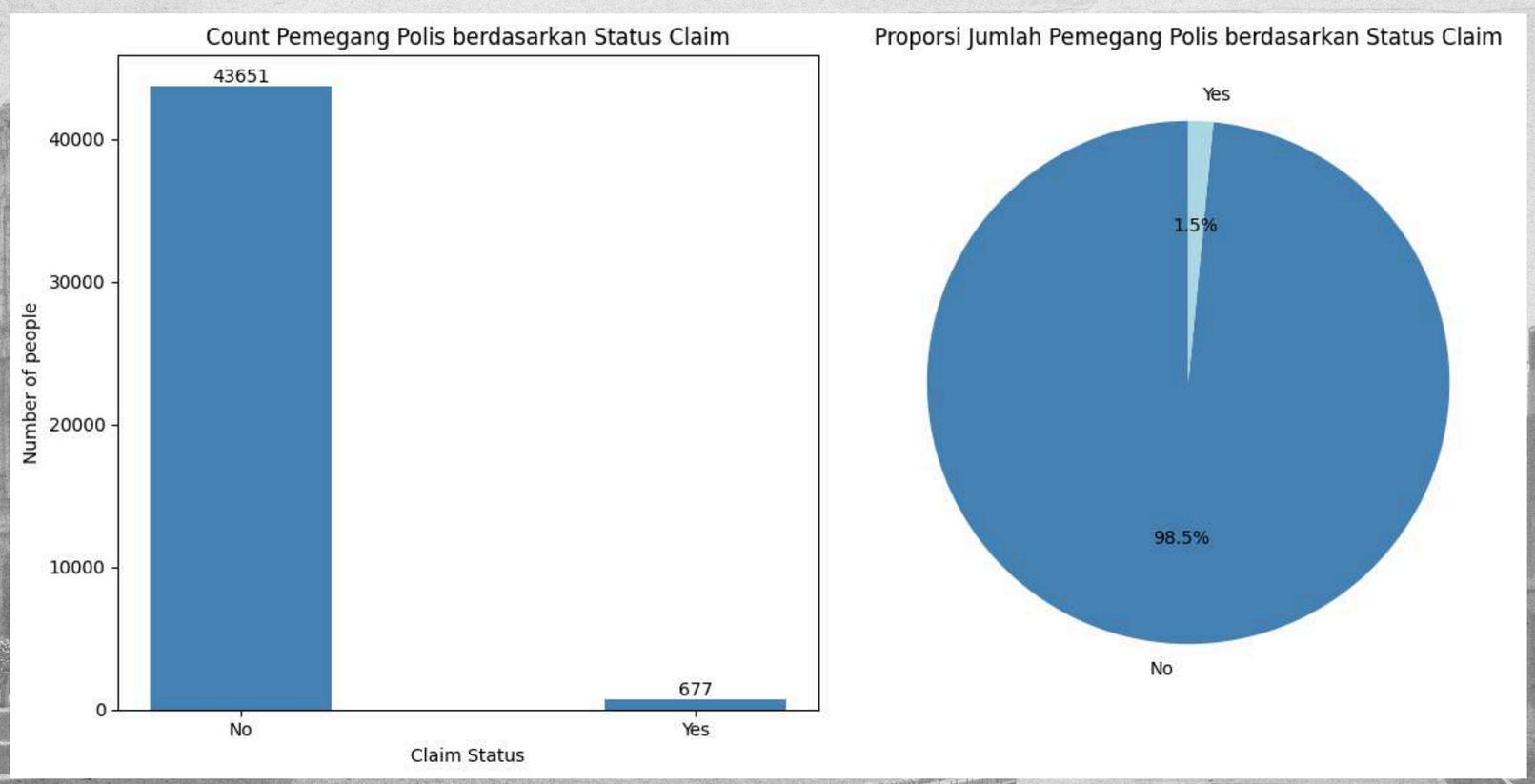
- Age: Mayoritas usia produktif, terdapat beberapa nilai ekstrem → outlier ringan
- Duration: Beberapa perjalanan berdurasi sangat panjang → outlier jelas
- Net Sales & Commission: Distribusi sangat right-skewed dengan nilai finansial tinggi → outlier finansial

- Outlier tidak bermasalah untuk model berbasis tree (LGBM, RF, XGBoost)
- Untuk model linear/jarak → dapat ditangani dengan log transform / robust scaling
- Outlier dipertahankan karena masih masuk akal secara bisnis (polis bernilai tinggi)



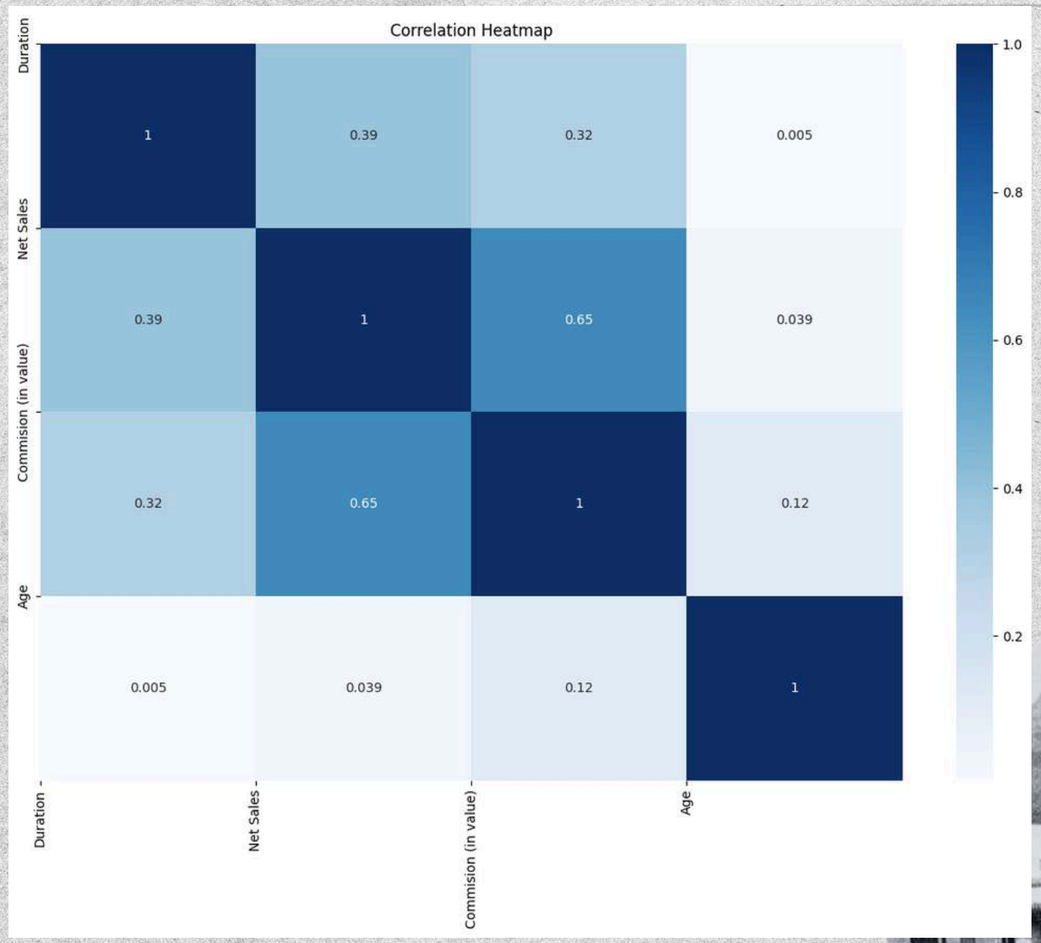


Data Balance/Imbalance



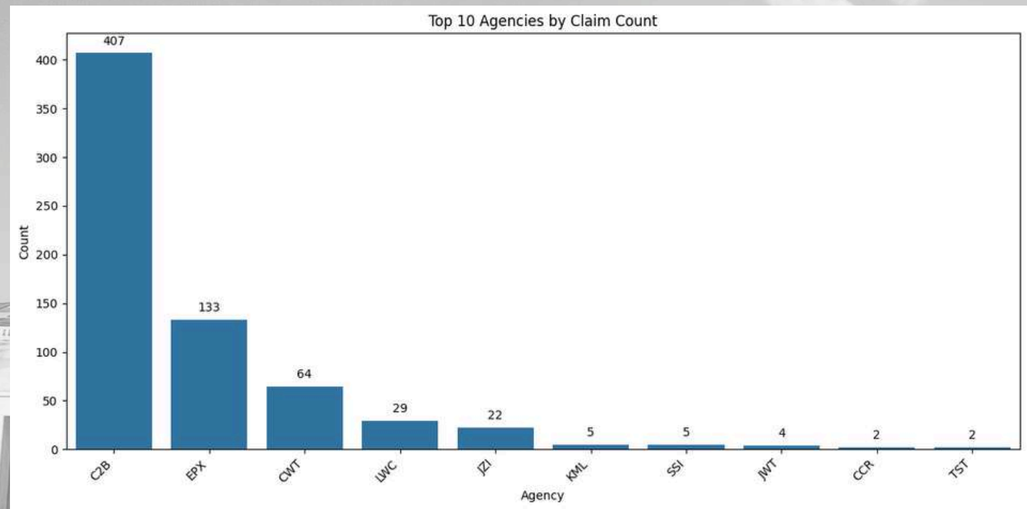
Pemegang polis dengan status claim merupakan kelas minoritas, sehingga data bersifat imbalanced.

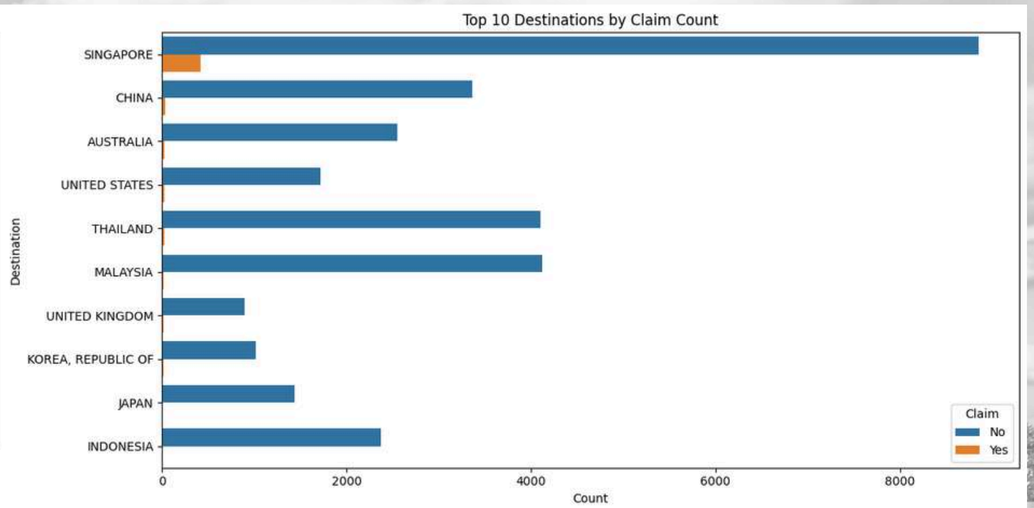
Data Correlation

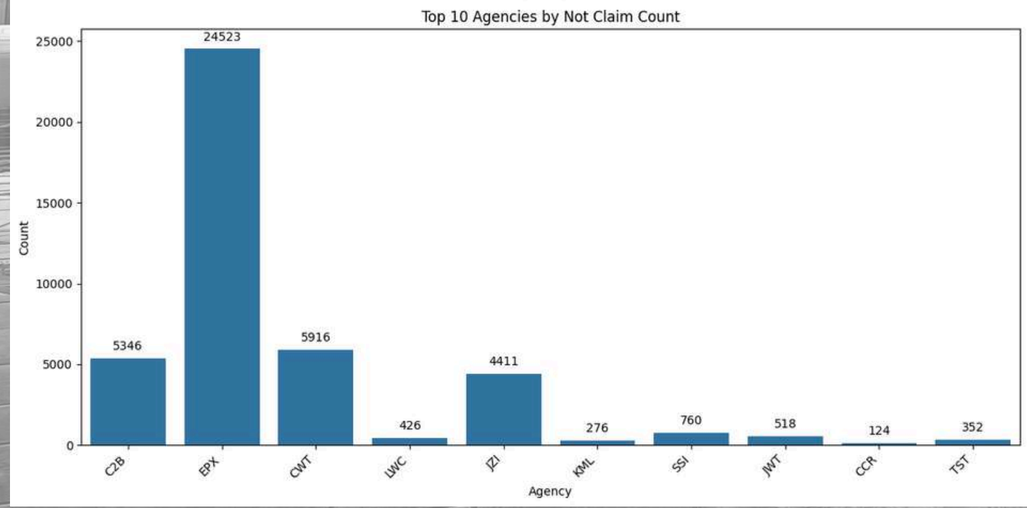


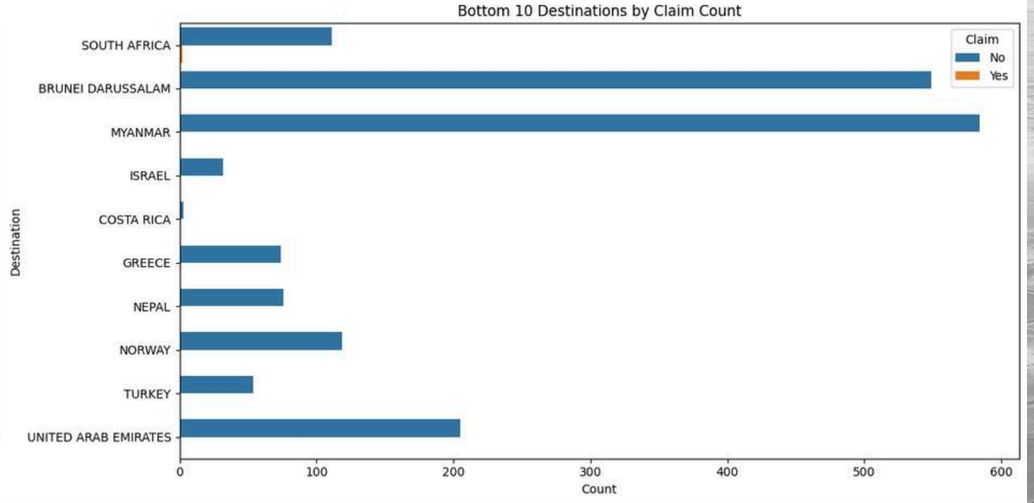
- Korelasi Kuat
  - Net Sales dan Commission (≈ 0.65)
- Korelasi Sedang
  - Duration dan Net Sales (≈ 0.39)
  - Duration dan Commission (≈ 0.32)
- Korelasi Lemah / Hampir Tidak Ada
  - Age dengan variabel lain (≈ 0.005 – 0.12)

Data Analysis

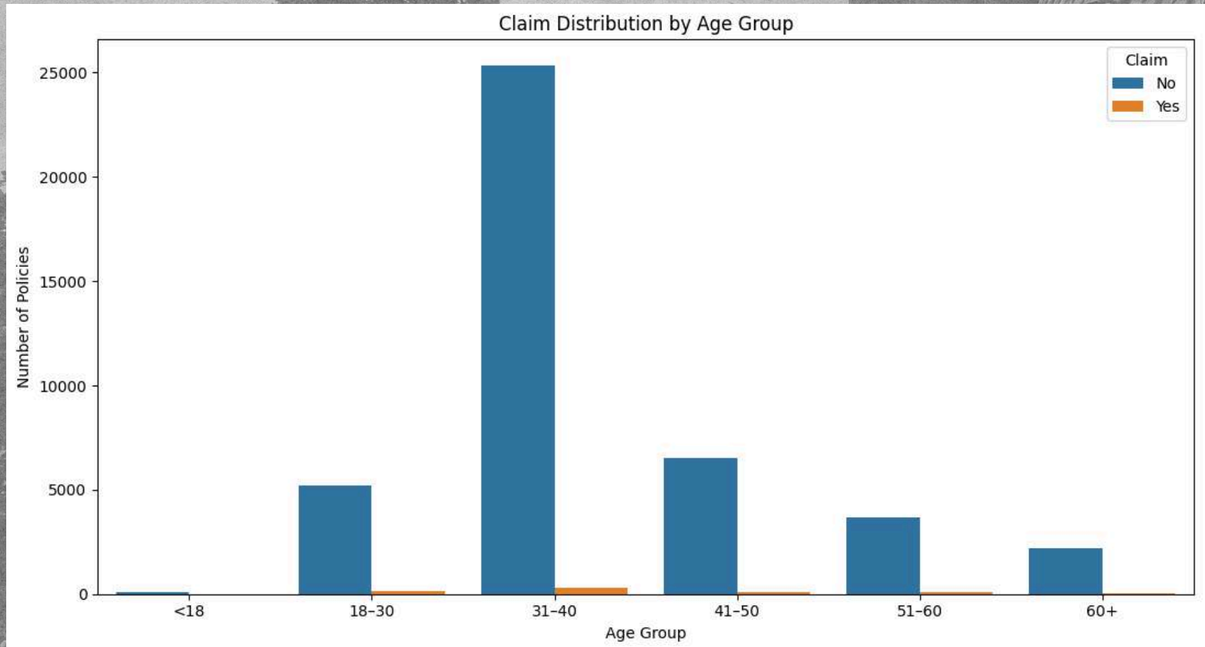


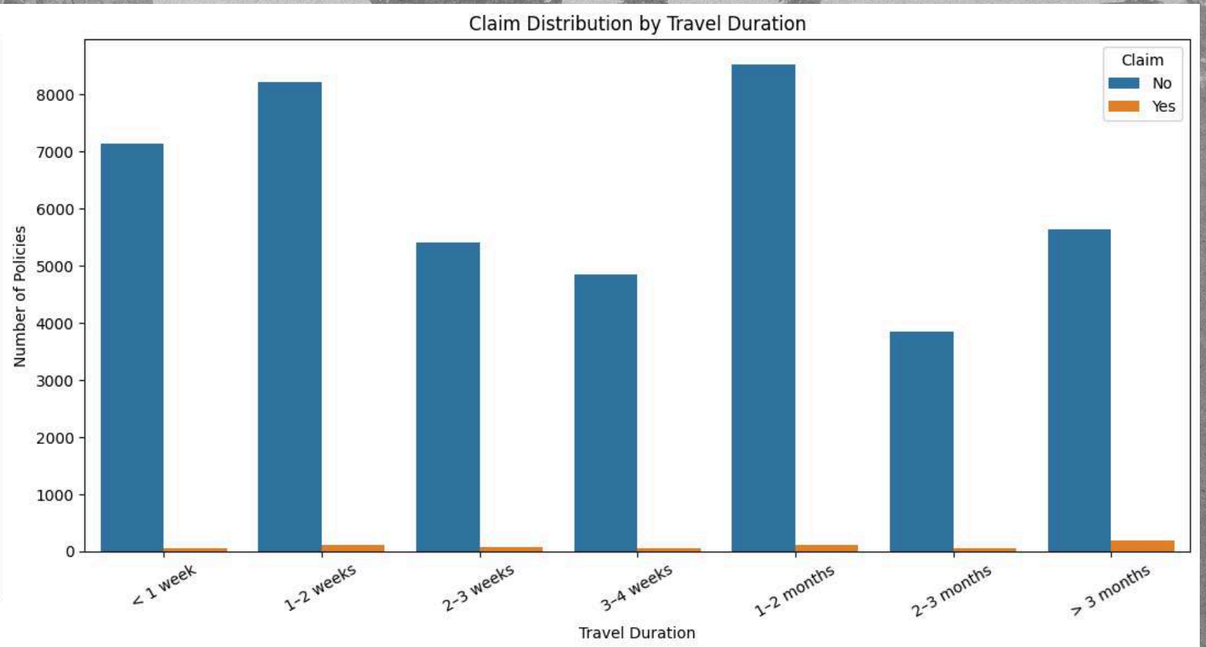






Data Analysis





Data Preparation

Encoding

Fitur Kategorikal

Agency, Product Name, Destination
→ Binary Encoding
(kardinalitas tinggi, non-ordinal, lebih efisien daripada One Hot)

Agency Type, Distribution Channel
→ One Hot Encoding
(kategori sedikit, non-ordinal)

Gender
→ Dikeluarkan dari modeling
(missing value sangat tinggi)

Fitur Numerik

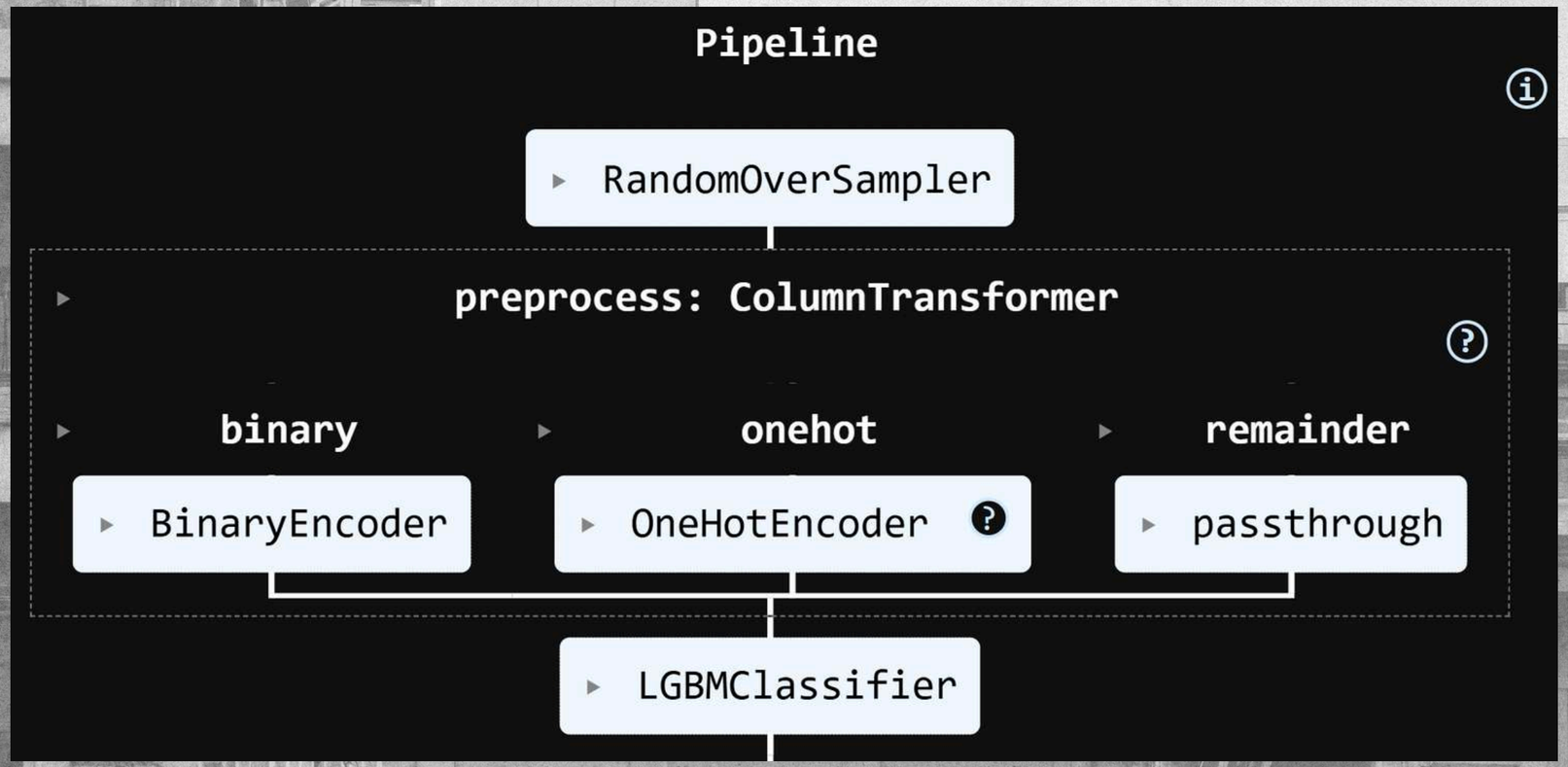
Age, Duration, Net Sales, Commission (in value)
→ Dipertahankan sebagai numerik

Duration
→ Dilakukan binning untuk menangkap pola durasi perjalanan

Claim
→ Dikodekan menjadi biner

0 = No Claim | 1 = Claim

Pipeline


Pipeline
RandomOverSampler
preprocess: ColumnTransformer
binary
onehot
remainder
BinaryEncoder
OneHotEncoder
passthrough
LGBMClassifier

Modeling

```
logreg = LogisticRegression()
knn = KNeighborsClassifier()
dt = DecisionTreeClassifier()
rf = RandomForestClassifier()
xgb = XGBClassifier()
lgbm = lgb.LGBMClassifier()
```

Model Benchmarking

model	mean_roc_auc	std
LightGBM	0.810717	0.015823
Logistic Regression	0.803261	0.011938
XGBoost	0.782092	0.015123
Random Forest	0.701583	0.013823
KNN	0.605010	0.031426
Decision Tree	0.539002	0.019062

K-Fold

model	roc_auc score
Logistic Regression	0.815622
LightGBM	0.812543
XGBoost	0.774712
Random Forest	0.676821
KNN	0.592728
Decision Tree	0.545614

Test Data

Evaluation Metrics

	Train Accuracy	Test Accuracy	Train ROC AUC	Test ROC AUC	Train F1 Score	Test F1 Score	Train Recall	Test Recall	Train Precision	Test Precision
0	0.985837	0.983930	0.979834	0.810986	0.563480	0.495950	0.071869	0.000000	1.000000	0.000000
1	0.986088	0.984494	0.979675	0.775958	0.577625	0.496093	0.088296	0.000000	1.000000	0.000000
2	0.985932	0.984772	0.982996	0.805105	0.570457	0.496164	0.079918	0.000000	1.000000	0.000000
3	0.986057	0.984490	0.981285	0.842937	0.577464	0.496092	0.088115	0.000000	1.000000	0.000000
4	0.985994	0.984490	0.984962	0.827097	0.575565	0.513635	0.086066	0.018519	0.976744	0.333333
5	0.985994	0.983644	0.979164	0.842036	0.578712	0.495877	0.090164	0.000000	0.936170	0.000000
6	0.985932	0.984490	0.982894	0.790288	0.570457	0.496092	0.079918	0.000000	1.000000	0.000000
7	0.985994	0.984772	0.981818	0.812458	0.575565	0.496164	0.086066	0.000000	0.976744	0.000000
8	0.986026	0.984772	0.979678	0.765130	0.575722	0.496164	0.086066	0.000000	1.000000	0.000000
9	0.986026	0.984490	0.980620	0.820431	0.575722	0.496092	0.086066	0.000000	1.000000	0.000000
Average	0.985988	0.984434	0.981293	0.809242	0.574077	0.497832	0.084254	0.001852	0.988966	0.033333

WITHOUT OVERSAMPLING

	Train Accuracy	Test Accuracy	Train ROC AUC	Test ROC AUC	Train F1 Score	Test F1 Score	Train Recall	Test Recall	Train Precision	Test Precision
0	0.931271	0.888920	0.974071	0.799823	0.931126	0.536411	0.977250	0.545455	0.894953	0.075188
1	0.930078	0.889202	0.975129	0.741003	0.929979	0.528874	0.967640	0.472727	0.900027	0.066667
2	0.937826	0.879301	0.973598	0.789980	0.937700	0.527276	0.982850	0.537037	0.901658	0.067130
3	0.934644	0.866046	0.973921	0.820685	0.934486	0.513159	0.983772	0.481481	0.895758	0.054968
4	0.935981	0.875917	0.975201	0.836611	0.935868	0.530098	0.977886	0.592593	0.902266	0.071111
5	0.930985	0.882121	0.970976	0.828327	0.930832	0.538321	0.978013	0.629630	0.893933	0.078704
6	0.934644	0.872250	0.974199	0.775948	0.934484	0.517099	0.984122	0.481481	0.895506	0.057650
7	0.939115	0.860688	0.974622	0.767058	0.938926	0.513153	0.994782	0.518519	0.895124	0.056452
8	0.934469	0.871968	0.973783	0.732030	0.934329	0.511564	0.980718	0.425926	0.897685	0.051570
9	0.932274	0.879865	0.973502	0.793594	0.932123	0.529485	0.979350	0.555556	0.895077	0.069444
Average	0.934129	0.876628	0.973900	0.788506	0.933985	0.524544	0.980638	0.524040	0.897199	0.064888

WITH OVERSAMPLING

Hyperparameter Tuning

LightGBM

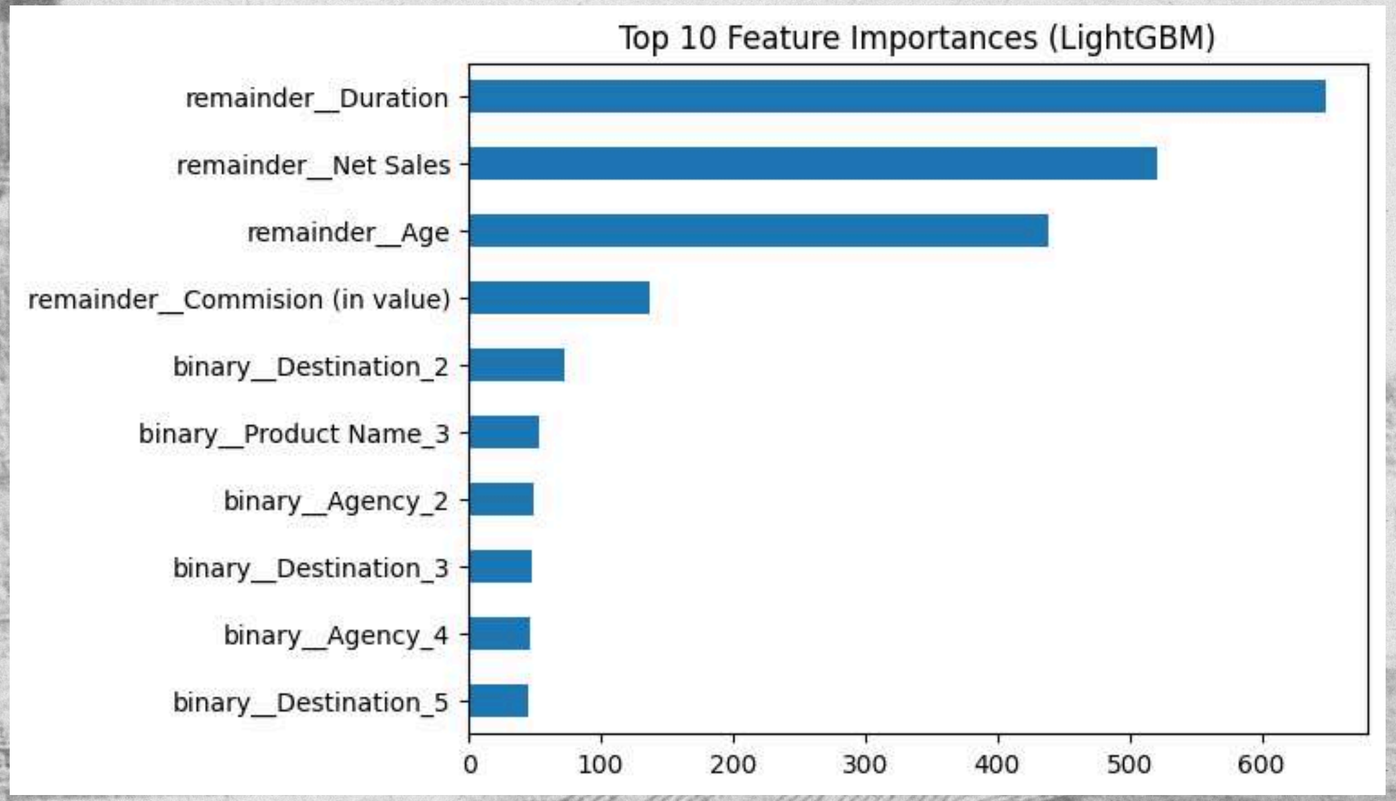
```
hyperparam_space=[{
    'model__max_bin': [255, 275, 300, 230],
    'model__num_leaves':[31, 51],
    'model__min_data_in_leaf': [20, 40],
    'model__num_iterations':[100, 150],
    'model__learning_rate': [0.1, 0.05],
    'model__random_state': [42]
}]
```

```
Default hyperparameter LGBM
max_bin = 255
num_leaves = 31
min_data_in_leaf = 20
num_iterations = 100
learning_rate = 0.1
```

```
Best ROC AUC: 0.8162229203065399
Best Parameters:
model__random_state: 42
model__num_leaves: 31
model__num_iterations: 75
model__min_data_in_leaf: 20
model__max_bin: 230
model__learning_rate: 0.05
```

```
ROC AUC Score Default LGBM : 0.798857201033355
ROC AUC Score Tuned LGBM   : 0.8211307516427204
```

Feature Importance



Conclusion

Classification Report Tuned LGBM :

	precision	recall	f1-score	support
0	0.99	0.83	0.91	8731
1	0.06	0.65	0.10	135
accuracy			0.83	8866
macro avg	0.53	0.74	0.50	8866
weighted avg	0.98	0.83	0.89	8866

Model Performance – Tuned LightGBM

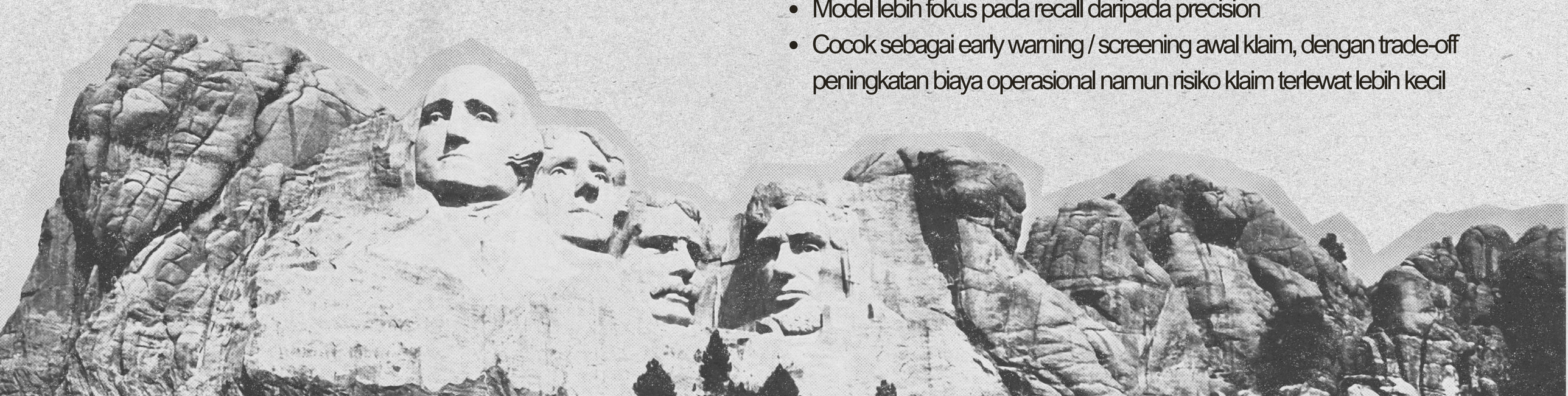
- Recall klaim (65%) → Model mampu mendeteksi mayoritas klaim aktual

Mendukung tujuan bisnis: meminimalkan False Negative & potensi kerugian finansial

- Precision rendah (6%) → Banyak False Positive

Berpotensi menambah biaya investigasi klaim yang tidak valid

- Model lebih fokus pada recall daripada precision
- Cocok sebagai early warning / screening awal klaim, dengan trade-off peningkatan biaya operasional namun risiko klaim terlewat lebih kecil



Recommendation

Bisnis

- Gunakan model sebagai early warning system klaim
- Mendukung:
  - Mitigasi risiko lebih dini
  - Alokasi cadangan klaim
  - Penentuan premi yang lebih tepat
  - Fokus untuk prioritisasi polis berisiko tinggi, bukan pengganti keputusan bisnis

Model

- Retrain berkala (6–12 bulan)
- Monitor recall & F2-score

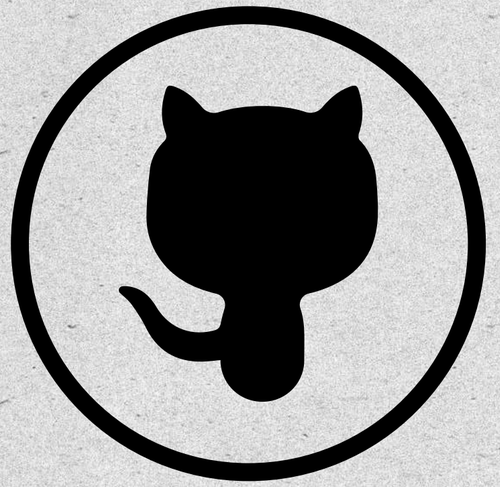
Data

- Tambahkan fitur penting:

Riwayat klaim, jenis perjalanan, risiko negara

- Kurangi missing value (Gender) agar fitur dapat dimanfaatkan

Thank You



Gabriella Davintia

Github