Travel Insurance Claim Prediction

- **Problem:** Travel insurance companies faced challenges in managing insurance portfolio risk, which requires long time in assessing policyholder risk resulting in claims.
- **Proposed Solution:** Develop a model to predict the likelihood of a customer filing a claim, enhancing risk assessment and minimizing losses.
- **Goals:** Identify high-risk customers and reduce claim processing times.

PROJECT LINK

Approach

- **Preprocessed the dataset** by select relevant features, handling missing values and outliers, and encoding categorical features.
- Conducted **exploratory data analysis** to identify relationships between variables.
- **Applied machine learning techniques**, including regression and boosting based model, to predict claim probabilities.
- **Evaluated model performance** with metrics such as recall and ROC-AUC to ensure effectiveness in identifying potential claims.

Tools

- **Programming Language:** Python
- **Tools:** Jupyter Notebook
- **Libraries:** Pandas, NumPy, Matplotlib, Scikit-Learn, XGBoost

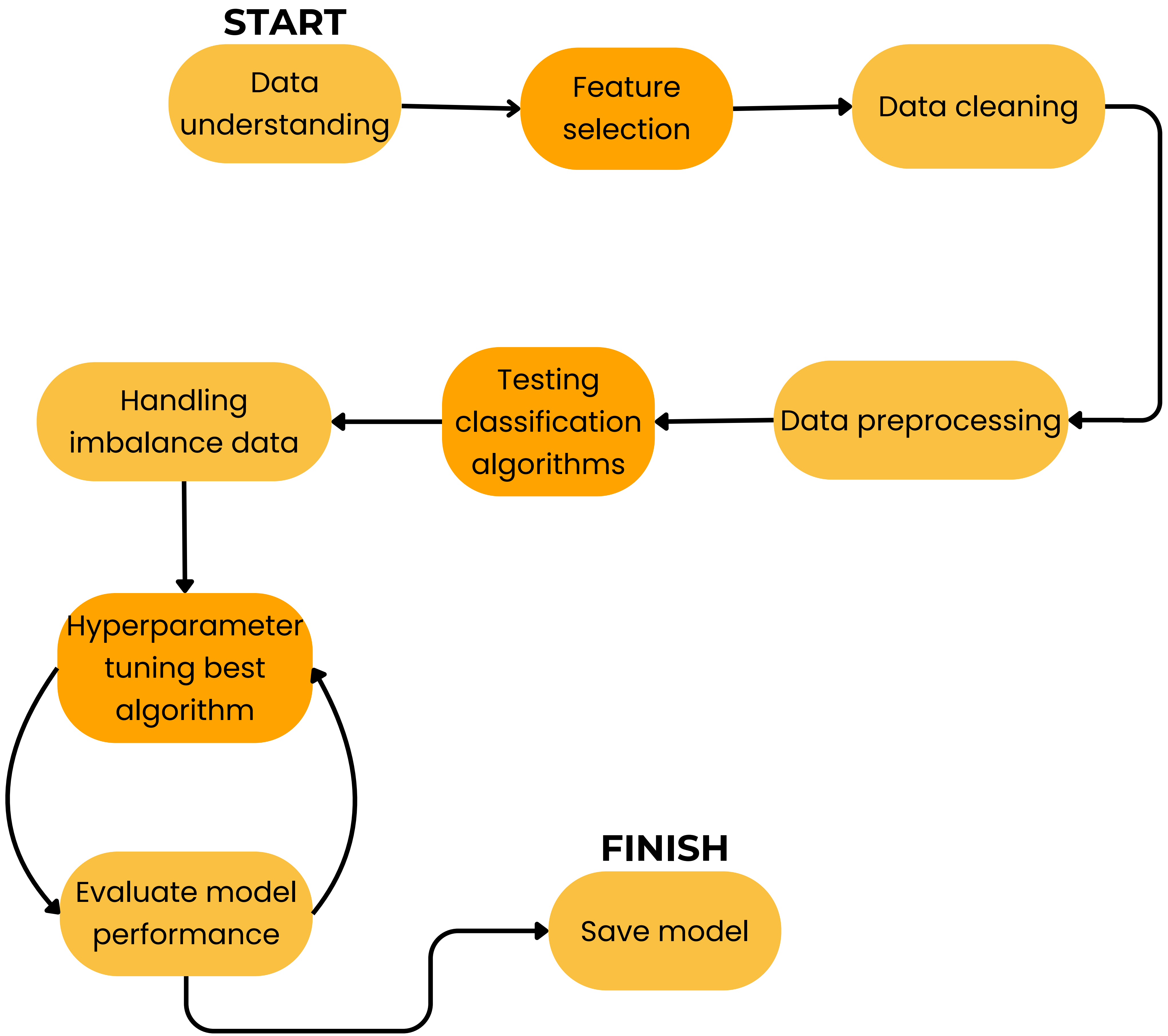
Datasets

- Variable information includes **insurance agency**, **product type**, and **revenue data**.
- Dataset has **44328 rows** and **11 variables**.
- Data was a third-party travel insurance servicing company that is based in Singapore.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44328 entries, 0 to 44327
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   agency                44328 non-null  object 
 1   agency_type           44328 non-null  object 
 2   distribution_channel  44328 non-null  object 
 3   product_name          44328 non-null  object 
 4   gender                12681 non-null  object 
 5   duration              44328 non-null  int64  
 6   destination           44328 non-null  object 
 7   net_sales_SGD         44328 non-null  float64
 8   commission_SGD        44328 non-null  float64
 9   age                   44328 non-null  int64  
 10  claim                 44328 non-null  object 
dtypes: float64(2), int64(2), object(7)
memory usage: 3.7+ MB
```

Building Model Process

1. **Data Understanding:** Analyze the data to understand its characteristics and unique value.
2. **Feature Selection:** Choose features that relevant with business objectives.
3. **Data Cleaning:** Clean the data using tailored techniques for business needs.
4. **Model Testing:** Test multiple classification algorithms with robust methods.
5. **Handling Imbalance Data:** Balancing data classes to ensure the model's predictions are unbiased and robust
6. **Model Tuning:** Tune the best algorithm's parameters and evaluate performance using recall.
7. **Deployment:** Train the final model on the entire dataset and prepare it for deployment.



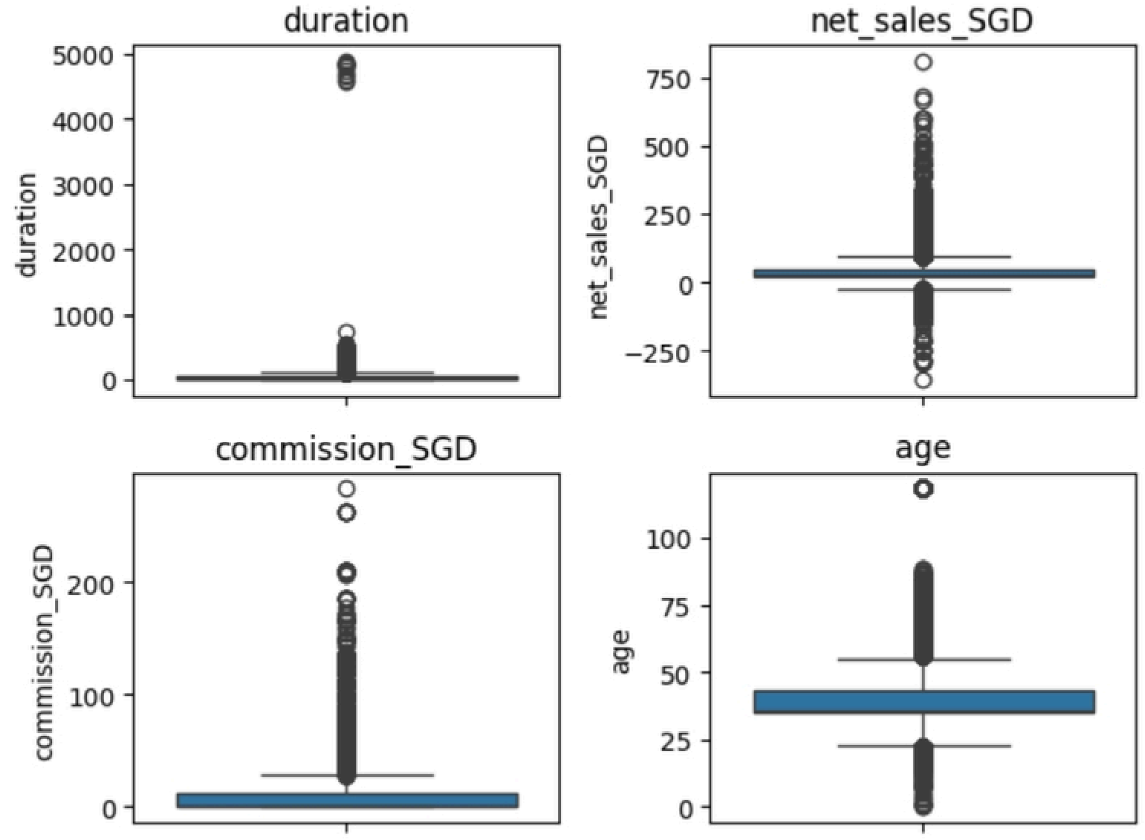
Data Cleaning

- Given the significant amount of missing data, **71% gender data missing, I will analyze how gender influences insurance claim tendencies** to determine the most effective handling strategy.
- The minimum **value for the Duration feature is zero**, which is unrealistic for an insurance policy. **Durations over 365 days will be removed**, as they exceed the product's maximum coverage of one year.
- **The Net Sales (SGD)** feature includes negative values, **requiring further detailed analysis**.
- **Large values in the Commission (SGD)** feature are outliers and **will be further investigated**.
- **Age values of 0 and above 75** are unrealistic based on Allianz Travel's coverage (1-79 years) and **will be excluded**.

Missing value

	0
agency	0.0
agency_type	0.0
distribution_channel	0.0
product_name	0.0
gender	71.0
duration	0.0
destination	0.0
net_sales_SGD	0.0
commission_SGD	0.0
age	0.0
claim	0.0

Outlier



Exploratory data analysis: Clean Net Sales (SGD)

- **Negative Values:** Identified negative Net Sales values only in unclaimed insurance policies, likely due to admin costs, commission, or discounts. Removed these rows (1% of the dataset).
- **Zero Values:** Found zero Net Sales in policies that were unclaimed or canceled. After analysis, determined no useful pattern and removed these rows (3% of the dataset).
- **Action Taken:** Cleaned Net Sales data by excluding rows with negative or zero values to enhance model accuracy.

```
df[(df['net_sales_SGD'] < 0) & (df['claim'] == 'Yes')]
```

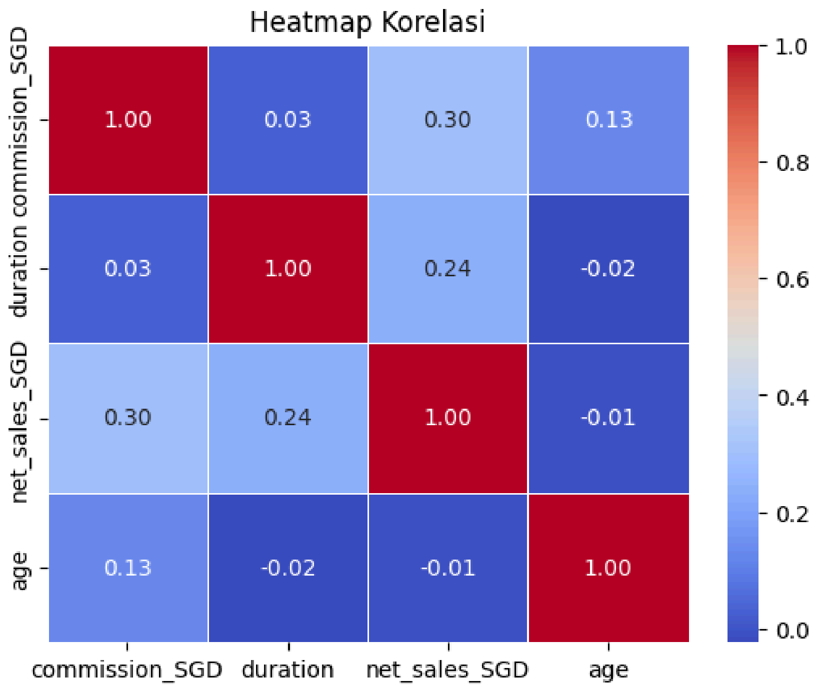
agency_type	product_name	gender	duration	destination	net_sales_SGD	commission_SGD	age	claim

```
df[(df['net_sales_SGD'] < 0) & (df['claim'] == 'No')]
```

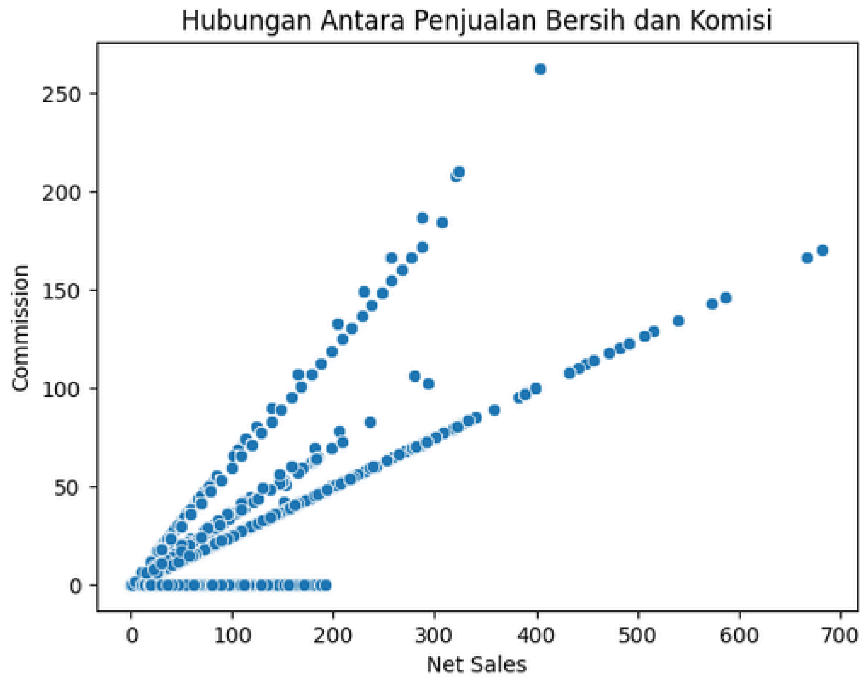
	agency_type	product_name	gender	duration	destination	net_sales_SGD	commission_SGD	age	claim
94	Airlines	Annual Silver Plan	M	365	SINGAPORE	-216.75	54.19	36	No
121	Travel Agency	Rental Vehicle Excess Insurance	NaN	77	JAPAN	-29.70	17.82	59	No
199	Travel Agency	Cancellation Plan	NaN	29	HONG KONG	-12.00	0.00	36	No
241	Travel Agency	Rental Vehicle Excess Insurance	NaN	57	AUSTRALIA	-59.40	35.64	28	No
597	Travel Agency	Rental Vehicle Excess Insurance	NaN	15	AUSTRALIA	-19.80	11.88	23	No

Exploratory data analysis: Clean Commission (SGD)

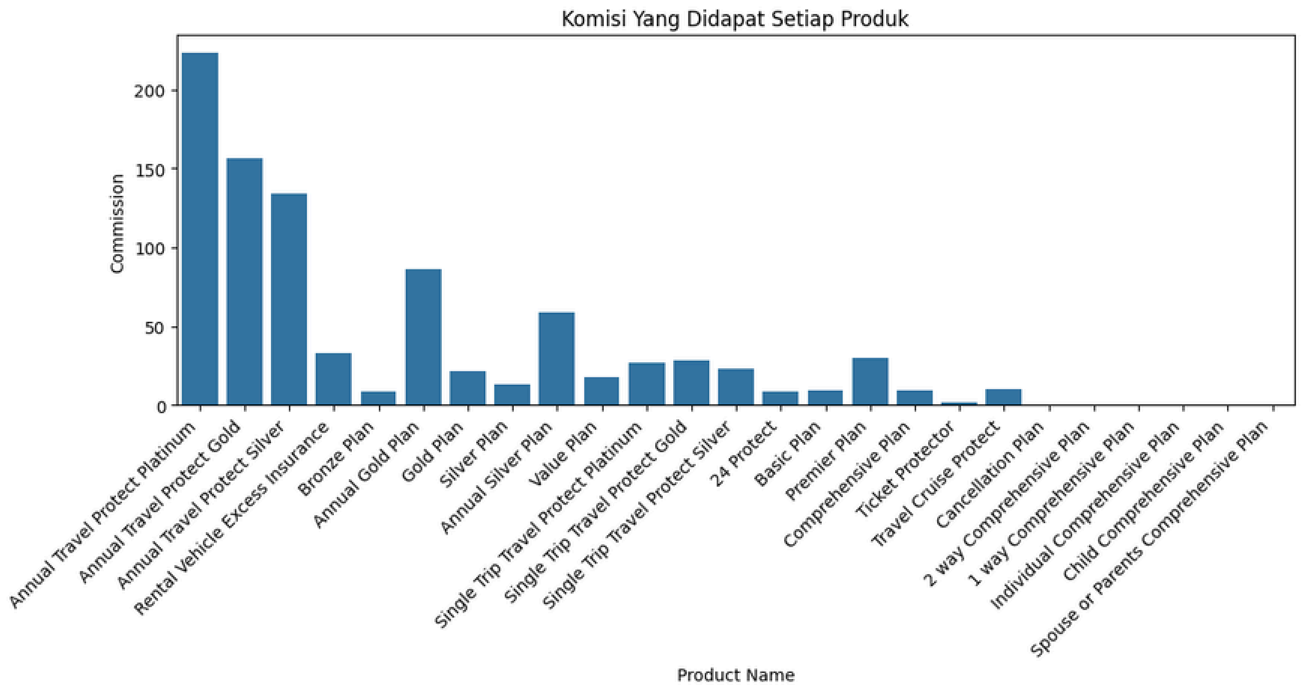
- **Correlation Analysis:** No strong correlations between numerical features; moderate positive correlation found between Net Sales and Commission.
- **Scatterplot Insights:** Higher Net Sales generally lead to higher commissions for agents, though some sales do not provide commissions depending on product type and agreements.
- **Barplot Conclusion:** Annual insurance products yield the highest commissions due to longer policy periods and higher associated risks.
- **Action Taken:** This will serve as an insight but will not be incorporated into the training model.



Correlation Analysis



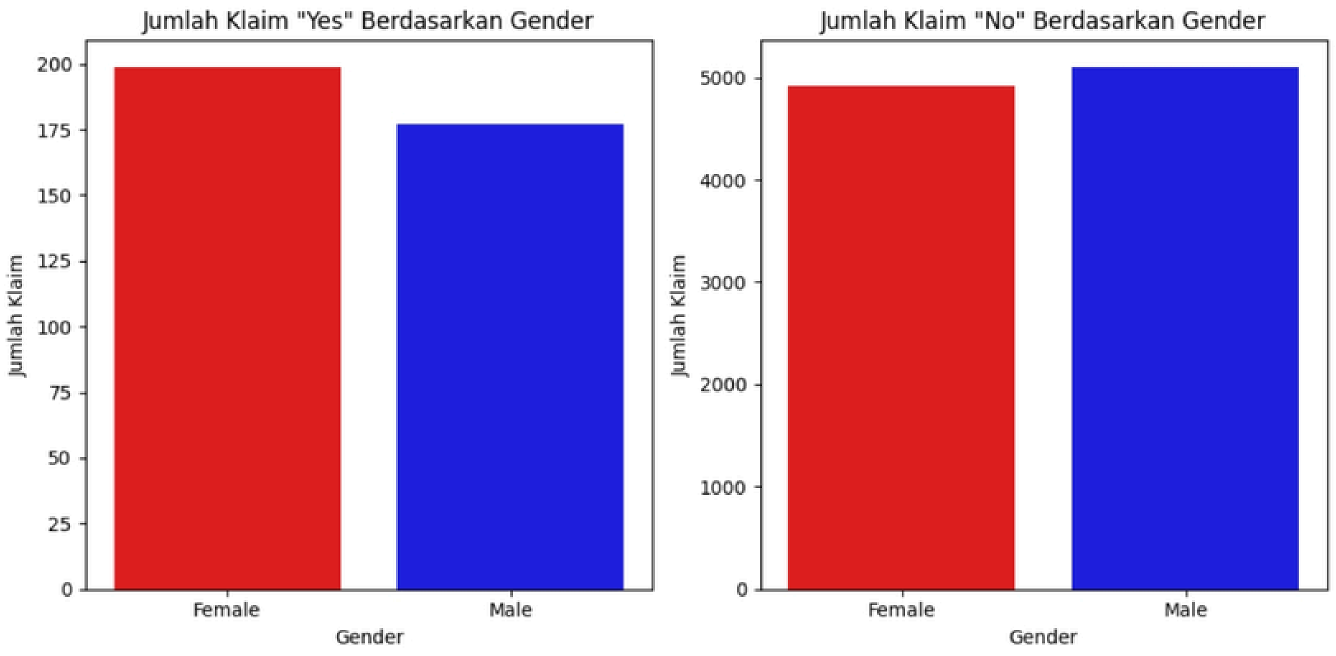
Scatterplot Insights



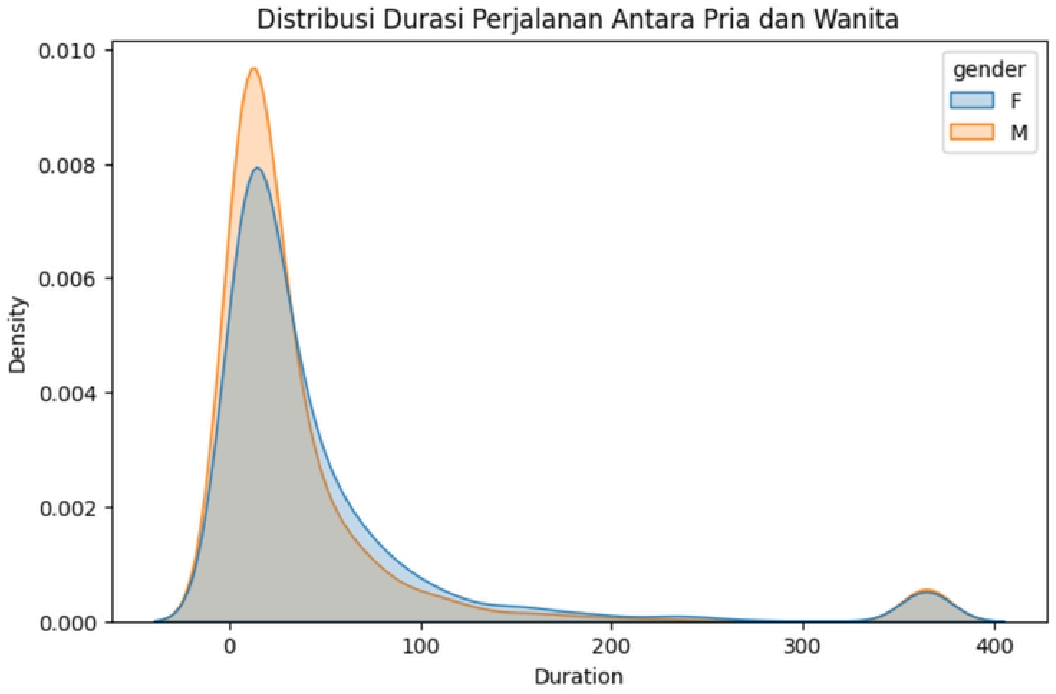
Barplot Conclusion

Exploratory data analysis: Handling Gender missing value

- **Gender Correlation:** Initial analysis showed minor differences in claim counts between males and females. Conducted a Chi-Square test, which confirmed no significant difference in insurance claim rates.
- **Travel Duration Insights by Gender:** KDE plot shows similar distributions for both genders, especially for short trips (0-50 days), indicating no significant preference differences.
- **Action Taken:** As no risk difference was found, missing values in the Gender feature were filled with "Not Specified" to retain all data for modeling.



Gender Correlation



Travel Duration Insights by Gender

Data Preprocessing

Action	Description
Binning	Applied to Age feature into categories defined by Allianz Travel's age requirements: 0-17 (Child Travelers), 28-30 (Young Adult Travelers), 30-60 (Adult Travelers), and 60-79 (Senior Citizen Travelers). This binning reduces noise by grouping similar values.
One-hot encoding	Applied to Agency Type and Gender features. Since these are nominal features, enabling the model to differentiate between them effectively.
Binary encoding	Applied to the Product Name and Destination features. These categorical features contain numerous categories without a specific order, making binary encoding an efficient choice.
Ordinal encoding	Applied to the binned Age feature, which represents categories with an inherent order. Senior Citizen Travelers (highest risk) are ranked first, followed by Adult Travelers, Young Adult Travelers, and Child Travelers, based on health considerations and claim frequency. This encoding assigns numerical values reflecting the order.

Testing Classification Algorithms

- **Cross-Validation:** Used Stratified K-Fold to assess model performance, ensuring consistent class proportions across folds.
- **Pipeline Automation:** Built a structured Pipeline to automate data processing and model training for consistent results.
- **Metric:** Evaluated models using recall to align with business objectives.
- **Results:** Compared average scores and standard deviations to determine the best algorithm. **Logistic Regression emerged as a top performer and was selected for its robustness and reliability**.

Training recall score

	Model	Average Train Score	Std Train Score
0	Logistic Regression	0.663791	0.008972
1	K-Nearest Neighbors	0.004348	0.008696
2	Decision Tree	0.117064	0.032917
3	Random Forest	0.069378	0.008322
4	AdaBoost	0.639972	0.019896
5	Gradient Boosting	0.637728	0.039367
6	XGBoost	0.004348	0.005325
7	LightGBM	0.481604	0.026026

Test recall score

	Model	Test Score
0	Logistic Regression	0.704348
1	K-Nearest Neighbors	0.008696
2	Decision Tree	0.095652
3	Random Forest	0.060870
4	AdaBoost	0.730435
5	Gradient Boosting	0.704348
6	XGBoost	0.000000
7	LightGBM	0.521739

Handling Imbalance Data

Resampling Techniques Tested:

1. SMOTE: Creates synthetic samples for minority class.
2. SMOTEENN: Combines SMOTE with Edited Nearest Neighbours to add synthetic data and remove noisy samples.
3. ADASYN: Adaptive Synthetic Sampling to generate minority samples.

Outcome: SMOTEENN delivered the best performance, balancing the dataset by increasing minority samples while reducing noise, resulting in improved model accuracy.

```
smote = SMOTE(random_state=42, n_jobs=-1)
smoteenn = SMOTEENN(random_state=42, n_jobs=-1)
adasyn = ADASYN(random_state=42, n_jobs=-1)

resample_list = [smote, smoteenn, adasyn]

best_resampler = None
best_score_train = 0
best_score_test = 0
best_X_train_resampled = None
best_y_train_resampled = None

for resampler in resample_list:
    X_train_resampled, y_train_resampled = resampler.fit_resample(X_train, y_train)
```

```
Teknik resampling terbaik: SMOTEENN

Skor recall terbaik pada data train: 0.7433362683102538

Skor recall terbaik pada data test: 0.6869565217391305
```

Hyperparameter Tuning Best Algorithm

Created a Pipeline for preprocessing, resampling, and modeling to automate and streamline the workflow.

In [75]:
```
logreg_final_model = LogisticRegression(random_state=42)
resampling = SMOTEENN(random_state=42)
```

In [76]:
```
logreg_final_model = ImbPipeline([
    ('preprocessor', preprocessor),
    ('resampling', resampling),
    ('model', logreg_final_model)
])
```

In [77]:
```
hyperparam_space = {
    'model__penalty': ['l1', 'l2', 'elasticnet', 'none'],
    'model__C': [0.001, 0.01, 0.1, 1, 10, 100],
    'model__solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'],
    'model__max_iter': [100, 200, 300, 500, 1000],
}

random_search = RandomizedSearchCV(
    logreg_final_model,
    param_distributions = hyperparam_space,
    n_iter = 120,
    cv=skfold,
    scoring='recall',
    random_state=42,
    n_jobs=-1,
)
```

In [78]:
```
random_search.fit(X_train, y_train)
```

Results

- The **model achieved a recall of 73%**, effectively identifying high-risk customers and improving risk prediction accuracy.
- Because the data used is dummy data and the data classes are not balanced, so naturally the other scores look bad, but the focus of the metrics in this project is recall.

Classification Report Data Test:

	precision	recall	f1-score	support
0	0.99	0.67	0.80	7185
1	0.03	0.73	0.07	115
accuracy			0.68	7300
macro avg	0.51	0.70	0.43	7300
weighted avg	0.98	0.68	0.79	7300