Rakamin Academy

Insurance Upselling Intelligence

Stage 3
Data Wizard:

* Dzulfikar Hanif Maulana (Ketua)
* Abdul Hardia Amin
* Haerunnisa
* Nisrina Widya Nur Farhani

Latar Belakang

Perusahaan asuransi kendaraan X menghadapi tantangan dalam meningkatkan tingkat konversi pelanggan yang tertarik untuk menggunakan asuransi kendaraan. Pada tahap ini kami sebagai data scientist akan melakukan pemodelan oleh machine learning yang mencakup:

- **Machine Learning Modelling,** pemodelan machine learning dengan beberapa model klasifikasi.
- **Feature Importance**, mencari fitur yang paling penting atau berpengaruh pada model.

Overview Dataset

- Gender: Jenis kelamin pelanggan, dapat berupa "Male" atau "Female".
- Age: Usia pelanggan, adalah variabel numerik yang mewakili usia pelanggan.
- Driving_License: Variabel biner yang menunjukkan apakah pelanggan sudah memiliki SIM atau tidak.
- Region_Code: Kode wilayah unik untuk pelanggan, digunakan untuk mengidentifikasi wilayah pelanggan.
- Previously_Insured: Variabel biner yang menunjukkan apakah pelanggan sudah memiliki Asuransi Kendaraan atau tidak.
- Vehicle_Age: Usia kendaraan pelanggan, mungkin berupa "1-2 Years", "< 1 Year", atau "> 2 Years".
- Vehicle_Damage: Variabel biner yang menunjukkan apakah kendaraan pelanggan pernah mengalami kerusakan atau tidak.
- Annual_Premium: Jumlah premi yang harus dibayar pelanggan dalam setahun.
- Policy_Sales_Channel: Kode anonim yang menggambarkan saluran komunikasi dengan pelanggan.
- Vintage: Jumlah hari pelanggan telah menjadi pelanggan perusahaan.
- Response: Variabel target yang menunjukkan apakah pelanggan tertarik pada
- Asuransi Kendaraan atau tidak (1 untuk tertarik, 0 untuk tidak tertarik).

Rakamin Academy

Machine Learning

List Machine Learning

Berikut adalah algoritma Machine Learning Classification yang akan digunakan dalam pemodelan:

1. Logistic Regression
2. Decision Trees
3. KNN (K-Nearest Neighbors)
4. Naive Bayes
5. XGBoost
6. Random Forest
7. Gradient Boosting
8. Neural Network

Keterangan Modeling

1. Metriks utama: Presisi (target 80% atau 0.8)
2. Setelah dataset di split, ada dua perlakuan yang dilakukan terhadap data:
    a. Standardization pada data train dan data test
    b. Handling class imbalance dengan SMOTE pada data train
3. Modeling dibagi menjadi dua tahap, tanpa SMOTE dan dengan SMOTE untuk melihat hasil pemodelan terbaik.
4. Cross Validation dan Tuning Hyperparameter dilakukan pada model dengan nilai presisi tertinggi
5. Fitur yang digunakan:

- dari dataset: 'Age', 'Region_Code','Previously_Insured', 'Gender_Label', 'Vehicle_Damage_Label', 'Vehicle_Age_1-2 Year', 'Vehicle_Age_< 1 Year', 'Vehicle_Age_> 2 Years'

- dari feature engineering: 'Previous_Claims_Count', 'Ownership_Duration'

Logistics Regression

- Tanpa SMOTE (presisi = 0.76)

```
Model: Logistic Regression
Train Accuracy: 0.8788
Test Accuracy: 0.8762
Train Precision: 0.7723
Test Precision: 0.7676
Train Recall: 0.8788
Test Recall: 0.8762
Train F1-Score: 0.8221
Test F1-Score: 0.8183
Confusion Matrix:
[[64468     0]
 [ 9113     0]]
```

- dengan SMOTE (presisi = 0.9027)

```
Model: Logistic Regression
Train Accuracy: 0.7831
Test Accuracy: 0.6416
Train Precision: 0.8315
Test Precision: 0.9027
Train Recall: 0.7831
Test Recall: 0.6416
Train F1-Score: 0.7749
Test F1-Score: 0.7017
Confusion Matrix:
[[38310 26158]
 [  216  8897]]
```

Decision Trees

- Tanpa SMOTE (presisi = 0.82)

```
Model: Decision Tree
Train Accuracy: 0.9689
Test Accuracy: 0.8302
Train Precision: 0.9687
Test Precision: 0.8211
Train Recall: 0.9689
Test Recall: 0.8302
Train F1-Score: 0.9674
Test F1-Score: 0.8255
Confusion Matrix:
[[58776  5692]
 [ 6800  2313]]
```

- dengan SMOTE (presisi = 0.82)

```
Model: Decision Tree
Train Accuracy: 0.9813
Test Accuracy: 0.8182
Train Precision: 0.9814
Test Precision: 0.8241
Train Recall: 0.9813
Test Recall: 0.8182
Train F1-Score: 0.9813
Test F1-Score: 0.8210
Confusion Matrix:
[[57425  7043]
 [ 6337  2776]]
```

KNN

- Tanpa SMOTE (presisi = 0.82)

```
Model: KNN
Train Accuracy: 0.8972
Test Accuracy: 0.8563
Train Precision: 0.8816
Test Precision: 0.8221
Train Recall: 0.8972
Test Recall: 0.8563
Train F1-Score: 0.8824
Test F1-Score: 0.8347
Confusion Matrix:
[[61472  2996]
 [ 7574  1539]]
```

- dengan SMOTE (presisi = 0.85)

```
Model: KNN
Train Accuracy: 0.8931
Test Accuracy: 0.7569
Train Precision: 0.8982
Test Precision: 0.8523
Train Recall: 0.8931
Test Recall: 0.7569
Train F1-Score: 0.8927
Test F1-Score: 0.7909
Confusion Matrix:
[[50163 14305]
 [ 3581  5532]]
```

Naive Bayes

- Tanpa SMOTE (presisi = 0.89)

```
Model: Naive Bayes
Train Accuracy: 0.6962
Test Accuracy: 0.6995
Train Precision: 0.8964
Test Precision: 0.8951
Train Recall: 0.6962
Test Recall: 0.6995
Train F1-Score: 0.7489
Test F1-Score: 0.7506
Confusion Matrix:
[[43150 21318]
 [  793  8320]]
```

- dengan SMOTE (presisi = 0.9032)

```
Model: Naive Bayes
Train Accuracy: 0.7840
Test Accuracy: 0.6416
Train Precision: 0.8332
Test Precision: 0.9032
Train Recall: 0.7840
Test Recall: 0.6416
Train F1-Score: 0.7757
Test F1-Score: 0.7017
Confusion Matrix:
[[38295 26173]
 [  196  8917]]
```

XG Boost

- Tanpa SMOTE (presisi = 0.82)

```
Model: XGBoost
Train Accuracy: 0.8798
Test Accuracy: 0.8761
Train Precision: 0.8691
Test Precision: 0.8288
Train Recall: 0.8798
Test Recall: 0.8761
Train F1-Score: 0.8252
Test F1-Score: 0.8196
Confusion Matrix:
[[64417    51]
 [ 9064    49]]
```

- dengan SMOTE (presisi = 0.88)

```
Model: XGBoost
Train Accuracy: 0.8303
Test Accuracy: 0.7402
Train Precision: 0.8438
Test Precision: 0.8849
Train Recall: 0.8303
Test Recall: 0.7402
Train F1-Score: 0.8287
Test F1-Score: 0.7825
Confusion Matrix:
[[46933 17535]
 [ 1579  7534]]
```

Random Forest

- Tanpa SMOTE (presisi = 0.82)

```
Model: Random Forest
Train Accuracy: 0.9688
Test Accuracy: 0.8461
Train Precision: 0.9681
Test Precision: 0.8211
Train Recall: 0.9688
Test Recall: 0.8461
Train F1-Score: 0.9680
Test F1-Score: 0.8317
Confusion Matrix:
[[60396  4072]
 [ 7251  1862]]
```

- dengan SMOTE (presisi = 0.82)

```
Model: Random Forest
Train Accuracy: 0.9813
Test Accuracy: 0.8207
Train Precision: 0.9813
Test Precision: 0.8295
Train Recall: 0.9813
Test Recall: 0.8207
Train F1-Score: 0.9813
Test F1-Score: 0.8249
Confusion Matrix:
[[57340  7128]
 [ 6064  3049]]
```

Gradient Boosting

- Tanpa SMOTE (presisi = 0.76)

```
Model: Gradient Boosting
Train Accuracy: 0.8788
Test Accuracy: 0.8762
Train Precision: 0.7723
Test Precision: 0.7676
Train Recall: 0.8788
Test Recall: 0.8762
Train F1-Score: 0.8221
Test F1-Score: 0.8183
Confusion Matrix:
[[64468     0]
 [ 9113     0]]
```

- dengan SMOTE (presisi = 0.89)

```
Model: Gradient Boosting
Train Accuracy: 0.8149
Test Accuracy: 0.7100
Train Precision: 0.8395
Test Precision: 0.8948
Train Recall: 0.8149
Test Recall: 0.7100
Train F1-Score: 0.8115
Test F1-Score: 0.7591
Confusion Matrix:
[[43996 20472]
 [  866  8247]]
```

Neural Network

- Tanpa SMOTE (presisi = 0.85)

```
Model: Neural Network
Train Accuracy: 0.8788
Test Accuracy: 0.8762
Train Precision: 0.8447
Test Precision: 0.8529
Train Recall: 0.8788
Test Recall: 0.8762
Train F1-Score: 0.8224
Test F1-Score: 0.8186
Confusion Matrix:
[[64463     5]
 [ 9102    11]]
```

- dengan SMOTE (presisi = 0.89)

```
Model: Neural Network
Train Accuracy: 0.7964
Test Accuracy: 0.6927
Train Precision: 0.8212
Test Precision: 0.8983
Train Recall: 0.7964
Test Recall: 0.6927
Train F1-Score: 0.7924
Test F1-Score: 0.7451
Confusion Matrix:
[[42469 21999]
 [  610  8503]]
```

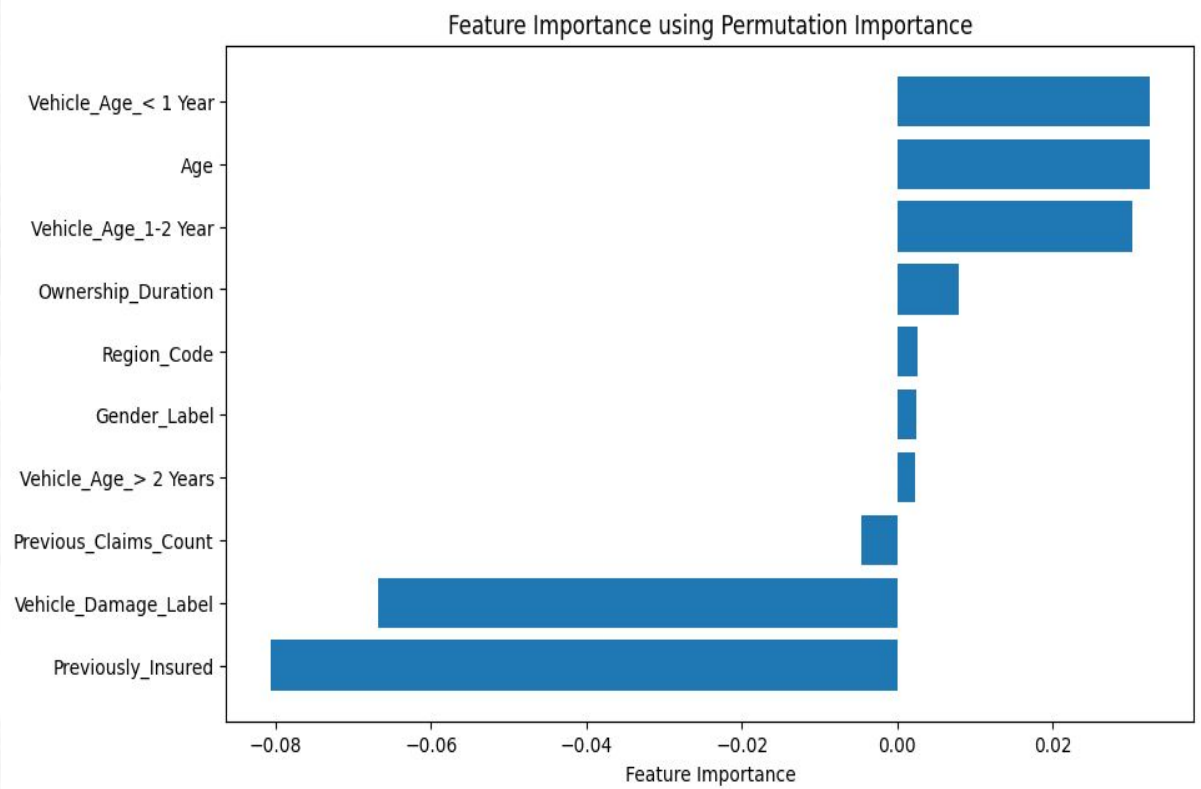
Kesimpulan Modeling

- Berdasarkan hasil algoritma-algoritma yang dicoba, nilai presisi tertinggi dimiliki oleh algoritma Naive Bayes (sebelum SMOTE dan setelah SMOTE).
- Cross validation dan tuning hyperparameter menggunakan Naive Bayes:
  - Cross Validation sebelum SMOTE (perbandingan nilai evaluasi train dengan test menunjukkan best fit)
    mean precision train: 0.89 ; mean precision test: 0.9014
  - Tuning Hyperparameter sebelum SMOTE (nilai presisi sama dengan sebelum tuning)
    param var soothing dengan hasil tertinggi: 0.89
  - Cross Validation setelah SMOTE (perbandingan nilai evaluasi train dengan test menunjukkan underfit)
    mean precision train: 0.83 ; mean precision test: 0.9014
  - Tuning Hyperparameter setelah SMOTE (nilai presisi menjadi lebih rendah)
    param var soothing dengan hasil tertinggi: 0.83
- Oleh karena itu karena nilai presisi naive bayes tanpa smote tinggi dan sudah best fit, maka kedepannya akan digunakan model Naive Bayes tanpa SMOTE dan tanpa tuning.

Feature Importance

Feature Importance

- Dari 8 algoritma yang diuji coba, didapatkan performa yang paling baik dengan menggunakan algoritma klasifikasi Naive bayes, yang menunjukkan hasil evaluasi yang best fit, dengan nilai presisi 89% baik pada data latih maupun data uji. Selanjutnya akan kita lakukan tuning hyperparameter.
- Dari model yang telah dibentuk, kita dapat mengetahui feature mana yang berperan penting untuk meningkatkan hasil klasifikasi yang disebut feature importance. Hasil dari feature importance ditujukkan dibawah ini.



- Selanjutnya feature importance di samping dapat diljadikan landasan untuk feature selection pada klasifikasi selanjutnya untuk menghasilkan hasil evaluasi yang lebih baik.

Business Insight

Insight Bisnis

1. **Vehicle_Age_1-2 Year**: Kendaraan yang berusia 1-2 tahun mungkin memiliki risiko klaim yang lebih rendah.
2. **Vehicle_Age_<1 Year**: Kendaraan baru cenderung lebih aman karena masih dalam kondisi optimal dan mungkin dilengkapi dengan teknologi keamanan terbaru.
3. **Vehicle_Age_>2 Years**: Kendaraan yang lebih tua cenderung memiliki risiko klaim yang lebih tinggi.
4. **Ownership_Duration**: Pemilik kendaraan lama cenderung lebih berhati-hati dan memiliki risiko klaim yang lebih rendah.
5. **Previously Insured**: Pelanggan yang sudah diasuransikan sebelumnya cenderung lebih aman dan memiliki risiko klaim yang lebih rendah.
6. **Vehicle_Damage_Label:** Kendaraan yang pernah mengalami kerusakan memiliki risiko klaim yang lebih tinggi.
7. **Previous_Claims_Count:** Riwayat klaim mempengaruhi risiko klaim di masa depan.

Action Items

1. **Diskon untuk Kendaraan Baru dan 1-2 Tahun**: Berikan diskon premi untuk kendaraan yang berusia kurang dari 2 tahun untuk menarik lebih banyak pelanggan dan mengurangi risiko klaim.
2. **Paket Asuransi untuk Kendaraan Tua**: Tawarkan paket asuransi khusus dengan premi yang lebih tinggi atau cakupan tambahan untuk kendaraan yang berusia lebih dari 2 tahun
3. **Fokus pada Pelanggan Baru**: Buat kampanye untuk menarik pelanggan baru yang belum pernah diasuransikan sebelumnya dan berikan insentif untuk pelanggan yang sudah diasuransikan untuk tetap bersama perusahaan.