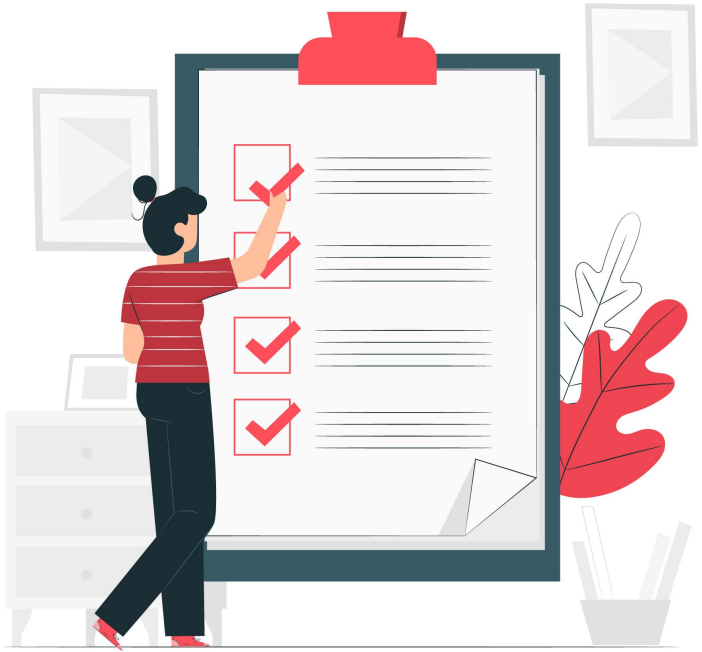
Penggunaan Metode Handling Imbalance Dataset pada Kasus Fake Job Dataset

Naufal Hilmi Irfandi
1806186673

Outline

---

1. Deskripsi Dataset
2. Permasalahan
3. Penyelesaian Masalah
4. Perbandingan Performa
5. Penutup



Deskripsi Dataset

Dataset Fake Job Posting

Merupakan dataset yang berisikan lowongan kerja yang merupakan lowongan kerja asli maupun lowongan kerja palsu.

Penjabaran isi Dataset

17880 Data lowongan pekerjaan dan 16 attribute yang mendeskripsikan lowongan pekerjaan.

Terdapat 2 Class yang membagi data dimana 0 untuk lowongan pekerjaan asli dan 1 untuk lowongan pekerjaan palsu

Permasalahan

Data Imbalance

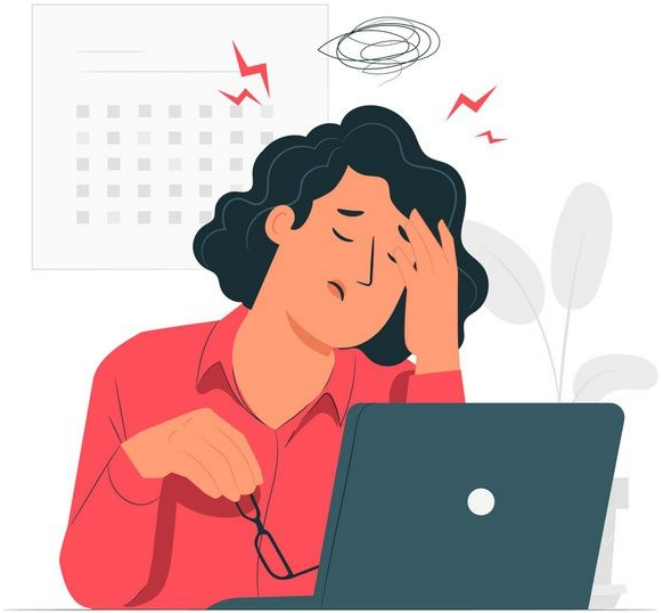
```
In [33]: counter = Counter(target_fake_job)
         print(counter)

         Counter({0: 17014, 1: 866})
```

Pembagian class 0 dan 1 tidak balanced dimana **95%** merupakan data dengan class 0.

Alasan

Data imbalance akan menyebabkan bias pada Machine Learning yang digunakan. Machine akan cenderung memprediksi data baru ke class yang dominan walaupun salah.

Dampak



Ini berdampak timbulnya **False Positive** dan **False Negatif** yang banyak. Dimana False Positive dan False Negatif sangat tidak diinginkan pada dataset Fake Job Posting.

Penyelesaian Masalah

Penggunaan Metrik dan Teknik

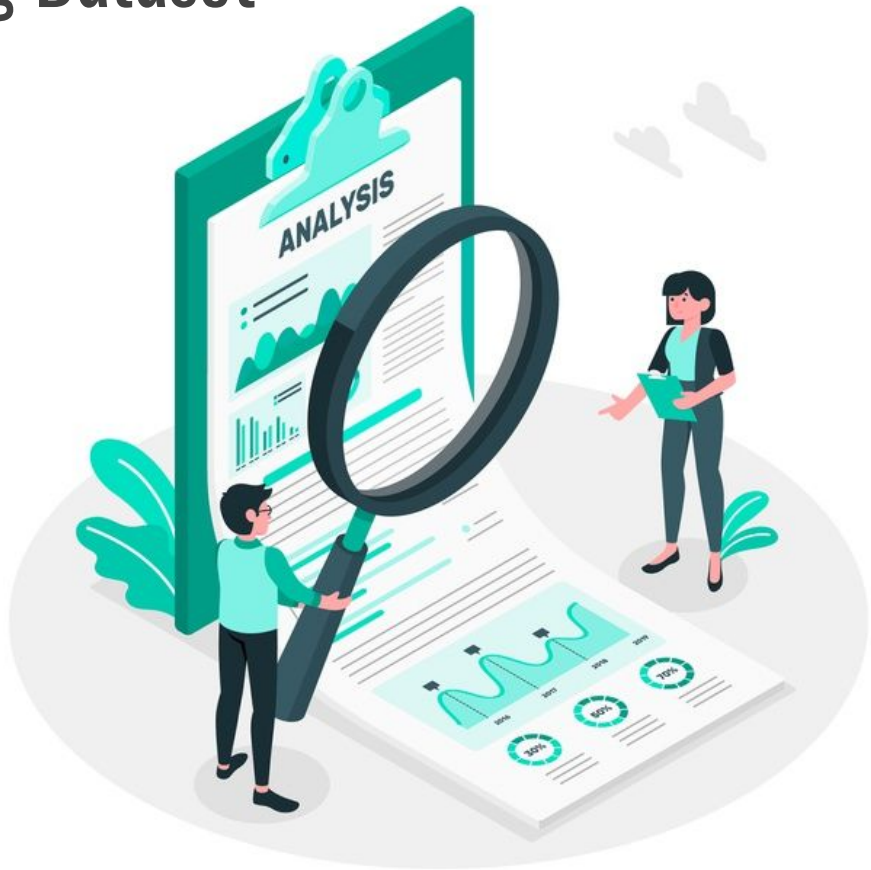
Menggunakan metrik **roc_auc_score** dan teknik,

1. SMOTE (Over-Sampling)
2. Near Miss (Under-Sampling)
3. SMOTE dan ENN (Combine)

Teknik ini akan dibandingankan dengan hasil metrik roc_auc_score data awal.

Data Processing Fake Job Posting Dataset

1. **Pengecekan Null Data.**
   Null data akan digantikan dengan nilai baru.
2. **Pemilihan Attribute Pendukung.**
   Diambil 5 attribute yang cukup merepresentasikan dan 5 attribute ini merupakan kategorik attribute.



Teknik SMOTE (Over-Sampling)

Teknik Smote (Over Sampling)

```
oversample = SMOTE()
X_over, y_over = oversample.fit_resample(data_fake_job_kategorik_2, target_fake_job_2)
```

```
counter = Counter(y_over)
print(counter)
```

```
Counter({0: 17014, 1: 17014})
```

Membuat jumlah class seimbang ke jumlah terbanyak.

Teknik Near Miss

Teknik Near Miss (Under Sampling)

```
undersample = NearMiss(version=1, n_neighbors=3)
X_under, y_under = undersample.fit_resample(data_fake_job_kategorik_2, target_fake_job_2)
```

```
counter = Counter(y_under)
print(counter)
```

```
Counter({0: 866, 1: 866})
```

Membuat jumlah class seimbang ke nilai tersedikit.

Teknik SMOTE dan ENN

Teknik Smote ENN (Combine)

```
combine = SMOTEENN()
X_combine, y_combine = combine.fit_resample(data_fake_job_kategorik_2, target_fake_job_2)
```

```
counter = Counter(y_combine)
print(counter)
```

```
Counter({0: 10744, 1: 5207})
```

Membuat jumlah class seimbang dimana akan menambahkan data ke data terbanyak namun menghilangkan data noisy.

Perbandingan Hasil

Perbandingan ROC AUC score

	Normal	SMOTE	NearMiss	SMOTEENN
Naive Bayes	0.5	0.737	0.734	0.933
KNN (n=12)	0.57	0.782	0.757	0.997
SVC (C=0.001, tol=pow(10,-5), loss='hinge', max_iter=5000)	0.57	0.782	0.757	0.997
LinearSVC (C=0.001, tol=pow(10,-5), loss='hinge', max_iter=5000)	0.57	0.782	0.757	0.997

Penutup

Kesimpulan

Dari perbandingan diatas, metode **SMOTE dan ENN** merupakan yang terbaik dikarenakan memiliki nilai roc_auc_score yang tertinggi.

Terima Kasih

Visual Art by https://storyset.com/

Referensi

---

1. Slide Mata Kuliah Sains Data, Imbalance Classification