

TUGAS UTS OULAD

Nama : Putri Maharani Isnainiyah

NIM : 202010370311355

1. Create dataset (integration of multiple dataset based on your knowledge about OULAD)
—> ALL OF DATA AMOUNT, not recommendation using sample.
2. Create Summary Data and Exploratory Data Analysis and describe clearly.
3. Find at least A REFERENCE INTERNATIONAL JOURNAL about modelling with machine learning/ deep learning
4. Create machine learning model/ deep learning model from your dataset about oulad.
5. Describe clearly step by step in your modelling (preprocessing, what model that you should use, evaluation)
6. Give conclusion about model and recommendation.
7. Collect in PDF. This PDF must include:
 1. Explanation about integration of dataset. Why and what data that you use from oulad
 2. Your dataset sample (head or tail dataset)
 3. Explanation about modelling (preprocessing until evaluation)
 4. Conclusion and recommendation
 5. Source code
 6. JOURNAL REFERENCE that you used (link journal and screenshot the title, author, and abstract)

Explanation about integration of dataset —————

Dataset yang dibuat adalah “*Student Performance Dataset*”. Tujuan dibuatnya dataset ini untuk melakukan prediksi final result siswa. Dengan adanya prediction ini, mahasiswa akan dapat diprediksi lulus atau tidaknya hanya dengan melihat beberapa features.

Dataset tersebut merupakan gabungan dari data-data berikut.

a. Assessment

Data *Assessment* berisi tentang informasi tentang penilaian dalam presentasi modul. Biasanya setiap presentasi memiliki sejumlah penilaian yang dilanjutkan dengan ujian akhir.

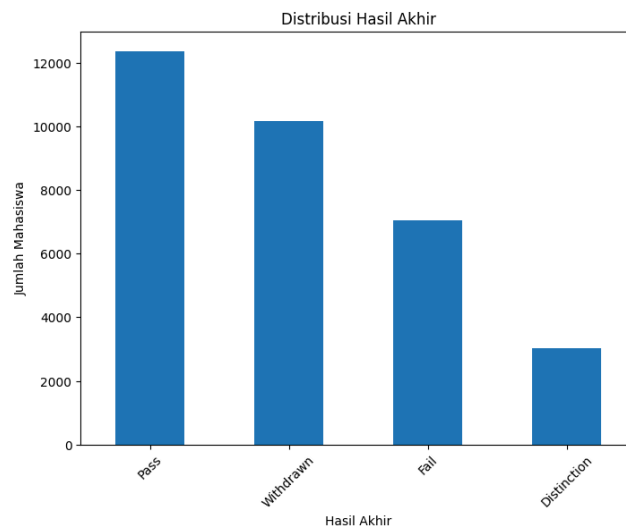
b. Student Assessment

Data *Student Assessment* berisi tentang hasil dari assessment murid-murid.

c. Student Info

Data *Student Info* berisi informasi demografis tentang siswa beserta hasilnya.

Dataset ini memuat 21 fitur. Dari data-data tersebut dapat memprediksi apakah siswa akan pass, withdrawn, fail atau distinction. Dari data yang didapat, berikut adalah visualisasi distribusi dari final result dari para siswa.



Dataset Sample

Berikut adalah sampel dari dataset “Student Performance Dataset”.

	code_m...	a...	a...	c...	c...	d...	da...	dis...	final...	imd...	is_ban...	num_of...	region	score	studied_credits	weight	Record ...
1..	BBB	3...	C...	B...	2...	16...	168	N	Pass	50-60%	0	0	Scotland	100.0	60	1.0	8
2..	BBB	3...	C...	B...	2...	54...	56	N	Pass	50-60%	0	0	Scotland	100.0	60	1.0	8
3..	BBB	3...	C...	B...	2...	54...	56	N	Pass	20-30%	0	0	Ireland	100.0	60	1.0	7
4..	CCC	3...	C...	C...	2...	18...	21	N	Distin...	null	0	0	North Region	100.0	60	2.0	7
5..	DDD	0...	C...	D...	2...	51...	54	N	Pass	null	0	0	North Region	94.0	60	3.0	7
6..	CCC	0...	C...	C...	2...	18...	21	N	Distin...	null	0	0	North Region	100.0	120	2.0	7
7..	BBB	3...	C...	B...	2...	96...	102	N	Pass	50-60%	0	0	Scotland	100.0	60	1.0	7
8..	BBB	0...	C...	B...	2...	54...	56	N	Pass	30-40%	0	0	Wales	100.0	60	1.0	7
9..	BBB	0...	C...	B...	2...	47...	40	N	Pass	20-40%	0	0	East Anglia	100.0	60	1.0	7

MODELING

1. Preprocessing

a. Pengumpulan data

Data dikumpulkan dengan cara melakukan integrasi pada data *Assessment*, *Student Assessment* dan *Student Info*,

b. Pembersihan data (Data cleaning)

Pembersihan data yang dilakukan berupa mengidentifikasi dan menangani data yang kosong. Gambar berikut adalah gambar yang menunjukkan jumlah nilai null (missing values) dalam setiap kolom dari dataset.

```
code_module_x      0
code_presentation_x 0
id_assessment      0
assessment_type     0
date               4018
weight             0
id_student         0
date_submitted     0
is_banked          0
score              227
code_module_y      0
code_presentation_y 0
gender             0
region             0
highest_education  0
imd_band           9315
age_band           0
num_of_prev_attempts 0
studied_credits    0
disability         0
final_result       0
dtype: int64
```

Dapat dilihat dari gambar diatas, kolom “date” memiliki 4.018 nilai null, “score” memiliki 227 nilai null dan “imd_band” memiliki 9.315 nilai null. Nilai-nilai yang bernilai null tersebut akan dihapus.

```
code_module_x      0
code_presentation_x 0
id_assessment      0
assessment_type     0
date               0
weight             0
id_student         0
date_submitted     0
is_banked          0
score              0
code_module_y      0
code_presentation_y 0
gender             0
region             0
highest_education  0
imd_band           0
age_band           0
num_of_prev_attempts 0
studied_credits    0
disability         0
final_result       0
dtype: int64
```

c. Pemilihan features yang berkaitan

Fitur pada dataset berjumlah 21. Beberapa dari fitur tersebut tidak relevan dengan hasil final dari siswa. Maka dari itu, harus dilakukan pemilihan

fitur. Fitur-fitur yang tidak relevan akan di drop. Berikut adalah fitur yang relevan dengan hasil final dari siswa.

- 1) Assessment Type
- 2) Score
- 3) Gender
- 4) Highest education
- 5) IMD band
- 6) Age band
- 7) Num of prev attempts
- 8) Studied credits
- 9) Disability
- 10) Final result

d. Penyandian Label (Label Encode)

```
assessment_type    object
score              float64
gender             object
region            object
highest_education  object
imd_band          object
age_band          object
num_of_prev_attempts  int64
studied_credits    int64
disability         object
final_result       object
dtype: object
```

Beberapa fitur dari dataset tersebut merupakan kategorial (object). Agar fitur-fitur yang memiliki data type kategorik menjadi numerik, maka dilakukan label encoder.

```

assessment_type    int64
score              float64
gender             int64
region            int64
highest_education  int64
imd_band           int64
age_band           int64
num_of_prev_attempts int64
studied_credits    int64
disability         int64
final_result       int64
dtype: object

```

e. Pemisahan features dan labels

Pemisahan kolom-kolom dataset menjadi dua bagian terpisah, yaitu features dan labels. Labels yang digunakan dalam model ini adalah “*final_result*”, yang merupakan variabel target yang ingin diprediksi.

```

▶ features = result.drop(columns=['final_result'])
  labels = result['final_result']

```

f. Pembagian train set dan test set

Selanjutnya, dilakukan pembagian dataset menjadi train set sebesar 80% dan test set sebesar 20%. Train set yang terdiri dari 80% dari keseluruhan dataset, digunakan untuk melatih model machine learning, sehingga model dapat memahami pola dan hubungan dalam data. Sedangkan data tes yang terdiri dari 20% sisa data digunakan untuk menguji kinerja model.

```

X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.20, random_state=70)

```

2. Modeling

Metode yang digunakan dalam prediction machine learning ini adalah decision tree. Dengan 4 kelas, yaitu pass, withdrawn, fail atau distinction. Berikut adalah langkah membuat model decision tree dan melatih model tersebut pada train set.

```
# Membuat model Decision Tree
clf = DecisionTreeClassifier()

# Melatih model Decision Tree pada data latih
clf.fit(X_train, y_train)
```

3. Evaluasi

Setelah melatih model Decision Tree pada Train set, dilakukanlah prediksi pada test set. Berikut adalah hasil dari laporan klasifikasi model pada laporan ini.

	precision	recall	f1-score	support
0	0.50	0.55	0.52	6060
1	0.43	0.44	0.44	6741
2	0.71	0.70	0.70	23550
3	0.42	0.38	0.40	5113
accuracy			0.60	41464
macro avg	0.51	0.52	0.52	41464
weighted avg	0.60	0.60	0.60	41464

Dapat dilihat dari laporan klasifikasi tersebut, accuracy dari model ini adalah 60%. Dengan nilai Precision paling tinggi adalah sekitar 71% untuk kelas 2. Nilai recall tertinggi yaitu sekitar 70% untuk kelas 2. Dan untuk nilai F1-Score yang tertinggi adalah 70% untuk kelas 2.

Conclusion and Recommendation. —————

Data “*Student Performance Dataset*” merupakan dataset yang terdiri dari data Assessment, Student Assessment dan Student Info. Dataset ini berisi tentang fitur-fitur yang menggambarkan performa siswa. Pembuatan model dengan dataset ini bertujuan untuk melakukan prediksi apakah seorang siswa akan lulus atau tidak berdasarkan fitur-fitur yang berkaitan.

Proses Preprocessing yang dilakukan berupa pembersihan data dengan menghilangkan nilai null, pemilihan fitur yang relevan dari 21 fitur menjadi 10 fitur, melakukan label encoder untuk 8 fitur, pemilihan fitur dan label, serta pembagian data menjadi 80% train set dan 20% test set.

Selanjutnya, dilakukan pembuatan model Decision Tree yang digunakan untuk melatih dan memprediksi hasil akhir siswa, dengan empat kelas yang berbeda, yaitu pass,

withdrawn, fail atau distinction. Hasil evaluasi model menunjukkan tingkat akurasi sekitar 60%. Dengan nilai Precision paling tinggi adalah sekitar 71% untuk kelas 2. Nilai recall tertinggi yaitu sekitar 70% untuk kelas 2. Dan untuk nilai F1-Score yang tertinggi adalah 70% untuk kelas 2.

Dikarenakan tingkat akurasi berada di angka 60%, maka disarankan untuk melakukan eksplorasi berbagai teknik dan model machine learning lainnya, seperti Random Forest, Support Vector Machines atau Neural Network, untuk meningkatkan akurasi prediksi. Selain itu, analisis lebih mendalam tentang fitur yang paling mempengaruhi hasil akhir siswa dapat membantu dalam peningkatan model.

SOURCE CODE

Link Colab :

<https://colab.research.google.com/drive/1AMCQRyq8YT9gb5ggg62wIKoZHrly-yqx?usp=sharing>


Untuk Source Code, dilampirkan di halaman selanjutnya.

JOURNAL REFERENCE

The use of decision tree based predictive models for improving the culvert inspection process

Ce Gao, Hazem Elzarka*

Department of Civil and Architectural Engineering and Construction Management, University of Cincinnati, Cincinnati, OH 45221-0071, USA



ARTICLE INFO

Keywords:
Decision tree
Synthetic minority over-sampling technique
Imbalanced data

ABSTRACT

Culverts are important components of a roadway and should be properly maintained to ensure adequate road surface drainage and public safety. Culvert maintenance greatly relies on culvert inspection which is time consuming and requires a large number of skilled labor hours. Currently, State Departments of Transportation use rigid methods for scheduling culvert inspection based on one or two factors such as culvert size and/or condition. The objective of the research described in the paper is to develop a more intelligent scheduling system for culvert inspection to improve the utilization of limited resources. The proposed intelligent system first predicts the conditions of the culverts that are due for inspection in a given year and based on the prediction results, only schedule inspections for those predicted to be in poor condition. The prediction models utilized a Decision Tree algorithm together with the Synthetic Minority Over-Sampling Technique to deal with the highly imbalanced data in the culvert inventory database. The case study presented in the paper utilized 12,400 culvert records from the Ohio Department of Transportation to train and test the prediction models. The developed prediction models have achieved accuracies over 80% for the training set and 75% for the testing set and satisfactory values for the areas under the curve of 0.8. The case study concluded that by implementing the proposed intelligent culvert inspection scheduling system, the number of culverts needing inspections is reduced by 44%. Implementation of the proposed system could assist state and local agencies with prioritizing inspection of culverts needing attention while maximizing the use of limited resources. While this study is applied to culverts in Ohio, the proposed framework can be used on any similarly available culvert data set worldwide. The paper ends by providing suggestions to improve the quality of the data in culvert inventory databases.

Link : <https://sci-hub.se/10.1016/j.aei.2020.101203>