**Tugas Pra-UTS**

Dosen Pengampu : Vinna Rahmayanti Setyaning Nastiti S.Si., M.Si



Disusun oleh :

Erik Dwi Prasetya 202010370311508

**UNIVERSITAS MUHAMMADIYAH MALANG**

**FAKULTAS TEKNIK**

**INFORMATIKA**

**2023**

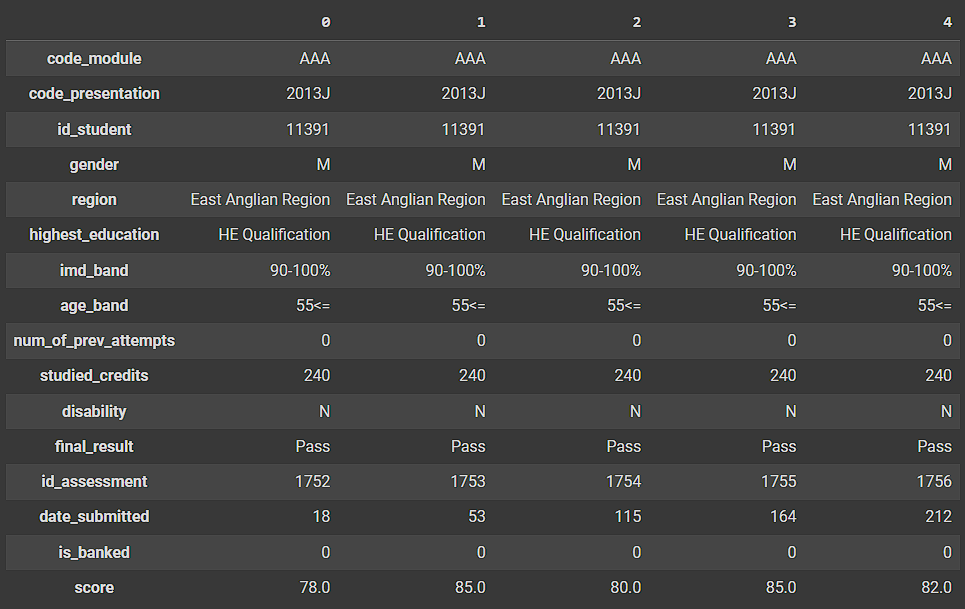
1. Create a dataset (integration of multiple dataset based on your knowledge about OULAD) —> ALL OF DATA AMOUNT, not recommendation using sample.
2. Create Summary Data and Exploratory Data Analysis and describe clearly.
3. Find at least A REFERENCE INTERNATIONAL JOURNAL about modeling with machine learning/ deep learning
4. Create a machine learning model/ deep learning model from your dataset about oulad.
5. Describe clearly step by step in your modeling (preprocessing, what model that you should use, evaluation)
6. Give conclusions about the 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 :

Kumpulan data ini berasalah dari Universitas Terbuka yaitu Universitas negeri Inggris yang juga memiliki jumlah mahasiswa sarjana terbanyak di Inggris. Ini adalah institusi akademik terbesar di Inggris (dan salah satu yang terbesar di Eropa) dengan 2 Juta mahasiswa terdaftar sejak didirikan pada tahun 1969. Sesuai dengan namanya, Universitas Terbuka sebagian besar dihuni oleh mahasiswa luar kampus yang milik Platform Pembelajaran Online Universitas Terbuka (Juga disebut sebagai "Lingkungan Pembelajaran Virtual (VLE)") yang digunakan mahasiswa di luar kampus untuk mengakses konten kursus, diskusi forum, mengirimkan penilaian dan memeriksa tanda tugas, dll. Terdiri dari 7 kursus yang dipilih (disebutkan sebagai modul dalam kumpulan data). Presentasi berbeda ditandai dengan huruf "B" dan "J" setelah tahun masing-masing untuk semester 2 dan semester 1. Selain itu, kumpulan data tersebut mencakup demografi siswa seperti lokasi, kelompok umur, disabilitas, tingkat pendidikan, jenis kelamin, dll. Nilai penilaian siswa, interaksi dengan Virtual Learning Environment (VLE) juga disertakan.

* 1. Your dataset sample (head or tail dataset) :

Description Column :

* code\_module – nama kode modul, yang berfungsi sebagai pengenal.
* code\_presentation – nama kode presentasi. Terdiri dari tahun dan “B” untuk presentasi yang dimulai pada bulan Februari dan “J” untuk presentasi yang dimulai pada bulan Oktober.
* id\_student – nomor identifikasi unik untuk siswa.
* gender – jenis kelamin siswa.
* region – mengidentifikasi wilayah geografis, tempat tinggal siswa saat mengambil presentasi modul.
* Highest\_Education – tingkat pendidikan siswa tertinggi saat masuk ke presentasi modul.
* imd\_band – menentukan pita Indeks Kerusakan Berganda dari tempat siswa tinggal selama presentasi modul.
* age\_band – kelompok usia siswa.
* num\_of\_prev\_attempts – berapa kali siswa mencoba modul ini.
* study\_credits – jumlah total SKS untuk modul yang sedang dipelajari siswa.
* disabilitas – menunjukkan apakah siswa telah dinyatakan cacat.
* final\_result – hasil akhir siswa dalam presentasi modul.



* 1. Explanation about modeling (preprocessing until evaluation) :
     1. Pre-Processing :
        1. Beberapa perbaikan pada kerangka data sebelum memasukkannya ke model. Pertama mengkodekan semua variabel kategori. Di sini melakukan pengkodean label daripada pengkodean one-hot, hanya karena ini berfungsi untuk algoritma berbasis pohon, dan lebih mudah dibaca jika nanti kita ingin memeriksa setiap pohon secara visual.

| def create\_Xy(final\_df):  X = final\_df.drop(['final\_result','id\_student','imd\_band'],axis = 1)  column\_names = X.columns  y = final\_df['final\_result']  le = LabelEncoder()  encode\_dict = {}  Xcat\_features = ['code\_module', 'code\_presentation',  'gender', 'region',  'highest\_education',  'age\_band','disability',  ]  for cat\_feature in Xcat\_features:  X[cat\_feature] = le.fit\_transform(X[cat\_feature])  encode\_dict[cat\_feature] = le.classes\_  y = le.fit\_transform(y)  encode\_dict['final\_result'] = le.classes\_  X = X.to\_numpy()  return X,y,column\_names,encode\_dict |
| --- |

* + - 1. Kode berikut menangani potensi kolinearitas dalam data. Meskipun kolinearitas tidak berdampak pada model Random Forest, hal ini mungkin berdampak negatif pada analisis kepentingan fitur yang kami lakukan setelahnya.

| corr = spearmanr(X).correlation  corr\_linkage = hierarchy.ward(corr)  cluster\_ids = hierarchy.fcluster(corr\_linkage, 1, criterion='distance')  cluster\_id\_to\_feature\_ids = defaultdict(list)  for idx, cluster\_id in enumerate(cluster\_ids):  cluster\_id\_to\_feature\_ids[cluster\_id].append(idx)  selected\_features = [v[0] for v in cluster\_id\_to\_feature\_ids.values()]  X = X[:, selected\_features]  column\_names = column\_names[selected\_features] |
| --- |

* + 1. Modeling :
       1. Di sini menggunakan metode Random Forest karena model ini bagus untuk klasifikasi pada data tabular. dan juga bagus, yaitu tidak terlalu mempermasalahkan hyperparameter. Selain itu, ada implementasi langsung untuk pentingnya fitur permutasi.

| reg = RandomForestClassifier(n\_estimators = 200,  max\_features = 'sqrt',  min\_samples\_split = 10,  )  kf = KFold(n\_splits=5, shuffle = True)  def train(X,y,reg,kf):  if kf == False:  reg.fit(X,y)  else:  for train\_index, test\_index in kf.split(X):  X\_train, X\_test = X[train\_index], X[test\_index]  y\_train, y\_test = y[train\_index], y[test\_index]  reg.fit(X\_train,y\_train)  print(f'evaluate score: {reg.score(X\_test,y\_test)}') |
| --- |

* + - 1. Evaluasi (data train)

| train(X,y,reg,kf) |
| --- |
| Output :  evaluate score: 0.7504737839545167  evaluate score: 0.7372078332280481  evaluate score: 0.7532112023583912  evaluate score: 0.7523167649536647  evaluate score: 0.7377843302443134 |

* + - 1. Evaluasi berdasarkan hari

| days\_list = [120,150,180,210]  for days in days\_list:  score\_deadline = days  click\_deadline = days    score\_df = create\_score\_df(score\_deadline)  click\_df = create\_click\_df(click\_deadline)  final\_df = create\_final\_df(withdraw\_deadline, score\_df, click\_df)  X,y,column\_names,encode\_dict = create\_Xy(final\_df)    print(f'Number of days in data: {days}')  train(X,y,reg,kf)  print('\n') |
| --- |

* + - * 1. Hari ke-120

evaluate score: 0.7580916351408155

evaluate score: 0.7587221521647751

evaluate score: 0.7614543926019336

evaluate score: 0.75010508617066

evaluate score: 0.7527853689299979

* + - * 1. Hari ke-150

evaluate score: 0.7719555648710962

evaluate score: 0.7633619786208342

evaluate score: 0.770278767553972

evaluate score: 0.7784531544749529

evaluate score: 0.7719077568134172

* + - * 1. Hari ke-180

evaluate score: 0.7912824811399832

evaluate score: 0.7954735959765298

evaluate score: 0.7816429170159263

evaluate score: 0.7942162615255658

evaluate score: 0.7917015926236379

* + - * 1. Hari ke-210

evaluate score: 0.8085061805992039

evaluate score: 0.8041064320134088

evaluate score: 0.7967735177037503

evaluate score: 0.80720871751886

evaluate score: 0.8049036043587594

* + 1. Feature Importance :

| from sklearn.inspection import permutation\_importance  train(X,y,reg,kf = False)  importance = permutation\_importance(reg, X, y, n\_repeats=10, random\_state = 0)  importance\_mean = np.round(importance['importances\_mean'],3)  importance\_table = pd.DataFrame({'importance': importance\_mean,  'column': column\_names,  })  importance\_table.sort\_values(by = 'importance', ascending = False) |
| --- |

* 1. Conclusion and recommendation

Dalam buku catatan ini, kami membahas pembuatan model Random Forest untuk memprediksi hasil akhir siswa berdasarkan kinerja awal mereka. Model ini bekerja cukup baik, dengan akurasi validasi silang sebesar 80%. Kami akan merekomendasikan instruktur kursus di masa depan untuk menjalankan model tersebut sekitar hari ke 180, dan menjangkau siswa yang kemungkinan besar akan gagal. Bagi siswa yang diprediksi lulus, sebaiknya instruktur melakukan pengecekan kewarasan dengan cara memeriksa siswa yang rata-rata nilainya di bawah ambang batas tertentu. Ambang batas ini harus dipilih oleh instruktur berdasarkan pengalaman masa lalu. Kami tidak merekomendasikan ambang batas universal apa pun: karena setiap kursus memiliki struktur yang berbeda, kami harus merekomendasikan ambang batas yang sangat rendah sehingga akan menjadi kontraproduktif. Untuk analisis versi mendatang, kami ingin menggunakan data skor yang disediakan dengan lebih baik daripada sekadar membuat rata-ratanya. Kami juga ingin mengeksplorasi lebih jauh siswa yang salah klasifikasi dan melihat apakah kami dapat merekayasa lebih banyak fitur yang dapat membantu menyempurnakan model saat ini. Terakhir, kami ingin menulis ulang analisis kami dengan CatBoost sehingga kami dapat melatih GPU lebih cepat.

* 1. Source code

Link github : <https://github.com/ED-Prasetya/BigData_PraUTS>

* 1. JOURNAL REFERENCE that you used (link journal and screenshot the title, author, and abstract)
     1. Link : <https://ieeexplore.ieee.org/abstract/document/9300123>
     2. Judul : MOOC`s Student Results Classification by Comparing PNN and other Classifiers with Feature Selection
     3. Author : Ahmed Mahmoud Nazif, Ahmed Ahmed Hesham Sedky, Osama Mohamed Badawy
     4. Abstract:

An urgent necessity during year 2020, it became a must that all universities around the world to move from traditional classrooms, COVID-19 epidemic forced schools and universities to change their plans by e-learning strategy and/or hosting Massive Open Online Courses (MOOCs). Since dropouts and failure rates of MOOCs' students is a well noticed problem, this paper proposes a new methodology in classifying students' results throughout MOOCs modules. By using Open University Learning Analytics Dataset (OULAD) and applying modern machine learning techniques, it becomes more useful to monitor factors affecting student performance and achievement. The proposed methodology contributed a new model that uses various feature selection algorithms and various classification algorithms including Probabilistic Neural Network (PNN) and other classification algorithms. Results showed that using certain feature selection algorithms in combination with PNN resulted in enhancing trend exploration and accuracy.