UAS Komputasi Lanjut dan Big Data Option 1 : Drug Recommender Systems

Sayyid Abdullah 2206130800

June 25, 2023

Abstract

Drug recommender systems based on decision tree algorithms have gained significant attention in the healthcare industry. These systems utilize decision trees, a popular machine learning technique, to provide personalized recommendations for medications based on patient characteristics, medical history, and other relevant factors. This abstract explores the application of decision trees in drug recommender systems, highlighting their advantages, challenges, and potential impact on patient care. Decision tree algorithms excel at handling both categorical and numerical features, making them suitable for analyzing diverse patient datasets. They partition the data based on various attributes, creating a hierarchical structure that allows for straightforward interpretation and understanding of the decision rules. The development of drug recommender systems using decision trees involves constructing decision tree models, training them on historical patient data, and using them to predict appropriate medications for new patients.

1 Pendahuluan

Di bidang kesehatan, pemilihan obat yang paling tepat untuk pasien merupakan tugas yang krusial dan memiliki dampak besar terhadap hasil pengobatan dan kesejahteraan pasien. Dengan kemajuan teknologi dan tersedianya data kesehatan yang luas, sistem rekomendasi obat telah menjadi alat berharga untuk membantu para profesional kesehatan dalam membuat keputusan yang terinformasi mengenai pilihan obat. Sistem-sistem ini menggunakan berbagai algoritma dan teknik, salah satunya adalah algoritma pohon keputusan, untuk memberikan rekomendasi obat yang dipersonalisasi berdasarkan karakteristik pasien dan faktor-faktor terkait lainnya.

Tujuan utama dari sistem rekomendasi obat adalah untuk mengoptimalkan perawatan pasien dengan menyarankan obat-obatan yang sesuai dengan kebutuhan, preferensi, dan persyaratan terapi individu. Algoritma pohon keputusan memberikan kerangka kerja yang transparan dan dapat diinterpretasikan dalam memberikan rekomendasi obat, sehingga memungkinkan para profesional kesehatan untuk memahami proses pengambilan keputusan yang mendasarinya. Algoritma ini membangun struktur hierarkis dari aturan-aturan keputusan yang mempartisi data berdasarkan fitur-fitur tertentu, sehingga memudahkan interpretasi dan pemahaman terhadap proses pengambilan keputusan tersebut. Secara keseluruhan, sistem rekomendasi obat berbasis pohon keputusan menawarkan pendekatan yang transparan dan dapat diinterpretasikan dalam memberikan rekomendasi obat. Dengan memanfaatkan algoritma pohon keputusan, sistem-sistem ini memungkinkan para profesional kesehatan untuk menganalisis data pasien, menghasilkan saran yang dipersonalisasi, dan meningkatkan hasil pengobatan pasien. Melalui penelitian dan pengembangan lebih lanjut, sistem rekomendasi obat berbasis pohon keputusan memiliki potensi besar untuk merevolusi pengambilan keputusan mengennai pemilihan obat. Pada penelitian ini akan dilakukan analisis rekomendasi obat menggunakan decision tree . Data yang digunakan diambil dari drug recommendations. Tools yang digunakan adalah google collab.

2 Metodologi

2.1 Decision Tree

Decision tree (pohon keputusan) adalah salah satu metode analisis data dan pembuatan keputusan yang digunakan dalam ilmu komputer, statistik, dan machine learning. Metode ini menggunakan struk-

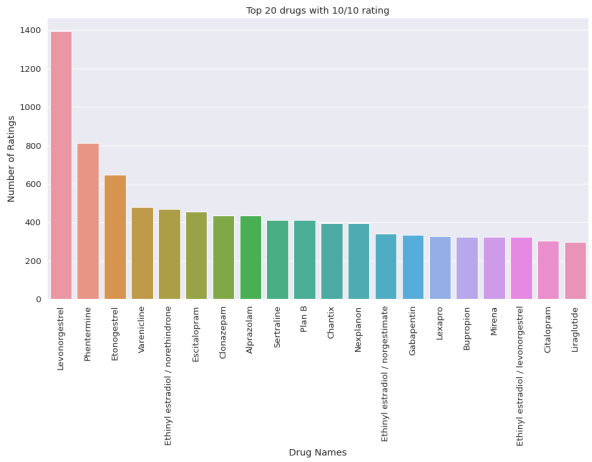

Figure 1: top 20 obat dengan rating 10/10

tur seperti pohon untuk menggambarkan keputusan dan konsekuensi yang terkait. Pada dasarnya, sebuah decision tree terdiri dari node-node yang merepresentasikan keputusan atau pernyataan, cabang-cabang yang merepresentasikan kemungkinan hasil atau konsekuensi dari keputusan tersebut, dan daun-daun yang merepresentasikan hasil akhir atau output. Decision tree biasanya digunakan untuk membuat keputusan pada masalah klasifikasi dan regresi, dengan mengambil beberapa variabel input dan menghasilkan prediksi atau klasifikasi pada variabel output. Metode ini juga sering digunakan dalam pembuatan keputusan bisnis dan manajemen, seperti dalam analisis risiko dan pengambilan keputusan investasi.

Algoritma dari decision tree dapat dijelaskan dalam beberapa langkah berikut:

1. Memilih variabel input: Langkah pertama adalah memilih variabel input yang akan digunakan dalam pembuatan decision tree. Variabel input ini harus memiliki pengaruh yang signifikan terhadap variabel output.

2. Membuat decision node: Setelah variabel input dipilih, kita membuat decision node pada level atas pohon keputusan. Decision node ini merepresentasikan pertanyaan atau keputusan yang akan diambil berdasarkan variabel input.

3. Membuat cabang-cabang: Berdasarkan nilai variabel input, kita membuat cabang-cabang yang merepresentasikan kemungkinan hasil atau konsekuensi dari keputusan tersebut.

4. Menentukan kondisi terminal: Kondisi terminal merupakan daun-daun pada pohon keputusan yang merepresentasikan hasil akhir atau output. Kondisi terminal ini dicapai ketika tidak ada lagi variabel input yang dapat ditambahkan pada cabang-cabang tersebut.

5. Membangun sub-tree: Langkah terakhir adalah membangun sub-tree dari decision node sampai kondisi terminal terpenuhi. Proses ini dilakukan secara rekursif hingga terbentuk decision tree yang lengkap.

3 Hasil dan Pembahasan

Langkah pertama adalah memvisualisasikan data drug recommendations nya. Dalam data tesebut terdapat sebanyak 6 variabel.Variabel yang divisualisasi berupa variabel drug name dan rating. Hasil visualiasi seperti Gambar 1. Pada Gambar 1, ditampilkan 20 obat dengan rating tertinggi. Obat dengan rating paling banyak adalah Levonorgestrel. selanjutnya divisualisasikan 20 jenis obat dengan rating palig kecil seperti pada Gambar 2. obat dengan rating kecil terbanyak adalah miconazole. Selanjutnya adalah memvisualisasikan top 10 kondisi penderita sepeerti Gambar 4. Kondisi paling banyak adalah birth control (kontrol kelahiran). selanjutnya memvisualiasikan obat yg digunakan oleh penderita dengan kondisi kontrol kelahiran seperti Gambar 5. Obat paling banyak digunakan berupa etonogestrel. Hapus nilai "condition" yang hanya memiliki satu obat yang terkait dengannya, lalu

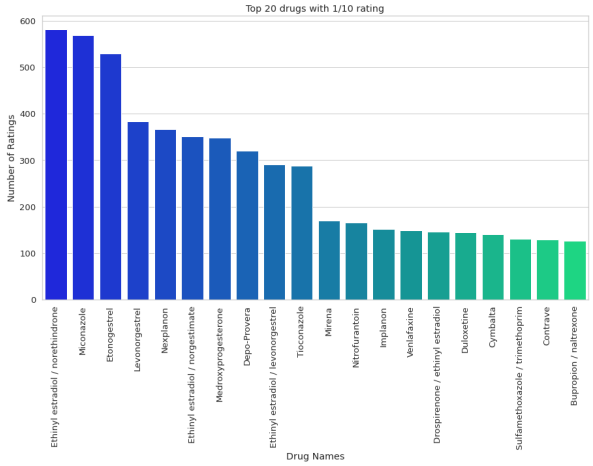

Figure 2: top 20 obat dengan rating 1/10

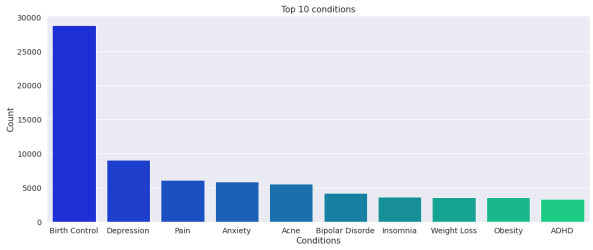

Figure 3: top 10 condition

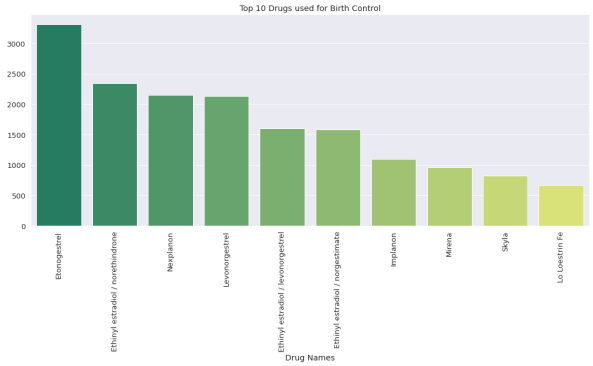

Figure 4: top 10 drugs used birth control

```
df_condition_1 = df_condition[df_condition['drug name'] == 1].reset_index()

all_list = set(df.index)

# deleting them
condition_list = []
for i,j in enumerate(df['condition']):
    for c in list(df_condition_1['condition']):
        if j == c:
            condition_list.append(i)

new_idx = all_list.difference(set(condition_list))
df = df.iloc[list(new_idx)].reset_index()
del df['index']
```

```
df.shape
```

```
(160684, 7)
```

Figure 5: code untuk menghapus nilai kondisi

setel ulang indeks untuk memastikan bahwa indeks diurutkan setelah menghapus beberapa baris data seperti pada Gambar 5.

Hapus nilai "kondisi" yang berisi , lalu setel ulang indeks untuk memastikan bahwa indeks diurutkan setelah menghapus beberapa baris data.

Menampilkan reviewcsore max dan min seperti pada Gambar7. Selanjutnya menambahkan colom MeanNormalizedScore ke dataset seperti pada Gambar 8. Kemudian mengurutkan dataset sesuai kondisi dan jenis obatnya seperti pada Gambar 9.

Langkah selanjutnya adalah memodelkannya dengan menggunakan regresi logistik. Dataset dibagi menjadi dua yaitu dataset training dan dataset testing. Codenya seperti pada Gambar 10. Code untuk regresi logistiknya seperti pada Gambar 11. Oputput yang dihasilkan pada bagian akhir berupa nilai akurasi regresi logistik sebesar 0.8410925474000376 atau 84,1 persen. Atinya model yang diperoleh cukup baik untuk memprediksi datasetnya.

```
# removing the conditions with <span> in it.

all_list = set(df.index)
span_list = []
for i,j in enumerate(df['condition']):
    if "</span>" in str(j):
        span_list.append(i)
new_idx = all_list.difference(set(span_list))
df = df.iloc[list(new_idx)].reset_index()
del df['index']
```

Figure 6: code untuk menghapus nilai kondisi dengan ¡span¿

```
# part 1---vader sentiment analyzer for c_review
analyzer = SentimentIntensityAnalyzer()
# create new col vaderReviewScore based on C-review
df['vaderReviewScore'] = df['review_clean'].apply(lambda x: analyzer.polarity_scores(x)['compound'])

# define the positive, neutral and negative
positive_num = len(df[df['vaderReviewScore'] >=0.05])
neutral_num = len(df[(df['vaderReviewScore'] >-0.05) & (df['vaderReviewScore']<0.05)])
negative_num = len(df[df['vaderReviewScore']<=-0.05])

# create new col vaderSentiment based on vaderReviewScore
df['vaderSentiment'] = df['vaderReviewScore'].map(lambda x:int(2) if x>=0.05 else int(1) if x<=-0.05 else int(0) )
df['vaderSentiment'].value_counts() # 2-pos: 99519; 1-neg: 104434; 0-neu: 11110

# label pos/neg/neu based on vaderSentiment result
df.loc[df['vaderReviewScore'] >=0.05,"vaderSentimentLabel"] ="positive"
df.loc[(df['vaderReviewScore'] >-0.05) & (df['vaderReviewScore']<0.05),"vaderSentimentLabel"]= "neutral"
df.loc[df['vaderReviewScore']<=-0.05,"vaderSentimentLabel"] = "negative"
```

```
df['vaderReviewScore'].max()
```

```
0.9935
```

```
df['vaderReviewScore'].min()
```

```
-0.9973
```

Figure 7: reviewscore max dan min

```
#Final Normalized Score combining Rating and normalVaderScore
df['meanNormalizedScore'] = (df['rating'] + df['normalVaderScore'])/2
df.head()
```

ıame	condition	review	rating	date	useful count	review_clean	vaderReviewScore	vaderSentiment	vaderSentimentLabel	normalVaderScore	meanNormalizedScore
ırtan	Left Ventricular Dysfunction	"It has no side effect, I take it in combinati...	9.0	May 20, 2012	27	no side effect take combin bystol mg fish oil	-0.2960	1	negative	3	6.0
cine	ADHD	"My son is halfway through his fourth week of ...	8.0	April 27, 2010	192	son halfway fourth week intuniv becam concern ...	0.6929	2	positive	9	8.5
ʻbrel	Birth Control	"I used to take another oral contraceptive, wh...	5.0	December 14, 2009	17	use take anoth oral contracept pill cycl happi...	0.2732	2	positive	3	4.0
Evra	Birth Control	"This is my first time using any form of birth...	8.0	November 3, 2015	10	first time use form birth control glad went pa...	0.4199	2	positive	8	8.0
		"Suboxone				suboxon					

Figure 8: dataset dengan kolom meanNormalizedScore

		meanNormalizedScore
		mean
condition	drug name	
ADHD	Adderall	6.990625
	Adderall XR	7.172897
	Adzenys XR-ODT	6.805556
	Amantadine	5.666667
	Amphetamine	6.522727
	Amphetamine / dextroamphetamine	6.941926
	Aptensio XR	5.833333
	Armodafinil	6.937500
	Atomoxetine	5.182266
	Bupropion	6.879310

Figure 9: dataset yang sudah diurutkan

```python
from sklearn.model_selection import train_test_split

x_train, x_test = train_test_split(df, test_size = 0.25, random_state = 0)
```

Figure 10: code pembagian dataset training dan testing

```python
#LogisticRegression
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression

# Create a CountVectorizer for text feature extraction
vectorizer = CountVectorizer()

# Fit the vectorizer on the training data and transform the data
X_train_vec = vectorizer.fit_transform(X_train)

# Transform the test data using the same vectorizer
X_test_vec = vectorizer.transform(X_test)

# Create a Logistic Regression classifier
logreg = LogisticRegression()

# Fit the classifier on the vectorized training data and corresponding labels
logreg.fit(X_train_vec, y_train)

# Make predictions on the test data
y_pred_logreg = logreg.predict(X_test_vec)

# Calculate the accuracy score
logreg_accuracy = accuracy_score(y_test, y_pred_logreg)
print("Log Reg: ", logreg_accuracy)
```

Figure 11: code regressi logistik