باسمه تعالی

دانشگاه صنعتی اصفهان

دانشکده برق و کامپیوتر

گزارش پروژه درس داده‌کاوی

موضوع پروژه:

پیش‌بینی اختلال خواب افراد بر اساس سبک زندگی آنها

اعضای تیم:

فاطمه جان‌نثاری (۴۰۱۱۴۳۰۳)

حوری دهش (۹۸۲۱۴۱۳)

اردیبهشت ۱۴۰۳

۱) فاز شناخت و فهم مسئله

معرفی مسئله:

کیفیت خواب یکی از محصولات اصلی سبک‌زندگی سالم است. طبق نظر دانشمندان، ما حدود یک سوم از عمرمان را در خواب می‌گذرانیم. خواب به شدت هم روی سلامت جسم و هم سلامت روان تاثیرگذار است.

اختلالات خواب از جمله بی‌خوابی از مشکلاتی‌است که امروزه بسیار شایع شده و گریبان‌گیر بسیاری از مردم است. مسئله این است که بفهمیم چه عواملی در سبک‌زندگی افراد روی کیفیت خواب و اختلال خواب آنها موثر است .

و می‌خواهیم بتوانیم با مشاهده سبک‌زندگی افراد میزان اختلال خواب آنها را بررسی کنیم.

معرفی سوال و معیار ارزیابی و داده‌ها:

سوال: چگونه با استفاده از بررسی ویژگی‌های مختلف شخص که مربوط به سبک زندگی اوست بتوانیم اختلال خواب او را پیش‌بینی کنیم؟

داده‌ها: داده‌های مورد استفاده در این پژوهش، دیتاستی شامل ۱۳ ستون و ۳۷۴ سطر بود که عنوان ستون‌های آن موارد زیر است: آی‌دی، سن، شغل، مدت زمان خواب، کیفیت خواب، سطح فعالیت فیزیکی، سطح استرس، دسته BMI، فشار خون، ضربان قلب، گام‌های روزانه، اختلال خواب

نمونه‌ای از داده‌های دیتاست:

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder
0	1	Male	27	Software Engineer	6.100000	6	42	6	Overweight	126/83	77	4200	nan
1	2	Male	28	Doctor	6.200000	6	60	8	Normal	125/80	75	10000	nan
2	3	Male	28	Doctor	6.200000	6	60	8	Normal	125/80	75	10000	nan
3	4	Male	28	Sales Representative	5.900000	4	30	8	Obese	140/90	85	3000	Sleep Apnea
4	5	Male	28	Sales Representative	5.900000	4	30	8	Obese	140/90	85	3000	Sleep Apnea

ایده حل مسئله:

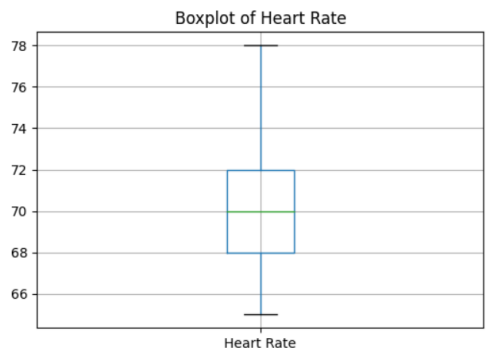
ایجاد یک مدل که بتواند روی داده‌های ما آموزش ببیند و بعد از اتمام یادگیری بتواند اختلال خواب افراد را پیش‌بینی کند.

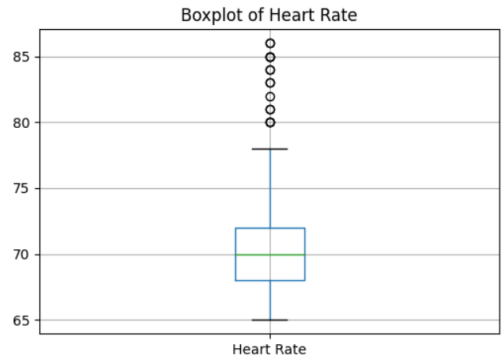
۲) آزمایشات فاز آماده‌سازی داده‌ها

- **تشخیص داده‌های پرت:**

پیدا کردن داده‌های nan: وجود داده‌های nan را در دیتاست بررسی کردیم که خوشبختانه هیچ دیتای nanای وجود نداشت.

پیدا کردن outlierها: برای این کار، نمودار box plot مربوط به هر یک از ویژگی‌های عددی را رسم کردیم و بررسی کردیم که آیا داده پرتی وجود دارد یا خیر. تنها برای ستون ضربان قلب داده پرت وجود داشت که آنها را از دیتاست حذف نمودیم.





- **تبدیل و استانداردسازی داده‌ها:**

تبدیل اطلاعات مربوط به فشار خون به دو دسته نرمال و غیرنرمال:

فشار خون ایده آل: سیستولیک (عدد بالا): کمتر از ۱۲۰، دیاستولیک (عدد پایین): کمتر از ۸۰

فشار خون طبیعی: سیستولیک (عدد بالا): در محدوده (۱۲۰ - ۱۲۹)، دیاستولیک (عدد پایین): در محدوده(84 - 80)

در غیر این صورت فشار خون بالاست. ما نیز بر همین اساس به داده‌های با وضعیت نرمال عدد ۰ و به داده‌هایی با وضعیت غیرنرمال عدد ۱ تخصیص دادیم.

```
sleep_data['Blood Pressure'] = sleep_data['Blood Pressure'].apply(lambda x:0 if x in ['120/80','126/83','125/80','128/84','129/84','117/76','118/76','115/75','125/82','122/80'] else 1)
# 0 = normal blood pressure
# 1 = abnormal blood pressure
```

تقسیم متغیرهای پیوسته به دسته‌های گسسته

```
# Binning (dividing continuous variable into discrete intervals or categories)
cleaned_data["Age"] = pd.cut(cleaned_data["Age"],2)
cleaned_data["Heart Rate"] = pd.cut(cleaned_data["Heart Rate"],4)
cleaned_data["Daily Steps"] = pd.cut(cleaned_data["Daily Steps"],4)
cleaned_data["Sleep Duration"] = pd.cut(cleaned_data["Sleep Duration"],3)
cleaned_data["Physical Activity Level"] = pd.cut(cleaned_data["Physical Activity Level"],4)
```

تبدیل داده‌های categorical به فرمت عددی

```
# convert categorical data into numerical format
LE = LabelEncoder()

categories=['Gender','Age','Occupation','Sleep Duration','Physical Activity Level','BMI Category','Heart Rate','Daily Steps','Sleep Disorder']
for label in categories:
    cleaned_data[label]=LE.fit_transform(cleaned_data[label])
```

حذف ویژگی person id

```
# remove Person ID
cleaned_data.drop(['Person ID'], axis=1, inplace=True)
```

تحلیل‌های آماری:

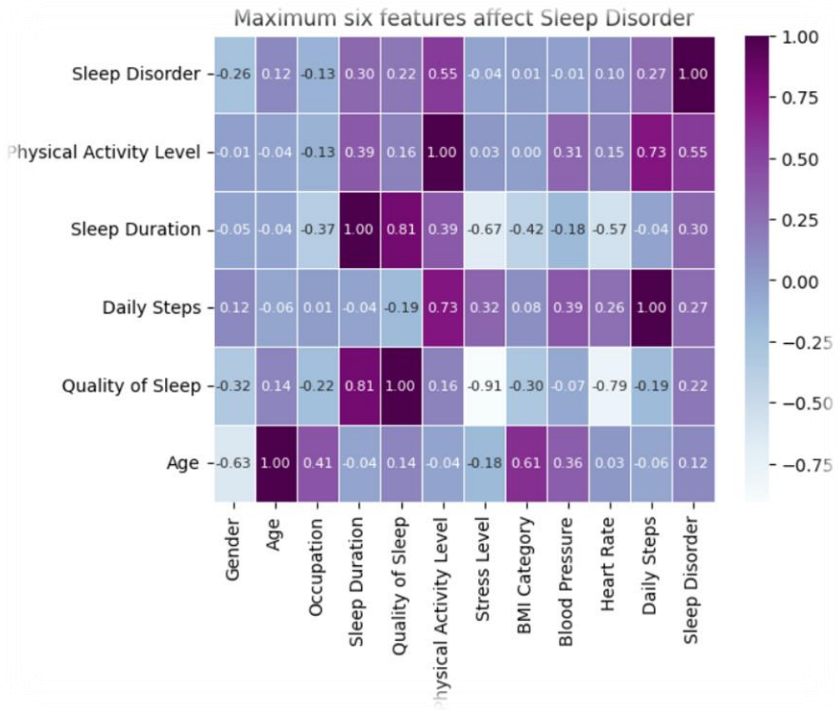
	Person ID	Age	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	Heart Rate	Daily Steps
count	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000
mean	187.500000	42.184492	7.132086	7.312834	59.171123	5.385027	70.165775	6816.844920
std	108.108742	8.673133	0.795657	1.196956	20.830804	1.774526	4.135676	1617.915679
min	1.000000	27.000000	5.800000	4.000000	30.000000	3.000000	65.000000	3000.000000
25%	94.250000	35.250000	6.400000	6.000000	45.000000	4.000000	68.000000	5600.000000
50%	187.500000	43.000000	7.200000	7.000000	60.000000	5.000000	70.000000	7000.000000
75%	280.750000	50.000000	7.800000	8.000000	75.000000	7.000000	72.000000	8000.000000
max	374.000000	59.000000	8.500000	9.000000	90.000000	8.000000	86.000000	10000.000000

	Gender	Occupation	BMI Category	Blood Pressure	Sleep Disorder
count	374	374	374	374	155
unique	2	11	4	25	2
top	Male	Nurse	Normal	130/85	Sleep Apnea
freq	189	73	195	99	78

۳) آزمایشات فاز تحلیل اکتشافی داده‌ها

بررسی و تحلیل روابط تک‌متغیره بین متغیرهای پیشین و متغیر هدف: در این مرحله ارتباط هر یک از متغیرهای پیشین را با متغیر هدف با استفاده از رسم نمودار مناسب به دست آوردیم که نمودارها و اطلاعات جزئی مربوط به آنها در کد و در اسلایدها موجود است.

بررسی روابط چندمتغیره بین متغیرها: برای بررسی ارتباط کل داده‌ها با هم از ماتریس correlation استفاده کردیم و سپس پنج ویژگی‌ای که بیشترین تاثیر را روی متغیر هدف داشته‌اند را نمایش دادیم:



۴) آزمایشات فاز پیش مدل

چارچوب تقسیم داده‌ها به دسته آموزش و تست: برای تقسیم داده‌ها به دو دسته آموزش و تست از روش cross-validation استفاده کردیم.

روش انجام cross-validation: به این نحو بود که ابتدا k را ۱۰ در نظر گرفتیم سپس ۱۰ فولد ایجاد کردیم که در هر کدام آنها ۳۲۳ رکورد برای داده آموزشی و ۳۶ رکورد برای داده تست در نظر گرفته شد.

```
Fold:1, Train set: 323, Test set:36
Fold:2, Train set: 323, Test set:36
Fold:3, Train set: 323, Test set:36
Fold:4, Train set: 323, Test set:36
Fold:5, Train set: 323, Test set:36
Fold:6, Train set: 323, Test set:36
Fold:7, Train set: 323, Test set:36
Fold:8, Train set: 323, Test set:36
Fold:9, Train set: 323, Test set:36
Fold:10, Train set: 324, Test set:35
```

۵) آزمایشات فاز مدل سازی

انتخاب و پیاده‌سازی الگوریتم‌های لازم

ما با بررسی کدهای مشابه و مدل‌های مختلف که برای این نوع کلاس‌بندی و پیش‌بینی استفاده شده بود، چند مدل را انتخاب کردیم اما بعضی از آنها دقت کافی را نداشتند و خیلی کارآمد نبودند به همین دلیل آنها را حذف کردیم و مدل‌های نهایی ما موارد زیر بودند: Random Forest، Extra Tree، Gradient Boosting

اطمینان از عملکرد بهتر نسبت به موارد قبلی: ما برای انتخاب مدل‌ها هر یک را اجرا می‌کردیم و دقت را بررسی می‌کردیم. به همین روش بود که بعضی از مدل‌ها که عملکرد خوبی نداشتند را حذف کردیم. روش اجرای الگوریتم‌ها نیز به این صورت بود که ابتدا با استفاده از کراس ولیدیشن ۱۰ فولد ایجاد می‌کردیم بعد روی هر کدام از این فولدها یکبار ترین می‌کردیم و بعد روی هر فولدی که دقت بیشتری داشت، آن فولد را انتخاب می‌کردیم و بعد آن فولد را با استفاده از تابع grid search تیون می‌کردیم که بهترین پارامترها را برای آن پیدا کنیم. و بعد که بهترین پارامترها پیدا شد یکبار دیگر روی آن فولد آموزش را انجام می‌دادیم تا به بالاترین دقت برسیم.

۶) آزمایشات فاز ارزیابی

معرفی مجموعه معیارهای ارزیابی و محاسبه آن و تفسیر نتایج

ما از معیار accuracy استفاده کردیم. همچنین از معیارهای cross val score، precision score، recall score و f1 score

برآورد خطا

نتایج به دست آمده برای هر مدل:

	Model	Best fold	Train_accuracy	Test_accuracy	Cross Val Score	Difference Train & Test	precision_score	recall_score	f1_score	Description
0	DT	7	0.925816	0.918919	0.898578	0.006897	0.869048	0.924603	0.893200	
1	DT2	3	0.925595	0.921053	0.903841	0.004543	0.922078	0.891775	0.903175	

	Model	Best fold	Train_accuracy	Test_accuracy	Cross Val Score	Difference Train & Test	precision_score	recall_score	f1_score	Description
0	RF	3	0.931548	0.921053	0.914651	0.010495	0.922078	0.891775	0.903175	
1	RF2	10	0.922849	0.918919	0.898506	0.003930	0.879630	0.922078	0.897852	

	Model	Best fold	Train_accuracy	Test_accuracy	Cross Val Score	Difference Train & Test	precision_score	recall_score	f1_score	Description
0	GB	1	0.931548	0.921053	0.908962	0.010495	0.933333	0.909091	0.912281	
1	GB2	9	0.925816	0.918919	0.919915	0.006897	0.899471	0.899471	0.899471	

	Model	Best fold	Train_accuracy	Test_accuracy	Cross Val Score	Difference Train & Test	precision_score	recall_score	f1_score	Description
0	ET	6	0.928783	0.918919	0.908962	0.009864	0.875000	0.895563	0.883915	
1	ET2	7	0.925816	0.918919	0.911664	0.006897	0.888889	0.916667	0.885714	

مشخص کردن بهترین مدل از بین مدل‌های اجرا شده به همراه پارامترهای تعیین شده

	Model	Train_accuracy	Test_accuracy	precision_score	recall_score	f1_score	Description
0	DT	0.925595	0.921053	0.922078	0.891775	0.903175	
1	RF	0.922849	0.918919	0.879630	0.922078	0.897852	
2	GB	0.925816	0.918919	0.899471	0.899471	0.899471	
3	ET	0.925816	0.918919	0.888889	0.916667	0.885714	

بهترین مدل برای دیتاست ما درخت تصمیم بود.

تحلیل نقاط قوت و ضعف کار انجام شده و پیشنهادات برای بهبود آینده

چالش‌ها:

۱. نتیجه خروجی نامناسب بعد از tune کردن: بعد از تیون کردن بعضی مدل‌ها هیچ بهبودی در عملکرد آنها حاصل نمی‌شد و دقت آنها افزایش پیدا نمی‌کرد.

راه حل:

تغییر هایپرپارامترهایGrid Search

گذاشتن مقدار دیفالت هایپرپارامترها

تغییر روش تقسیم‌بندی داده از holdout بهCross validation (در ابتدا برای تقسیم داده‌ها از روش holdout استفاده می‌کردیم ولی پس از آن مطلع شدیم کراس ولیدیشن مزیت‌های خیلی بیشتری دارد)

۲. در روش Cross validation کدام fold برای tune کردن در نظر گرفته شود.

راه حل:

در اینجا به علت زمان‌بر بودن tune کردن ما بهترین fold را از بین تمامیfold ها در نظر گرفتیم.

پیشنهادها برای بهبود:

بالانس کردن داده‌ها و استفاده از نسبت‌های برابر از هر کلاس در داده‌ها و تیون کردن مجدد و بررسی نتایج

استفاده از روش‌های دیگر برای پیدا کردن پارامترها به جای grid search مثلا naïve base یا روش‌های سرچ دیگر

استفاده از feature engineering

استفاده از ensemble learning