

بسم الله الرحمن الرحيم

مستند پروژه نهایی

عنوان: مشاور تغذیه هوشمند

استاد: دکتر دادگر

دانشجو: سبحان شهنواز

دانشگاه ملی مهارت فارس، شهید باهنر شیراز

سال تحصیلی ۴۰۴-۴۰۳، نیم سال دوم

فهرست مطالب:

۱.مقدمه و معرفی.....	۳
۲.ابزار توسعه.....	۳
۳.جمع آوری داده.....	۴
۴.آماده سازی داده و تست.....	۴
۵.مدل تخمین کالری.....	۵
۶.توسعه اولیه رابط گرافیکی.....	۵
۷.کارگذاری مدل تخمین کالری در رابط گرافیکی.....	۶
۸.مهندسی ویژگی در مدل تخمین کالری.....	۷
۹.علل عدم اکتفای مدل تخمین کالری.....	۷
۱۰.مقاله های بررسی شده.....	۷
۱۱.جمع آوری داده(۲).....	۸
۱۲.آماده سازی داده و تست(۲).....	۸
۱۳.بررسی علل غیرقابل قبول بودن داده ها.....	۹
۱۴.مدل پیشنهاد تمرین.....	۱۰
۱۵.مدل پیشنهاد تغذیه.....	۱۱
۱۶.بهبود رابط گرافیکی.....	۱۳
۱۷.کارگذاری مدل ها در رابط گرافیکی.....	۱۴
۱۸.ترمیم نهایی رابط گرافیکی.....	۱۵
۱۹.چالش ها.....	۱۷
۲۰.منابع.....	۱۸
۲۱.تقدیر و تشکر.....	۱۸

مقدمه و معرفی:

ایده های اولیه بسیاری برای موضوع اصلی پروژه نهایی در دسترس بود. برخی در نظر خود بنده جالب نبود و برخی نیز توسط استاد مناسب دیده نشد. در نهایت تصمیم نهایی به پیاده سازی یه مشاور تغذیه هوشمند تحت فضای وب ختم شد. در دیدگاه اولیه چالش های زیادی پیش رو نبود و در بررسی اولیه منابع برای هوشمندی سازی نیز یافت شد. در ادامه با آماده سازی داده و اجرای الگوریتم های توسعه مدل، تعداد زیادی از این داده ها عملکرد ضعیف داشتند و یک مورد نیز به حالت **Overfitt** ختم شد. هرچند یک مدل مناسب برای تخمین کالری حاصل شد، اما چالش اصلی یافتن داده مناسب برای سایر قابلیت ها بود.

در نهایت یک سیستم با سه مدل توسعه داده شد که بر اساس معیار های سلامت فرد خدمات لازم را ارائه می دهد. در این مستند به صورت جزئی روند توسعه این سیستم بررسی میشود. ابزارها، علت انتخاب رویکردها و راهبردها، علل شکست در داده کاوی، سایر چالش های توسعه و در نهایت علل به وجود آمدن حالت فعلی شرح داده خواهد شد.

ابزار توسعه:

• هوشمند سازی: پایتون

برای کار با داده ها و توسعه مدل ها از کتابخانه های پایتونی ذیل استفاده شده است:

NumPy: انجام محاسبات عددی.

Pandas: مدیریت داده ها و دیتافریم ها.

Matplotlib: رسم نمودارهای تحلیل داده و نتایج.

Scikit-learn: انجام پیش پردازش، مدل سازی و ارزیابی مدل ها. به کار رفته در یک مدل.

XGBoost: مدل سازی داده، با دقت و سرعت بالا. به کار رفته شده در دو مدل.

CatBoost: مدل سازی داده های دسته ای با دقت بالا.

LightGBM: مدل سازی سریع و سبک در حجم داده بالا.

PyTorch TabNet: به کار رفته در آزمایش یادگیری عمیق.

Imblearn: نمونه گیری برای طبقه بندی داده های نامتوازن.

Joblib: ذخیره مدل ها و بارگذاری در جنگو.

• توسعه رابط گرافیکی : جنگو

رابط گرافیکی اپلیکیشن مشاوره که به صورت تحت وب میباشد با چارچوب جنگو که برپایه پایتون است توسعه داده شده است. برای قالب ها از HTML، CSS و برای پویایی نیز چنخطی JS به کار رفته است.

جمع آوری داده (۱):

داده های مدل تخمین کالری از USDA جمع آوری شد. این داده شامل ۱۴۰ رکورد میباشد. جدول زیر شامل اطلاعات ویژگی های کلیدی این داده ها میباشد.

ویژگی	توضیحات
جنسیت	داده دسته ای، مقادیر زن و مرد
سن	داده عددی، بازه ۱۲ تا ۶۶
قد	داده عددی، واحد سانتی متر، بازه ۱۳۹ تا ۲۰۲
وزن	داده عددی، واحد کیلوگرم، بازه ۳۶ تا ۱۱۸
سطح فعالیت	داده دسته ای، ۴ دسته از غیرفعال تا بیش فعال
شاخص توده بدن	داده اعشاری، بازه ۱۵ تا ۳۴/۵
نیاز کالری روزانه	ویژگی هدف، عددی، بازه ۱۷۰۰ تا ۴۲۰۰

آماده سازی داده و تست (۱):

با استفاده از ابزار کاربردی برای پیش پردازش و دیتافریم؛ داده های اولیه آماده شد تا با الگوریتم های مدل سازی تست شود. در اولین تغییر حالت ویژگی جنسیت به عدد ۰ برای زن و ۱ برای مرد تغییر یافت. همچنین سطح فعالیت نیز طبق جدول زیر تغییر یافت.

مقدار	غیرفعال	کمی فعال	فعال	بسیار فعال
معادل	۱	۱/۴	۱/۶	۱/۹

دلیل استفاده از این اعداد معادل این بود که طبق بررسی ها پایگاه USDA برای محاسبات خود این اعداد معادل را در نظر میگیرد. در داده های مدل تخمین کالری از ویژگی های جنسیت، سن، قد، وزن، میزان فعالیت و شاخص توده بدنی استفاده میشود. هدف در این میزان کالری است تا به نسبت آن سایر درشت مغذی ها نیز پیشنهاد شود. داده های جمع آوری شده برای این مدل به صورت پیشفرض برای رگرسیون مناسب بود.

مدل تخمین کالری:

در این جا از رگرسیون خطی و رگرسیون جنگل تصادفی استفاده شد. دلیل کلی استفاده از رگرسیون، نوع ویژگی هدف میباشد که عددی بزرگ است. در جدول زیر دو الگوریتم مقایسه میشود.

معیار	رگرسیون خطی چندگانه	رگرسیون جنگل تصادفی
میانگین خطای مطلق	۵۷/۹۴	۱۰۷/۲۲
R2	۹۸	۹۴

البته در رگرسیون خطی با استفاده از StandardScaler ویژگی های عددی برای پیش بینی بهینه می شوند. از این دو الگوریتم کاملاً ساده و روی حالت پیش فرض استفاده شده است. به این معنی که پارامتر یا مرحله ای اضافه نشده است.

با توجه به نتایجی که در حالت رگرسیون خطی به دست آمد، استفاده از این مدل را در برنامه مناسب است. نتایج زیر نشان میدهد که الگوریتم با توجه به اینکه اعدادی در حدود بازه ۲۰۰۰ تا ۴۰۰۰ را حدس میزند. عملکرد بالایی دارد. همچنین با چندین نمونه تست شد که میانگین خطای حدودی ۹۰ کالری برای جنسیت مذکر و خطای حدود ۶۰ کالری داشت. صد البته که در گام های آتی برای بهبود این مقادیر تلاش خواهد شد.

MultipleLinearRegression Evaluation:
Intercept: 2616.6698213613745
Mean squared error: 4954.59
Mean absolute error: 57.94
Coefficient of determination (R^2): 0.98748

توسعه اولیه رابط گرافیکی:

در این مرحله کار با چارچوب جنگو آغاز شد. صفحات اولیه طراحی شد، توابع آن توسعه داده شد و مسیرها تنظیم شد. در طراحی ها سعی شد رنگ های زنده و روشن استفاده شود و رنگ های مختلف را به کار گرفته شود که تصاویر آن را در گزارش های زمان بندی شده مشاهده کردید. این مورد به این هدف بود که در نهایت یک تم رنگی نهایی انتخاب شود و به تمامی صفحات اعمال شود.

فرم اولیه ای که در زیر مشاهده میکنید به صورت آزمایشی برای تست مدل در چارچوب جنگو بود و در مراحل بعدی منظم تر خواهد بود.

هدف از توسعه رابط گرافیکی در این مرحله بررسی یکپارچه سازی مدل هوشمند و پلتفرم وب بود.

کارگذاری مدل تخمین کالری در رابط گرافیکی:

در این مرحله در قسمت کنترل توابع یا همان View جنگو مدل بارگذاری شد تا برای پیشبینی با مقادیر استفاده شده به کار برده شود. و چندین نتیجه با آن بررسی شد که باعث شد. با توجه به نتایج نیاز به بهبود مدل بیشتر به چشم آمد. کد های کلیدی این مورد را جدا شده و اینجا قرار داده شده است:

```
1 def NutritionPredict(request):
2     MODEL_DIR = os.path.join(os.path.dirname(__file__), "models")
3     ECDNMODEL_PATH = os.path.join(MODEL_DIR, "MlrEDCN.pkl")
4     ECDNmodel = joblib.load(ECDNMODEL_PATH)
5     ECDNscaled_input = ECDNscaler.transform([[Gender, Age, Height, Weight, Activitylevel]])
6     ECDNprediction = ECDNmodel.predict(ECDNscaled_input)
7     ECDN = ECDNprediction[0]
8     template = loader.get_template('NutritionPredict.html')
9     context = {
10         'ECDN': ECDN,
11     }
12     return HttpResponse(template.render(context, request))
13
```

مهندسی ویژگی در مدل تخمین کالری:

در این مرحله با مهندسی ویژگی ها عملکرد مدل بهبود یافت. این ویژگی ها را در اینجا بررسی خواهد شد.

- مساحت بدن، به جای شاخص توده بدنی: برای پیش بینی نیاز به کالری بدن این مورد نسبت به شاخص توده بدنی عملکرد بهتری دارد و با فرمول $\sqrt{3600 / (\text{قد} * \text{وزن})}$ محاسبه میشود.
- ضرب سن در وزن: وزن و سن افراد در میزان کالری ای که نیاز دارند تاثیر مستقیم دارد. با ترکیب این دو میتوان نتیجه را بهبود داد.
- فعالیت فیزیکی \times وزن: هرچه وزن بیشتر باشد، فعالیت فیزیکی نیاز به کالری را بیشتر افزایش میدهد.
- فعالیت فیزیکی \times قد: در این مورد نیز قد بیشتر — میزان فعالیت باعث افزایش نیاز به کالری میشود.
- فعالیت فیزیکی \times مساحت بدن: ترکیبی از دو ویژگی بالا بر پایه فرمول ذکر شده.
- فعالیت فیزیکی \times جنسیت: دلیل بکارگیری این ویژگی تفاوت میزان نیاز کالری در میزان فعالیت بالا توسط مردان میباشد.
- فعالیت فیزیکی \times مساحت بدن \times جنسیت: یک ویژگی ترکیبی از چندین مورد ذکر شده که با لیبیل هدف ارتباط مستقیم دارد.

نتایج عملکرد مدل پس از انجام این مرحله در زیر قرار داده شده است:

MultipleLinearRegression Evaluation:
Intercept: 2618.199544557975
Mean squared error: 2063.96
Mean absolute error: 33.31
Coefficient of determination (R^2): 0.99479

علل عدم اکتفای مدل تخمین کالری:

لیبل هدف این مدل را می توان بدون به کارگیری هوش مصنوعی با استفاده از فرمول های معینی به دست آورد. در نتیجه با اکتفا به این مدل مطلبی برای ارائه وجود ندارد. هرچند برخی موارد مانند فیبر مورد نیاز بدن در منابع جهانی تغذیه همچنان با هوش مصنوعی تخمین زده میشود، که این پروژه این ویژگی را هم داراست. اما افزایش مدل های این پروژه، دیدگاه مفیدتر و کاربردی تری برای گسترش اپلیکیشن بنظر می آید. در نتیجه به همین یک مدل اکتفا نشد و برای توسعه مدل های دیگر به جمع آوری و تست داده های جدید پرداخته شد.

مقاله های بررسی شده:

به توصیه دکتر دادگر در این فاز به جستجو در مقالات علمی برای یافتن داده های مناسب پرداخته شد. مقالات متعددی در ارتباط با توسعه یک سیستم هوشمند پیشنهاد تغذیه وجود داشت. در همان مرحله اول گزینه های قابل بررسی کاملاً محدود شد، زیرا بیش از نیمی از مقالات به صورت رایگان در دسترس نبود. تعدادی دیگر از مقاله ها هم بدون رفرنس به داده ها فقط اسلوب و رویکردهای پیاده سازی را شرح داده بودند. مطالعه روزنامه وار این موارد اندکی مسیر پیش رو را روشن تر کرد تا این دانش که

دقیقا یک مجموعه داده مناسب باید چگونه باشد، حاصل شد. در نهایت حدود ۶ مقاله با لینک داده ها یا لینک به رپازیتوری وجود داشت. پس از بررسی رپازیتوری ها مشاهده شد سیستم های توسعه داده شده هیچ معیاری برای بررسی صحت دسته بندی ها ندارند. در هر صورت باید داده ها را برای هدف مورد نیاز پروژه آماده می شد. و سپس روی آنها عملیات تست را انجام می شد.

جمع آوری داده (۲):

پس از بررسی مقالات تعدادی داده ذکر شده در آنها به دست آمد و پس از آن با رویکردی جدید در منابع مختلف برای یافتن داده به جست و جو پرداخته شد. سه مورد از داده های جدید به دست آمده دسته بندی شد و لیبل ها و اهداف آن ها بررسی شد. در حالت نوشتاری در مستند نوشتن این عمل چند خط را دربرمیگیرد اما این موضوع روزها درگیری در پی داشت. ۶ مورد از داده ها که همچنان موجود است را در یک مخزن در گیت هاب قرار داده شد، و لینک آن با دکتر دادگر به اشتراک گذاشته شد. یکی از داده ها برای آموزش دو مدل پیش رو به کار رفته است. منبع این داده سایت کگل می باشد. این داده شامل ۱۴۵۰۰ رکورد است. جدول زیر شامل اطلاعات ویژگی های کلیدی این داده ها میباشد.

ویژگی	توضیحات	ویژگی	توضیحات
جنسیت	دسته ای، مرد و زن	دیابت	بولین
سن	۱۸-۶۳	شاخص توده بدنی	اعشاری
قد	اعشاری، واحد متر	حالت بدن	دسته ای، کمبود یا اضافه وزن
وزن	اعشاری، واحد کیلوگرم	تمرین	دسته ای، نوع تمرین پیشنهادی
فشارخون	بولین	مواد غذایی	دسته ای، نوع مواد غذایی پیشنهادی

آماده سازی داده و تست (۲):

در این جا داده ها به دو دسته قابل قبول و غیرقابل قبول (نتایج غیرمنطقی) تقسیم میشود که دومین داده قابل قبول (اولی همان تخمین کالری است) دارای دو لیبل تمرین، تغذیه بود و در همان اوایل تست نتایج خوبی داشت. اما داده های دیگری که وجود دارند را با رویکرد مختلف، ویژگی های مختلف، رمزگذاری ویژگی ها با روش های متفاوت، الگوریتم ها و کتابخانه های متفاوت بررسی شده اند و نتیجه مناسبی نداشته اند. برخی از آنها را با ساده ترین الگوریتم ها تا الگوریتم های پیشرفته دسته بندی بررسی شد اما هیچ کدام نتایج قابل قبولی نداشتند.

در انتها با داده های که دو هدف در آن وجود داشت و نتایج قابل قبولی را به همراه داشت دو مدل ذخیره شد. در زیر پراکندگی هدف اول یعنی تمرین را پس از تمیز شدن داده ها، مشاهده میشود:

```
Encoded Exercises value counts:
Exercises_Label
0    3423
1    3137
2    2799
3    1390
4    1389
Name: count, dtype: int64
```


این سری داده لیبل های متنی داشت که عمل مپ روی آنها انجام شد و لیبل تغذیه نیز ۱۱ مورد یکتا داشت که با همان عمل مپینگ این داده ها برای مدل آماده شد.

دو مدل XGBoost با دو هدف متفاوت برای دسته بندی داده ها توسعه داده شد که کد آن در همان مخزن دیتاست ها موجود است.

نتیجه عملکرد مدل برای حدس لیبل تمرین:

```
XGBoost Results:
Accuracy: 0.9979406919275123
Classification Report:
              precision    recall  f1-score   support

     0              1.00        1.00        1.00         685
     1              1.00        1.00        1.00         627
     2              1.00        0.99        1.00         560
     3              0.99        1.00        1.00         278
     4              0.99        1.00        1.00         278

 accuracy              1.00        1.00        1.00        2428
 macro avg              1.00        1.00        1.00        2428
 weighted avg           1.00        1.00        1.00        2428
```

این نتایج بنظر ساده بدست آمده اند. برای این که دلیل این نتیجه را متوجه شوید بخشی هایی از عمل تمیزکاری در ادامه بررسی خواهد شد. البته که کد آن هم در مخزن موجود است.

- نبود مقدار خالی در بررسی اولیه دیتا
- بررسی صحت شاخص توده بدنی: یک مورد در این مرحله یافت شد که به صورت خودکار تصحیح شد.
- حذف مقادیر تکراری: حدود ۴۰۰۰ مقدار با شناسه یکسان وجود داشت که تمامی آنها به صورت خودکار حذف شدند.
- بررسی Outliners: مقادیر غیرعادی یافت شد اما به دلیل اینکه در دنیای واقعی مشابه آن وجود دارد (مثال: قد دو متری) نگه داشته شدند.
- تغییر رمزگذاری: جنسیت (مرد = ۱، زن = ۰)، فشارخون (۱ = دارد، ۰ = ندارد)، دیابت (۱ = دارد، ۰ = ندارد). و تغییر رمزگذاری سطح بدن به عددی (این مورد با حالت چند سطونی هم بررسی شد ولی تاثیری در نتیجه الگوریتم نداشت).
- مپ کردن دو ستون تارگت به حالت عددی با جداسازی مقدار متنی و نگه داری عدد و متن برابر در فایلی دیگر.

بررسی علل غیر قابل قبول بودن داده ها:

در اینجا پرداخته می شود به داده های غیر قابل قبول و دلایل نتایج ضعیف و کنار گذاشته شدن آن ها بررسی میشود.

- نبود ارتباط میان فیچرها و تارگت یا شناسایی نشدن آن توسط الگوریتم ها
- توزیع نامتوازن داده ها در لیبل هدف
- عدم شناسایی الگوهای دسته بندی توسط الگوریتم های مختلف
- مقدار فیچر های زیاد
- تعداد بالای دسته های لیبل

- تاثیر نداشتن الگوریتم های پردازش کلاس های نامتوازن در نتایج نهایی
- نتایج ضعیف با رمزگذاری های متفاوت
- Feature Correlation: خیلی کم برای تمامی فیچر یا مقدار خیلی زیاد برای یک فیچر
- مناسب نبودن داده ها برای طبقه بندی (و حتی رگرسیون)

مدل پیشنهاد تمرین:

این مدل با استفاده از هفت ویژگی ساده و در دسترس برای کاربر پنج حالت ورزشی مختلف را پیشنهاد میکند. این ویژگی ها عبارت اند از سن، جنسیت، قد، وزن، ابتلا به بیماری های فشار خون و دیابت. برای این مدل از الگوریتم XGBoost استفاده شد. این الگوریتم ساده، سبک، سریع، و دقیق است. هنگام جداسازی داده های آموزش و تست، با پارامتر `stratify=y` اطمینان حاصل میشود که توزیع برچسب در هر دو مجموعه مشابه باشد. برای مدیریت دسته های نامتوازن وزن هر کلاس طوری محاسبه می شود که کلاس های با تعداد کمتر وزن بیشتری داشته باشند. زمان تعریف مدل پارامترهای `eval_metric='mlogloss'` و `random_state=42` استفاده شده است. Mlogloss معیاری متداول برای مسائل طبقه بندی چندکلاسه است، این متریک به شما کمک می کند تا مدل را براساس عملکرد دقیق آن در پیش بینی کلاس ها ارزیابی کنید. پارامتر بعدی باعث می شود که نتایج مدل قابل تکرار باشد؛ به عبارت دیگر، هر زمان که کد اجرا شود، نتایج مشابهی تولید خواهد شد. هنگام آموزش نیز پارامتری به نام `sample_weight` به کار رفته که از همان وزن های محاسبه شده استفاده می شود تا مشکل عدم توازن کلاس ها کمرنگ شود. عملکرد این مدل:

```
XGBoost Results:
Accuracy: 0.9979406919275123
Classification Report:
              precision    recall  f1-score   support
0               1.00         1.00         1.00         685
1               1.00         1.00         1.00         627
2               1.00         0.99         1.00         560
3               0.99         1.00         1.00         278
4               0.99         1.00         1.00         278
accuracy                1.00
macro avg               1.00         1.00         1.00         2428
weighted avg            1.00         1.00         1.00         2428
```

```
XGBoost Feature Importance:
      Feature  Importance
2    Diabetes    0.449656
1  Hypertension    0.290078
3      Level     0.256876
6     Weight     0.001310
4       Age      0.001020
5     Height     0.000886
0        Sex     0.000173
```

مدل پیشنهاد تغذیه:

کدهای این مدل نیز در مخزن گیت هاب در قسمت داده های قابل قبول وجود دارد. برای بهبود عملکرد این مدل یک فیچر بنام میزان ناسالمی با استفاده از فشار خون، دیابت، دسته وضعیت بدنی (با توجه به بازه های شاخص توده بدنی) ساخته شده است. برای این مدل نیز الگوریتم XGBoost با روندی مشابه مدل قبل به کار گرفته شده است. همان پارامتر ها، وزن کلاس ها و روش ها در همینجا نیز به کار رفته است.

در اینجا نیز مقادیر اجباری برای کاربر همان قبلی ها هستند، سایر ویژگی های که در فرم هستند اگر وارد نشوند، از همین موارد اجباری قابل به دست آمدن هستند. عملکرد این مدل:

XGBoost Results for Diet_Label:

Accuracy: 0.943986820428336

Classification Report:

	precision	recall	f1-score	support
0	0.96	0.98	0.97	405
1	0.67	0.68	0.67	80
2	0.68	0.74	0.71	80
3	0.21	0.60	0.32	5
4	0.99	0.96	0.97	831
5	0.98	0.97	0.98	141
6	0.96	0.95	0.95	280
7	0.95	0.96	0.96	139
8	0.95	0.97	0.96	139
9	0.96	0.96	0.96	188
10	0.94	0.93	0.94	140
accuracy			0.94	2428
macro avg	0.84	0.88	0.85	2428
weighted avg	0.95	0.94	0.95	2428

XGBoost Feature Importance:

	Feature	Importance
3	Level	0.346584
8	Exercises_Label	0.213594
9	UnhealthyScore	0.152726
2	Diabetes	0.150290
1	Hypertension	0.048690

0	Sex	0.027330
6	Height	0.020288
4	Age	0.016686
7	Weight	0.012499
5	BMI	0.011313

Feature Correlation:

	Sex	Hypertension	Diabetes	Level	Age \
Sex	1.000000	0.066102	0.065967	-0.007114	0.011540
Hypertension	0.066102	1.000000	0.074669	-0.008148	0.022357
Diabetes	0.065967	0.074669	1.000000	-0.008224	0.022464
Level	-0.007114	-0.008148	-0.008224	1.000000	-0.006764
Age	0.011540	0.022357	0.022464	-0.006764	1.000000
BMI	-0.017126	-0.012915	-0.013009	0.933748	-0.017158
Height	-0.050633	-0.039877	-0.039626	0.004397	0.018747
Weight	-0.039907	-0.031661	-0.031661	0.903806	-0.005672
Exercises_Label	0.063990	0.159759	0.859920	0.054123	0.022654
UnhealthyScore	0.049179	0.769656	0.379510	0.298870	0.015085

	BMI	Height	Weight	Exercises_Label	UnhealthyScore
Sex	-0.017126	-0.050633	-0.039907	0.063990	0.049179
Hypertension	-0.012915	-0.039877	-0.031661	0.159759	0.769656
Diabetes	-0.013009	-0.039626	-0.031661	0.859920	0.379510
Level	0.933748	0.004397	0.903806	0.054123	0.298870
Age	-0.017158	0.018747	-0.005672	0.022654	0.015085
BMI	1.000000	-0.124808	0.899919	0.047774	0.275581
Height	-0.124808	1.000000	0.300774	-0.039411	-0.029785
Weight	0.899919	0.300774	1.000000	0.026416	0.251771
Exercises_Label	0.047774	-0.039411	0.026416	1.000000	0.565871
UnhealthyScore	0.275581	-0.029785	0.251771	0.565871	1.000000

بهبود رابط گرافیکی:

تم رنگی یکسانی در تمامی صفحات اعمال شد، حالت گزینه های صفحه اصلی مرتب تر شد و حالت لیست را به آن اعمال شد:

خوش آمدید



در چه موردی نیاز به کمک دارید؟

بررسی نیاز درشت مغذی ها

اطلاع از نیاز روزانه کالری؛ پروتئین کربوهیدرات و ... ؛ جهت ثبات وضعیت فعلی بدن.

پیشنهاد ورزش های ساده

پیشنهاد تمرین برای شما با توجه به وضعیت سلامتی بدنتان.

پیشنهاد مواد غذایی

مناسب برای افرادی که به دنبال بهینه سازی رژیم غذایی برای اهداف ورزشی یا سلامتی خاص هستند.

ظاهر فرم ها نیز بهینه سازی شد تا حالت مرتب تری داشته باشد:

فرم درشت مغذی ها

90% — + Reset



اطلاعات خود را وارد کنید.

قد:

سن:

جنسیت:

☒ مرد

☐ زن

وزن:

فعالیت فیزیکی :

☐ غیرفعال

☐ کمی فعال

☐ فعال

☐ بسیار فعال

ثبت

90%

+

Reset

فرم ثبت اطلاعات

لطفاً برای پاسخ درست اطلاعات خود را با دقت وارد کنید!

جنسیت:

☒ مرد
 ☐ زن

قد:

وزن:

سن:

وضعیت بدنی:

☐ دارای کسر وزن
☐ عادی
☐ اضافه وزن
☐ فربه
☐ اطلاعی ندارم!

بیماری ها:

☒ دیابت
☐ فشارخون

ثبت

همچنین صفحات نتایجی که نیاز به تغییرات داشت، تغییر داده شد، برای قسمت تغذیه نیز صفحه فرم و نتیجه طراحی اولیه شد که در مراحل بعدی کامل خواهد شد. در اینجا تصویری از صفحه نتایج فرم تمرین را مشاهده میشود:

نتایج

To exit full screen, press and hold Esc

جدول اول اطلاعات شماسست و جدول دوم پیش بینی مدل برای شما

ردیف	ماده مغذی	نیاز پیش بینی شده
1	شماره ست	2
2	تمرین ها	اسکات،ددلیفت،پرس سینه،پرس بالای سر
3	تمرین ها به انگلیسی	Squats, deadlifts, bench presses, and overhead presses

شاخصه	مقدار
جنسیت	مذکر
سن	27
قد	196
وزن	88
وضعیت بدن	اضافه وزن ❌ (عادی)
شاخص توده بدنی	22.9
بیماری ها:	بدون بیماری.

کارگذاری مدل ها در رابط گرافیکی:

مدل های تمرین و تغذیه که ذخیره شده بود، در پروژه دوباره بارگذاری شد تا برای حدس از آنها استفاده شود. بخشی از کد های کلیدی در توابع کنترلی جنگو در زیر قرار داده میشود.

```

○○○
1 def StandExercisePredictor(Profile):
2     MODEL_DIR = os.path.join(os.path.dirname(__file__), "models")
3     XGBMODEL = os.path.join(MODEL_DIR, "XGBExercise.pkl")
4
5     if Profile.count:
6         ExerciseModel = joblib.load(XGBMODEL)
7         ExerciseSet = ExerciseModel.predict(Profile)
8         ExLabel = int(ExerciseSet[0] if isinstance(ExerciseSet, tuple) else ExerciseSet)
9         return ExLabel

```

کد های تابع تغذیه نیز مشابه است. ضمن در قسمت پایگاه داده مپ ها تعریف شده تا دسترسی سریع به متون آن قابل اجرا باشد.

```

○○○
1 class ExerciseMaps(models.Model):
2     Label = models.IntegerField(default=0)
3     TextEnglish = models.CharField(max_length=255)
4     TextPersian = models.CharField(max_length=255)
5
6 class DietMaps(models.Model):
7     Label = models.IntegerField(default=0)
8     TextEnglish = models.CharField(max_length=511)
9     Veggies = models.CharField(max_length=255)
10    Protein = models.CharField(max_length=255)
11    Juice = models.CharField(max_length=255)

```

ترمیم نهایی رابط گرافیکی:

رابط گرافیکی به صورت کامل آماده سازی شد و تغییر برخی جزئیات اعمال شد، تمامی صفحات مرتب شد و سعی شده است که المنت های تعاملی کاربر پسند باشد. یک نمونه فرم تکمیل شده و نتایج آن را با استفاده از مدل تغذیه در ادامه قرار داده میشود. البته اگر در قسمت ست ورزشی گزینه "اطلاعی ندارم" انتخاب شود، تابع حدس ست ورزشی خودکار اجرا می شود و سپس مقدار برگشتی در ورودی مدل تغذیه به کار گرفته میشود. تابعی در هر دو حالت تمرین و تغذیه وجود دارد که وضعیت بدنی را خودکار بر اساس شاخص توده بدنی که از قد و وزن به دست می آید، محاسبه کند. در صورتی که وضعیت بدنی وارد بشود نیز صحت آن بررسی میشود و اگر کاربر آن را درست وارد نکرده باشد، در پیش بینی از مقدار درست به دست آمده استفاده میشود.

جنسیت:

صفحات طراحی شده برای حالت موبایل نیز بهینه سازی شده اند و اپلیکیشن در حالت موبایل نیز قابل استفاده است.

The image displays three mobile app screens for a health assessment tool. The first screen, titled 'فرم درشت مغذی ها' (Macronutrient Form), contains input fields for height, weight, and activity level, along with gender selection. The second screen, 'خوش آمديد' (Welcome), provides instructions and offers recommendations for diet and exercise. The third screen, 'نتايج' (Results), shows a table of calculated values for various health metrics.

شاخصه	مقدار
جنسیت	مذکر
سن	22
قد	181
وزن	76
وضعیت بدن	عادی
شاخص توده بدنی	23.2
بیماری ها:	فشار خون-

Below the table, it indicates 'ماده مغذی' (Nutrient) and 'نیاز پیش بینی شده' (Estimated need).

چالش ها:

در اینجا چالش ها صرفاً جهت اطلاع بیان میشود و این امر به قصد توجیه عملکرد نیست.

اولین و مهمترین چالش توسعه، یافتن داده ای مناسب بود که کاملاً مربوط باشد و لیبل با معنی برای ارائه در این زمینه داشته باشد. همچنان این چالش وجود دارد و مدل های توسعه داده شده از نظر تئوری برای مشاوره تغذیه کامل نیستند. به هر صورت این نهایت توانایی بنده در این شرایط است و زمان زیادی از این پروژه در فضای وب و جستجوی داده مناسب در وبسایت ها و پایگاه های مختلف صرف شده است. همچنین فرآیند تست داده های یافت شده و برخورد با نتایج نامطلوب بخش بزرگ دیگری بود.

در حال حاضر محتوایی که وجود دارد بسیاری کمتر از زمان انرژی صرف شده در این پروژه می باشد. زیرا همانطور که بیان شد وقت این پروژه صرف تولید کد و یا محتوا نشده است. البته این مسئله اشتباه شخصی بنده بود. فقط از روی تعداد داده های موجود این برداشت وجود داشت که داده مناسب برای این مسئله یافت خواهد شد. متأسفانه حالتی که انتظار میرفت صورت نگرفت.

چالش بعدی این موضوع گسسته بودن روند کاری بود. در میان انجام بخش هایی از پروژه سایر مسائلی مانند ارائه های کلاسی امتحانات و فرآیند کارآموزی باعث قطع روند کاری و به تاخیر افتادن جریان پیشرفت میشد. البته که این مسئله بسیار بدیهی است و دانشجو باید در کنار این مسائل فرآیند توسعه پروژه را نیز کنترل کند. اما نمی توان از تاثیر آن در گسسته شدن جریان کاری و ضعف در کیفیت پروژه چشم پوشی کرد.

چالش دیگر برخی مشکلات نرم افزاری و گاهی سخت افزاری بود. برخی از کتابخانه ها با به درستی نصب نمی شدند، و در هنگام رفع این مشکل به درستی کار نمیکردند. در برخی موارد پلتفرم آناکوندا و برنامه پایتون (در IDE VSCode) به تداخل بر

میخوردند. تمامب کتابخانه ها در هسته آناکوندا و هسته ای که در IDE استفاده میشد به صورت جداگانه نصب شدند که البته این موارد منشا در کوتاهی خود بنده در یکسان سازی ورژن پایتون و یکپارچه سازی این پلتفرم ها بود. از لحاظ سخت افزاری نیز گاهها زمان اجرای الگوریتم ها بسیار طولانی میشد و پیغام های کافی نبودن رم نیز نمایش داده میشد. به هر نحو، و با چالش های موجود، این اپلیکیشن به حالتی که می بینید در آمده است.

منابع:

۱. وزارت کشاورزی ایالات متحده، <https://www.usda.gov>

۲. پایگاه کگل، <https://www.kaggle.com>

۳. تارنمای هاگینگ فیس، <https://huggingface.co>

۴. گروک AI، <https://grok.com>

۵. دیپ سیک، <https://www.deepseek.com>

۶. گوگل اسکولار، <https://scholar.google.com>

۷. انجمن علمی ماشین های محاسباتی، <https://www.acm.org>

۸. گیت هاب، <https://www.github.com>

تقدیر و تشکر:

در پایان تقدیر و تشکر به عمل آورده میشود از:

استاد پروژه آقای دکتر مسعود دادگر، خانواده و دوستان که در این بازه زمانی در کنار اینجانب حضور مادی و معنوی داشتند، همچنین کمک ها و حمایتشان را دریغ نکردند.