



DIAGNOSIS OF DIABETES

The Final Machine Learning Project



Dr. F Keshavarz

Mohammad Mahdi Shafighi

Winter 1403



دانشگاه شاهرود

FEBRUARY 4, 2025

SHAHED-TEHRAN UNIVERSITY
Iran-Tehran

فهرست مطالب

۴.....	۱. معرفی مجموعه داده
۴.....	۲. مشخصات مجموعه داده
۵.....	۳. توضیح ویژگی داده ها
۵.....	۴. بررسی مشکلات احتمالی در داده ها
۵.....	• مقادیر صفر غیرمنطقی:
۵.....	• عدم توزیع یکنواخت داده ها
۵.....	• عدم تعادل کلاس ها:
۵.....	۵. روش های پیشنهادی برای پیش پردازش داده ها
۶.....	۱. مقدمه (روند پروژه)
۶.....	۲. تحلیل مجموعه داده (PIMA Indians Diabetes Dataset)
۶.....	۲,۱ مشخصات مجموعه داده
۶.....	۲,۲ بررسی کیفیت داده ها
۶.....	• مقادیر صفر در برخی ویژگی ها:
۶.....	• توزیع نامتعادل کلاس ها:
۶.....	• تفاوت در مقیاس داده ها:
۷.....	۳. مراحل پردازش داده ها
۷.....	۳,۱ پیش پردازش داده ها
۷.....	۱. حذف یا جایگزینی مقادیر غیرمنطقی:
۷.....	۲. استانداردسازی داده ها:
۷.....	۳. تقسیم داده ها:
۸.....	۴. پیاده سازی و آموزش مدل ها
۸.....	۴,۱ مدل SVM ماشین بردار پشتیبان

- ۴,۲ مدل **Random Forest** جنگل تصادفی ۸
۵. ارزیابی و مقایسه مدل ها ۹
۶. ارائه نتایج و تحلیل نهایی ۱۰
- ۶,۱ نتایج و تفسیر ۱۰
- ۶,۲ پیشنهادات برای بهبود مدل ۱۰
۷. خروجی های نهایی مورد انتظار ۱۰
۸. نتیجه گیری ۱۱

بسمه تعالی

گزارش تحلیل مجموعه داده PIMA Indians Diabetes

"ورژن ها ۱،۱ و ۱،۲ پروژه"

۱. معرفی مجموعه داده

مجموعه داده **PIMA Indians Diabetes** یکی از مشهورترین دیتاست‌ها در حوزه پزشکی و یادگیری ماشین است که برای تشخیص دیابت استفاده می‌شود. این مجموعه شامل اطلاعات پزشکی زنان بالای ۲۱ سال از قبیله **Pima Indians** در ایالات متحده است. هدف اصلی این مجموعه داده، پیش‌بینی احتمال ابتلا به دیابت بر اساس متغیرهای پزشکی و بالینی است.

۲. مشخصات مجموعه داده

- تعداد نمونه‌ها: ۷۶۸
- تعداد ویژگی‌ها: ۸ ویژگی عددی + ۱ برچسب کلاس (Outcome) یا هدف (Target)
- نوع برچسب خروجی (Target Variable) باینری (۰ و ۱)

○ * فرد سالم (دیابت ندارد)

○ ۱ فرد مبتلا به دیابت

```
[ ] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Pregnancies                          768 non-null    int64
1   Glucose                              768 non-null    int64
2   BloodPressure                        768 non-null    int64
3   SkinThickness                       768 non-null    int64
4   Insulin                             768 non-null    int64
5   BMI                                 768 non-null    float64
6   DiabetesPedigreeFunction             768 non-null    float64
7   Age                                 768 non-null    int64
8   Outcome                             768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

۳. توضیح ویژگی داده ها

نام ویژگی	توضیحات	محدوده مقادیر
Pregnancies	تعداد بارداری ها	تا ۱۷
Glucose	(mg/dL) سطح گلوکز خون	تا ۱۹۹
BloodPressure	(mm Hg) فشار خون دیاستولیک	تا ۱۲۲
SkinThickness	(به عنوان نماینده چربی (mm) ضخامت پوست بدن)	تا ۹۹
Insulin	(mu U/ml) سطح انسولین سرم	تا ۸۴۶
BMI	(kg/m ²) شاخص توده بدنی	تا ۶۷,۱
DiabetesPedigreeFunction	تابع شجره نامه دیابت (ریسک ژنتیکی)	تا ۰,۰۸۵ ۲,۴۲
Age	سن بیمار (سال)	تا ۲۱ ۸۱
Outcome	وضعیت ابتلا به دیابت (۰: خیر، ۱: بله)	یا ۱

۴. بررسی مشکلات احتمالی در داده ها

- مقادیر صفر غیرمنطقی: ویژگی هایی مانند **Glucose**, **BloodPressure**, **SkinThickness**, **Insulin** و **BMI** نباید مقدار صفر داشته باشند، زیرا مقدار صفر برای این ویژگی ها از نظر پزشکی غیرممکن است.
- عدم توزیع یکنواخت داده ها: ممکن است برخی ویژگی ها دارای چگالی داده های نابرابر باشند که بر مدل تأثیر بگذارد.
- عدم تعادل کلاس ها: بررسی تعداد نمونه های دیابتی (۱) و غیردیابتی (۰) ضروری است.

۵. روش های پیشنهادی برای پیش پردازش داده ها

- جایگزینی مقادیر صفر با میانگین یا میانه در ویژگی های پزشکی غیرمنطقی.
- نرمال سازی داده ها (به ویژه برای الگوریتم SVM که به مقیاس ویژگی ها حساس است).
- تقسیم داده ها به ۷۰٪ آموزش و ۳۰٪ تست.

گزارش تحلیل دیتاست و روند پروژه تشخیص دیابت

۱. مقدمه (روند پروژه)

دیابت یکی از بیماری‌های مزمن و شایع است که تشخیص زودهنگام آن می‌تواند به مدیریت بهتر بیماری و کاهش عوارض آن کمک کند. در این پروژه، با استفاده از الگوریتم‌های یادگیری ماشین شامل ماشین بردار پشتیبان (SVM) و جنگل تصادفی (Random Forest)، سعی می‌کنیم مدلهایی برای تشخیص ابتلا به دیابت توسعه دهیم. این پروژه شامل مراحل تحلیل داده، پیش‌پردازش، آموزش مدل، ارزیابی عملکرد و مقایسه الگوریتم‌ها است.

۲. تحلیل مجموعه داده (PIMA Indians Diabetes Dataset)

۲.۱ مشخصات مجموعه داده

مجموعه داده PIMA Indians Diabetes شامل اطلاعات ۷۶۸ زن بالای ۲۱ سال از قبیله Pima Indians در ایالات متحده است. این مجموعه داده از ۸ ویژگی پزشکی و یک برچسب خروجی (Outcome) تشکیل شده است.

نام ویژگی	توضیحات	نوع داده
Pregnancies	تعداد بارداری‌ها	عدد صحیح
Glucose	سطح گلوکز خون	عدد پیوسته
BloodPressure	فشار خون دیاستولیک	عدد پیوسته
SkinThickness	ضخامت پوست	عدد پیوسته
Insulin	سطح انسولین سرم	عدد پیوسته
BMI	شاخص توده بدنی	عدد پیوسته
DiabetesPedigreeFunction	تابع شجره‌نامه دیابت (ریسک ژنتیکی)	عدد پیوسته
Age	سن بیمار	عدد صحیح
Outcome	وضعیت ابتلا به دیابت (۰: سالم، ۱: بیمار)	عدد باینری

۲.۲ بررسی کیفیت داده‌ها

- مقادیر صفر در برخی ویژگی‌ها: در ویژگی‌هایی مانند Glucose، BloodPressure، SkinThickness، BMI و Insulin، مقدار صفر غیرمنطقی است و باید اصلاح شود.
- توزیع نامتعادل کلاس‌ها: ممکن است داده‌ها توزیع نامتوازن داشته باشند که بر عملکرد مدل‌ها تأثیر بگذارد.
- تفاوت در مقیاس داده‌ها: برخی ویژگی‌ها دارای بازه‌های متفاوتی هستند که می‌تواند روی عملکرد مدل‌ها تأثیر بگذارد.

۳. مراحل پردازش داده‌ها

۳.۱ پیش‌پردازش داده‌ها

۱. حذف یا جایگزینی مقادیر غیرمنطقی:

- مقادیر صفر در ویژگی‌های پزشکی با میانگین یا میانۀ جایگزین می‌شوند.

۱۱. استانداردسازی داده‌ها:

- داده‌ها با استفاده از **StandardScaler** استاندارد می‌شوند تا مقیاس‌بندی ویژگی‌ها هماهنگ شود.

۱۱۱. تقسیم داده‌ها:

- داده‌ها به دو بخش ۷۰٪ برای آموزش و ۳۰٪ برای تست تقسیم می‌شوند.

((همانطور که در شکل زیر مشاهده می‌شود، مقادیر غیر منطقی و یا صفر با میانۀ جایگزین شده اند.))

df.head(30)

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	125	33.6	0.627	50	1
1	1	85	66	29	125	26.6	0.351	31	0
2	8	183	64	29	125	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	4	137	40	35	168	43.1	2.288	33	1
5	5	116	74	29	125	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	72	29	125	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	29	125	32.3	0.232	54	1
10	4	110	92	29	125	37.6	0.191	30	0
11	10	168	74	29	125	38.0	0.537	34	1
12	10	139	80	29	125	27.1	1.441	57	0
13	1	189	60	23	846	30.1	0.398	59	1
14	5	166	72	19	175	25.8	0.587	51	1
15	7	100	72	29	125	30.0	0.484	32	1

٤. پیاده‌سازی و آموزش مدل‌ها

٤,١ مدل SVM ماشین بردار پشتیبان

- دو نوع کرنل بررسی می‌شود:
 - کرنل خطی
 - کرنل RBF یا شعاعی
- تنظیم هایپر پارامترها با Grid Search انجام می‌شود.

٤,٢ مدل Random Forest جنگل تصادفی

- از ١٠٠ درخت تصمیم‌گیری ($n_estimators=100$) استفاده می‌شود.
- بررسی اثر تعداد درخت‌ها بر دقت مدل.

```
# rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
# rf_model.fit(X_train, y_train)

# Random Forest model
# rf_model = RandomForestClassifier(n_estimators=110, random_state=42) ## --> n_estimators = 110
# rf_model = RandomForestClassifier(n_estimators=90, random_state=42) ## --> n_estimators = 90
# rf_model = RandomForestClassifier(n_estimators=95, random_state=42) ## --> n_estimators = 95
# rf_model = RandomForestClassifier(n_estimators=200, random_state=42) ## --> n_estimators = 200
rf_model = RandomForestClassifier(n_estimators=300, random_state=42) ## --> n_estimators = 300 **
# rf_model = RandomForestClassifier(n_estimators=1000, random_state=42) ## --> n_estimators = 1000
# rf_model = RandomForestClassifier(n_estimators=500, random_state=42) ## --> n_estimators = 500
# rf_model = RandomForestClassifier(n_estimators=310, random_state=42) ## --> n_estimators = 310

rf_model.fit(X_train, y_train)
rf_probs = rf_model.predict_proba(X_test)[: , 1]

# Make predictions on the test set

[ ] y_pred_rf = rf_model.predict(X_test)
```

```
[ ] accuracy_rf = accuracy_score(y_test, y_pred_rf)
print(f"Accuracy of the Random Forest model: {accuracy_rf}")
# Accuracy of the Random Forest model: 0.7597402597402597 ## ---> the first result

# n_estimator Changes:
## Change 1 :(n_estimators=110) Accuracy of the Random Forest model: 0.7467532467532467
## Change 2 :(n_estimators=90) Accuracy of the Random Forest model: 0.7532467532467533
## Change 3 :(n_estimators=95) Accuracy of the Random Forest model: 0.7597402597402597
## as samev the default 0.7597402597402597
## Change 4 :(n_estimators=200) Accuracy of the Random Forest model: 0.7662337662337663 (growth)
## Change 5 :(n_estimators=300) Accuracy of the Random Forest model: 0.7792207792207793 (growth more) **
## Change 6 :(n_estimators=1000) Accuracy of the Random Forest model: 0.7662337662337663 (decreased significantly == 200)
## Change 6 :(n_estimators=500) Accuracy of the Random Forest model: 0.7727272727272727 (increased significantly <= 300)
## Change 6 :(n_estimators=310) Accuracy of the Random Forest model: 0.7727272727272727 (increased significantly <= 300)
```

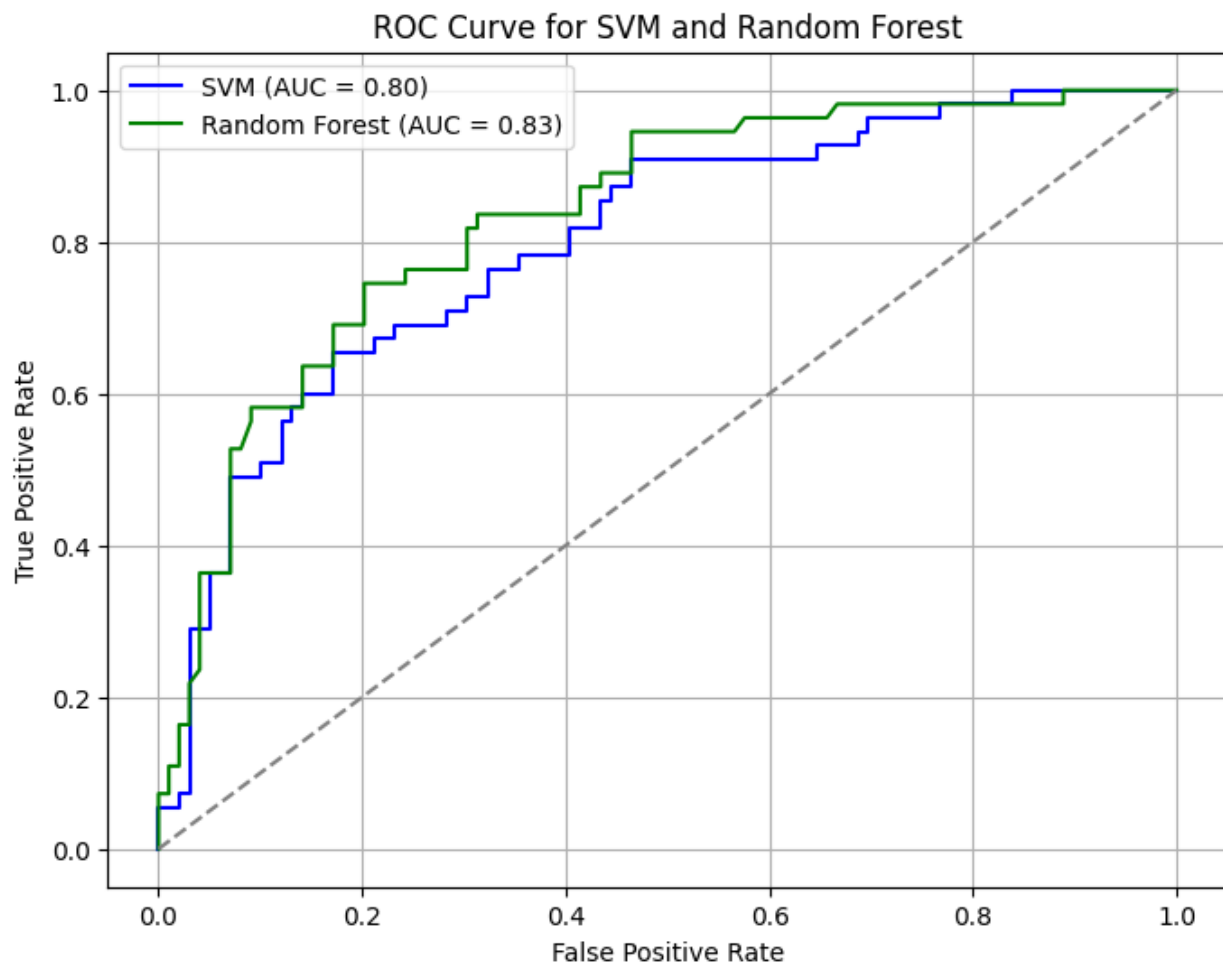
Accuracy of the Random Forest model: 0.7792207792207793

۵. ارزیابی و مقایسه مدل‌ها

- محاسبه معیارهای عملکرد برای هر مدل:

- دقت Accuracy
- دقت مثبت Precision
- بازخوانی مثبت Recall
- F1-Score
- AUC و ROC Curve

- مقایسه دو مدل بر اساس معیارهای فوق و تحلیل نقاط ضعف و قوت هر یک.



۶. ارائه نتایج و تحلیل نهایی

۶.۱ نتایج و تفسیر

- بررسی اینکه کدام مدل دقت بیشتری دارد؟
- بررسی عملکرد مدل‌ها در شناسایی بیماران دیابتی.
- تحلیل اینکه آیا **Random Forest** به برخی ویژگی‌ها حساس‌تر است؟

۶.۲ پیشنهادات برای بهبود مدل

- استفاده از روش‌های افزایش داده‌ها (**Data Augmentation**) برای مقابله با عدم توازن داده‌ها.
- بررسی دیگر الگوریتم‌های یادگیری ماشین مانند شبکه‌های عصبی مصنوعی (ANN).

۷. خروجی‌های نهایی مورد انتظار



کدهای اجرایی در محیط **Google Colab**



گزارش **PDF** شامل تحلیل داده و نتایج مقایسه مدل‌ها

نمودارهای مربوط به ارزیابی مدل‌ها **ROC Curve** ، **Confusion Matrix**.

نتیجه‌گیری درباره بهترین مدل برای تشخیص دیابت

```

➡ Best Hyperparameters: {'C': 100, 'gamma': 0.001, 'kernel': 'rbf'}
Accuracy of the optimized SVM model: 0.7792207792207793

```

	precision	recall	f1-score	support
0	0.82	0.85	0.83	99
1	0.71	0.65	0.68	55
accuracy			0.78	154
macro avg	0.76	0.75	0.76	154
weighted avg	0.78	0.78	0.78	154

۸. نتیجه گیری

این پروژه با هدف مقایسه عملکرد دو الگوریتم **SVM** و **Random Forest** در تشخیص دیابت انجام می شود. پس از تحلیل داده ها، پیش پردازش، آموزش مدل ها، و ارزیابی عملکرد، مشخص می شود که کدام مدل عملکرد بهتری در تشخیص بیماران دیابتی دارد و چگونه می توان مدل ها را بهبود داد.

"همچنین در ورژن ۱,۳ Grid Search هم بررسی خواهد شد."