**Gender**

Gender refers to the biological sex of the individual, which can have an impact on their susceptibility to diabetes. There are three categories in it male ,female and other.

**grid\_3x3agesort**

Age is an important factor as diabetes is more commonly diagnosed in older adults.Age ranges from 0-80 in our dataset.

**grid\_3x3hypertensionsort**

Hypertension is a medical condition in which the blood pressure in the arteries is persistently elevated. It has values a 0 or 1 where 0 indicates they don’t have hypertension and for 1 it means they have hypertension.

**grid\_3x3heart\_diseasesort**

Heart disease is another medical condition that is associated with an increased risk of developing diabetes. It has values a 0 or 1 where 0 indicates they don’t have heart disease and for 1 it means they have heart disease.

**text\_formatsmoking\_historysort**

Smoking history is also considered a risk factor for diabetes and can exacerbate the complications associated with diabetes.In our dataset we have 5 categories i.e not current,former,No Info,current,never and ever.

**grid\_3x3bmisort**

BMI (Body Mass Index) is a measure of body fat based on weight and height. Higher BMI values are linked to a higher risk of diabetes. The range of BMI in the dataset is from 10.16 to 71.55. BMI less than 18.5 is underweight, 18.5-24.9 is normal, 25-29.9 is overweight, and 30 or more is obese.

**grid\_3x3HbA1c\_levelsort**

HbA1c (Hemoglobin A1c) level is a measure of a person's average blood sugar level over the past 2-3 months. Higher levels indicate a greater risk of developing diabetes. Mostly more than 6.5% of HbA1c Level indicates diabetes.

**grid\_3x3blood\_glucose\_levelsort**

Blood glucose level refers to the amount of glucose in the bloodstream at a given time. High blood glucose levels are a key indicator of diabetes.

**grid\_3x3diabetessort**

Diabetes is the target variable being predicted, with values of 1 indicating the presence of diabetes and 0 not presence

Machine Learning Techniques:

**Diabetes Dataset All**

Our aim is to use the optimised package with the best ML techniques and optimise hyperparameters for each data in our digital health assistant tool.

We use AutoML techniques to find the best machine-learning model with optimized process parameters. In addition, we compare AotoML model techniques from several ML packages including Pycaret, H20, Auto-sklearn and TOPOT. We aim to use the best ML algorithms with optimised process parameters from the best fit packages.

I want to use AutoML techniques for all datasets to find the best ML models. I will compare the 5 best models from each library. I aim to find the best algorithm for the data.

AutoML Libraries Are:

**Pycaret**

H2O, Auto-sklearn, or TPOT

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **AUC** | **Recall** | **Prec.** | **F1** | **Kappa** | **MCC** | **TT (Sec)** |
| **Ada Boost Classifier** | **0.9720** | 0.9789 | 0.6891 | 0.9741 | 0.8071 | 0.7925 | 0.8063 | 1.6170 |
| Gradient Boosting Classifier | 0.9719 | 0.9790 | 0.6817 | 0.9826 | 0.8049 | 0.7903 | 0.8057 | 5.5110 |
| Light Gradient Boosting Machine | 0.9717 | 0.9784 | 0.6871 | 0.9715 | 0.8048 | 0.7900 | 0.8039 | 2.0710 |
| Extreme Gradient Boosting | 0.9708 | 0.9768 | 0.6929 | 0.9504 | 0.8014 | 0.7860 | 0.7975 | 0.5040 |
| Random Forest Classifier | 0.9699 | 0.9610 | 0.6872 | 0.9430 | 0.7949 | 0.7791 | 0.7905 | 4.9790 |
| Extra Trees Classifier | 0.9673 | 0.9542 | 0.6861 | 0.9064 | 0.7809 | 0.7635 | 0.7723 | 3.5290 |
| Logistic Regression | 0.9603 | 0.9613 | 0.6272 | 0.8691 | 0.7283 | 0.7075 | 0.7186 | 1.3840 |
| Linear Discriminant Analysis | 0.9562 | 0.9542 | 0.5847 | 0.8547 | 0.6940 | 0.6714 | 0.6856 | 0.1010 |
| SVM - Linear Kernel | 0.9543 | 0.9604 | 0.6605 | 0.7886 | 0.7115 | 0.6871 | 0.6944 | 1.0000 |
| K Neighbors Classifier | 0.9528 | 0.8726 | 0.5239 | 0.8687 | 0.6533 | 0.6297 | 0.6530 | 0.9090 |
| Decision Tree Classifier | 0.9508 | 0.8540 | 0.7363 | 0.7008 | 0.7179 | 0.6910 | 0.6914 | 0.1940 |
| Ridge Classifier | 0.9399 | 0.9542 | 0.2961 | 0.9892 | 0.4555 | 0.4333 | 0.5236 | 0.0780 |
| Dummy Classifier | 0.9150 | 0.5000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0570 |
| Quadratic Discriminant Analysis | 0.9050 | 0.9211 | 0.6519 | 0.4589 | 0.5385 | 0.4874 | 0.4968 | 0.1060 |
| Naive Bayes | 0.9028 | 0.9222 | 0.6471 | 0.4505 | 0.5311 | 0.4788 | 0.4887 | 0.0840 |

Best Model

|  | **Accuracy** | **AUC** |  | **Recall** | **Prec.** | **F1** | **Kappa** | **MCC** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fold** |  |  |  |  |  |  |  |  |
| **0** | 0.9733 | 0.9737 |  | 0.6857 | 1.0000 | 0.8136 | 0.7997 | 0.8162 |
| **1** | 0.9710 | 0.9769 |  | 0.6588 | 1.0000 | 0.7943 | 0.7794 | 0.7991 |
| **2** | 0.9713 | 0.9750 |  | 0.6622 | 1.0000 | 0.7968 | 0.7820 | 0.8013 |
| **3** | 0.9714 | 0.9746 |  | 0.6639 | 1.0000 | 0.7980 | 0.7833 | 0.8023 |
| **4** | 0.9710 | 0.9718 |  | 0.6588 | 1.0000 | 0.7943 | 0.7794 | 0.7991 |
| **5** | 0.9696 | 0.9733 |  | 0.6420 | 1.0000 | 0.7820 | 0.7665 | 0.7883 |
| **6** | 0.9731 | 0.9764 |  | 0.6840 | 1.0000 | 0.8124 | 0.7985 | 0.8152 |
| **7** | 0.9730 | 0.9746 |  | 0.6824 | 1.0000 | 0.8112 | 0.7972 | 0.8141 |
| **8** | 0.9704 | 0.9738 |  | 0.6521 | 1.0000 | 0.7894 | 0.7743 | 0.7948 |
| **9** | 0.9723 | 0.9758 |  | 0.6739 | 1.0000 | 0.8052 | 0.7909 | 0.8088 |
| **Mean** | 0.9716 | 0.9746 |  | 0.6664 | 1.0000 | 0.7997 | 0.7851 | 0.8039 |
| **Std** | 0.0012 | 0.0014 |  | 0.0139 | 0.0000 | 0.0100 | 0.0106 | 0.0089 |

Fitting 10 folds for each of 10 candidates, totalling 100 fits

Original model was better than the tuned model, hence it will be returned. NOTE: The display metrics are for the tuned model (not the original one).

|  | **Accuracy** | **AUC** | **Recall** | **Prec.** | **F1** | **Kappa** | **MCC** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Fold** |  |  |  |  |  |  |  |
| **0** | 0.9730 | 0.9778 | 0.6958 | 0.9810 | 0.8142 | 0.8001 | 0.8138 |
| **1** | 0.9719 | 0.9795 | 0.6824 | 0.9807 | 0.8048 | 0.7901 | 0.8052 |
| **2** | 0.9717 | 0.9785 | 0.6824 | 0.9783 | 0.8040 | 0.7892 | 0.8041 |
| **3** | 0.9720 | 0.9783 | 0.6874 | 0.9761 | 0.8067 | 0.7921 | 0.8063 |
| **4** | 0.9696 | 0.9759 | 0.6739 | 0.9548 | 0.7901 | 0.7743 | 0.7879 |
| **5** | 0.9699 | 0.9755 | 0.6655 | 0.9706 | 0.7896 | 0.7740 | 0.7900 |
| **6** | 0.9739 | 0.9789 | 0.7076 | 0.9791 | 0.8215 | 0.8078 | 0.8202 |
| **7** | 0.9737 | 0.9782 | 0.7025 | 0.9835 | 0.8196 | 0.8059 | 0.8191 |
| **8** | 0.9706 | 0.9778 | 0.6689 | 0.9779 | 0.7944 | 0.7792 | 0.7955 |
| **9** | 0.9734 | 0.9775 | 0.7025 | 0.9789 | 0.8180 | 0.8041 | 0.8170 |
| **Mean** | 0.9720 | 0.9778 | 0.6869 | 0.9761 | 0.8063 | 0.7917 | 0.8059 |
| **Std** | 0.0015 | 0.0012 | 0.0141 | 0.0078 | 0.0114 | 0.0121 | 0.0112 |

Fitting 10 folds for each of 10 candidates, totalling 100 fits

|  | **Accuracy** | **AUC** | **Recall** | **Prec.** | **F1** | **Kappa** | **MCC** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Fold** |  |  |  |  |  |  |  |
| **0** | 0.9727 | 0.9789 | 0.6924 | 0.9810 | 0.8118 | 0.7976 | 0.8117 |
| **1** | 0.9723 | 0.9809 | 0.6807 | 0.9902 | 0.8068 | 0.7924 | 0.8086 |
| **2** | 0.9719 | 0.9792 | 0.6773 | 0.9877 | 0.8036 | 0.7890 | 0.8053 |
| **3** | 0.9723 | 0.9795 | 0.6824 | 0.9878 | 0.8072 | 0.7928 | 0.8085 |
| **4** | 0.9706 | 0.9771 | 0.6723 | 0.9732 | 0.7952 | 0.7799 | 0.7955 |
| **5** | 0.9701 | 0.9769 | 0.6639 | 0.9777 | 0.7908 | 0.7753 | 0.7922 |
| **6** | 0.9741 | 0.9800 | 0.7025 | 0.9905 | 0.8220 | 0.8085 | 0.8224 |
| **7** | 0.9733 | 0.9797 | 0.6975 | 0.9834 | 0.8161 | 0.8022 | 0.8160 |
| **8** | 0.9710 | 0.9784 | 0.6655 | 0.9900 | 0.7960 | 0.7810 | 0.7989 |
| **9** | 0.9731 | 0.9789 | 0.6941 | 0.9857 | 0.8146 | 0.8006 | 0.8149 |
| **Mean** | 0.9721 | 0.9790 | 0.6829 | 0.9847 | 0.8064 | 0.7919 | 0.8074 |
| **Std** | 0.0012 | 0.0012 | 0.0128 | 0.0056 | 0.0096 | 0.0102 | 0.0091 |

Fitting 10 folds for each of 10 candidates, totalling 100 fits

|  | **Accuracy** | **AUC** | **Recall** | **Prec.** | **F1** | **Kappa** | **MCC** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Fold** |  |  |  |  |  |  |  |
| **0** | 0.9150 | 0.9726 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| **1** | 0.9150 | 0.9744 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| **2** | 0.9150 | 0.9722 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| **3** | 0.9150 | 0.9739 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| **4** | 0.9150 | 0.9686 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| **5** | 0.9150 | 0.9702 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| **6** | 0.9150 | 0.9734 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| **7** | 0.9150 | 0.9719 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| **8** | 0.9150 | 0.9712 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| **9** | 0.9150 | 0.9733 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| **Mean** | 0.9150 | 0.9722 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| **Std** | 0.0000 | 0.0017 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

Fitting 10 folds for each of 10 candidates, totalling 100 fits

Original model was better than the tuned model, hence it will be returned. NOTE: The display metrics are for the tuned model (not the original one).

1 to 3 of 3 entriesFilter

|  |  |  |
| --- | --- | --- |
| **Initiated** | . . . . . . . . . . . . . . . . . . | 15:20:24 |
| **Status** | . . . . . . . . . . . . . . . . . . | Compiling Final Results |
| **Estimator** | . . . . . . . . . . . . . . . . . . | Random Forest Classifier |

Show 102550100 per page

Like what you see? Visit the [data table notebook](https://colab.research.google.com/notebooks/data_table.ipynb) to learn more about interactive tables.

Fitting 10 folds for each of 10 candidates, totalling 100 fits

|  | **Accuracy** | **AUC** | **Recall** | **Prec.** | **F1** | **Kappa** | **MCC** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Fold** |  |  |  |  |  |  |  |
| **0** | 0.9733 | 0.9698 | 0.6857 | 1.0000 | 0.8136 | 0.7997 | 0.8162 |
| **1** | 0.9710 | 0.9729 | 0.6588 | 1.0000 | 0.7943 | 0.7794 | 0.7991 |
| **2** | 0.9713 | 0.9716 | 0.6622 | 1.0000 | 0.7968 | 0.7820 | 0.8013 |
| **3** | 0.9714 | 0.9719 | 0.6639 | 1.0000 | 0.7980 | 0.7833 | 0.8023 |
| **4** | 0.9711 | 0.9689 | 0.6605 | 1.0000 | 0.7955 | 0.7807 | 0.8002 |
| **5** | 0.9696 | 0.9689 | 0.6420 | 1.0000 | 0.7820 | 0.7665 | 0.7883 |
| **6** | 0.9731 | 0.9716 | 0.6840 | 1.0000 | 0.8124 | 0.7985 | 0.8152 |
| **7** | 0.9730 | 0.9722 | 0.6824 | 1.0000 | 0.8112 | 0.7972 | 0.8141 |
| **8** | 0.9706 | 0.9713 | 0.6538 | 1.0000 | 0.7907 | 0.7756 | 0.7959 |
| **9** | 0.9724 | 0.9700 | 0.6756 | 1.0000 | 0.8064 | 0.7922 | 0.8099 |
| **Mean** | 0.9717 | 0.9709 | 0.6669 | 1.0000 | 0.8001 | 0.7855 | 0.8042 |
| **Std** | 0.0012 | 0.0013 | 0.0138 | 0.0000 | 0.0099 | 0.0104 | 0.0088 |

**Top 5 Models with Optimized Hyperparameters:**

Model 1: AdaBoostClassifier(algorithm='SAMME.R', estimator=None, learning\_rate=1.0,

n\_estimators=50, random\_state=42)

Accuracy AUC Recall Prec. F1 Kappa MCC

Fold

0 0.9733 0.9698 0.6857 1.0 0.8136 0.7997 0.8162

1 0.9710 0.9729 0.6588 1.0 0.7943 0.7794 0.7991

2 0.9713 0.9716 0.6622 1.0 0.7968 0.7820 0.8013

3 0.9714 0.9719 0.6639 1.0 0.7980 0.7833 0.8023

4 0.9711 0.9689 0.6605 1.0 0.7955 0.7807 0.8002

5 0.9696 0.9689 0.6420 1.0 0.7820 0.7665 0.7883

6 0.9731 0.9716 0.6840 1.0 0.8124 0.7985 0.8152

7 0.9730 0.9722 0.6824 1.0 0.8112 0.7972 0.8141

8 0.9706 0.9713 0.6538 1.0 0.7907 0.7756 0.7959

9 0.9724 0.9700 0.6756 1.0 0.8064 0.7922 0.8099

Mean 0.9717 0.9709 0.6669 1.0 0.8001 0.7855 0.8042

Std 0.0012 0.0013 0.0138 0.0 0.0099 0.0104 0.0088

==================================================

Model 2: GradientBoostingClassifier(ccp\_alpha=0.0, criterion='friedman\_mse', init=None,

learning\_rate=0.05, loss='log\_loss', max\_depth=6,

max\_features='sqrt', max\_leaf\_nodes=None,

min\_impurity\_decrease=0.3, min\_samples\_leaf=4,

min\_samples\_split=10, min\_weight\_fraction\_leaf=0.0,

n\_estimators=270, n\_iter\_no\_change=None,

random\_state=42, subsample=0.7, tol=0.0001,

validation\_fraction=0.1, verbose=0,

warm\_start=False)

Accuracy AUC Recall Prec. F1 Kappa MCC

Fold

0 0.9733 0.9698 0.6857 1.0 0.8136 0.7997 0.8162

1 0.9710 0.9729 0.6588 1.0 0.7943 0.7794 0.7991

2 0.9713 0.9716 0.6622 1.0 0.7968 0.7820 0.8013

3 0.9714 0.9719 0.6639 1.0 0.7980 0.7833 0.8023

4 0.9711 0.9689 0.6605 1.0 0.7955 0.7807 0.8002

5 0.9696 0.9689 0.6420 1.0 0.7820 0.7665 0.7883

6 0.9731 0.9716 0.6840 1.0 0.8124 0.7985 0.8152

7 0.9730 0.9722 0.6824 1.0 0.8112 0.7972 0.8141

8 0.9706 0.9713 0.6538 1.0 0.7907 0.7756 0.7959

9 0.9724 0.9700 0.6756 1.0 0.8064 0.7922 0.8099

Mean 0.9717 0.9709 0.6669 1.0 0.8001 0.7855 0.8042

Std 0.0012 0.0013 0.0138 0.0 0.0099 0.0104 0.0088

==================================================

Model 3: LGBMClassifier(bagging\_fraction=0.6, bagging\_freq=5, boosting\_type='gbdt',

class\_weight=None, colsample\_bytree=1.0, feature\_fraction=1.0,

importance\_type='split', learning\_rate=0.1, max\_depth=-1,

min\_child\_samples=71, min\_child\_weight=0.001, min\_split\_gain=0.6,

n\_estimators=130, n\_jobs=-1, num\_leaves=4, objective=None,

random\_state=42, reg\_alpha=0.3, reg\_lambda=4, subsample=1.0,

subsample\_for\_bin=200000, subsample\_freq=0)

Accuracy AUC Recall Prec. F1 Kappa MCC

Fold

0 0.9733 0.9698 0.6857 1.0 0.8136 0.7997 0.8162

1 0.9710 0.9729 0.6588 1.0 0.7943 0.7794 0.7991

2 0.9713 0.9716 0.6622 1.0 0.7968 0.7820 0.8013

3 0.9714 0.9719 0.6639 1.0 0.7980 0.7833 0.8023

4 0.9711 0.9689 0.6605 1.0 0.7955 0.7807 0.8002

5 0.9696 0.9689 0.6420 1.0 0.7820 0.7665 0.7883

6 0.9731 0.9716 0.6840 1.0 0.8124 0.7985 0.8152

7 0.9730 0.9722 0.6824 1.0 0.8112 0.7972 0.8141

8 0.9706 0.9713 0.6538 1.0 0.7907 0.7756 0.7959

9 0.9724 0.9700 0.6756 1.0 0.8064 0.7922 0.8099

Mean 0.9717 0.9709 0.6669 1.0 0.8001 0.7855 0.8042

Std 0.0012 0.0013 0.0138 0.0 0.0099 0.0104 0.0088

==================================================

Model 4: XGBClassifier(base\_score=None, booster='gbtree', callbacks=None,

colsample\_bylevel=None, colsample\_bynode=None,

colsample\_bytree=None, device='cpu', early\_stopping\_rounds=None,

enable\_categorical=False, eval\_metric=None, feature\_types=None,

gamma=None, grow\_policy=None, importance\_type=None,

interaction\_constraints=None, learning\_rate=None, max\_bin=None,

max\_cat\_threshold=None, max\_cat\_to\_onehot=None,

max\_delta\_step=None, max\_depth=None, max\_leaves=None,

min\_child\_weight=None, missing=nan, monotone\_constraints=None,

multi\_strategy=None, n\_estimators=None, n\_jobs=-1,

num\_parallel\_tree=None, objective='binary:logistic', ...)

Accuracy AUC Recall Prec. F1 Kappa MCC

Fold

0 0.9733 0.9698 0.6857 1.0 0.8136 0.7997 0.8162

1 0.9710 0.9729 0.6588 1.0 0.7943 0.7794 0.7991

2 0.9713 0.9716 0.6622 1.0 0.7968 0.7820 0.8013

3 0.9714 0.9719 0.6639 1.0 0.7980 0.7833 0.8023

4 0.9711 0.9689 0.6605 1.0 0.7955 0.7807 0.8002

5 0.9696 0.9689 0.6420 1.0 0.7820 0.7665 0.7883

6 0.9731 0.9716 0.6840 1.0 0.8124 0.7985 0.8152

7 0.9730 0.9722 0.6824 1.0 0.8112 0.7972 0.8141

8 0.9706 0.9713 0.6538 1.0 0.7907 0.7756 0.7959

9 0.9724 0.9700 0.6756 1.0 0.8064 0.7922 0.8099

Mean 0.9717 0.9709 0.6669 1.0 0.8001 0.7855 0.8042

Std 0.0012 0.0013 0.0138 0.0 0.0099 0.0104 0.0088

==================================================

Model 5: RandomForestClassifier(bootstrap=True, ccp\_alpha=0.0, class\_weight={},

criterion='gini', max\_depth=9, max\_features='sqrt',

max\_leaf\_nodes=None, max\_samples=None,

min\_impurity\_decrease=0, min\_samples\_leaf=6,

min\_samples\_split=5, min\_weight\_fraction\_leaf=0.0,

monotonic\_cst=None, n\_estimators=120, n\_jobs=-1,

oob\_score=False, random\_state=42, verbose=0,

warm\_start=False)

Accuracy AUC Recall Prec. F1 Kappa MCC

Fold

0 0.9733 0.9698 0.6857 1.0 0.8136 0.7997 0.8162

1 0.9710 0.9729 0.6588 1.0 0.7943 0.7794 0.7991

2 0.9713 0.9716 0.6622 1.0 0.7968 0.7820 0.8013

3 0.9714 0.9719 0.6639 1.0 0.7980 0.7833 0.8023

4 0.9711 0.9689 0.6605 1.0 0.7955 0.7807 0.8002

5 0.9696 0.9689 0.6420 1.0 0.7820 0.7665 0.7883

6 0.9731 0.9716 0.6840 1.0 0.8124 0.7985 0.8152

7 0.9730 0.9722 0.6824 1.0 0.8112 0.7972 0.8141

8 0.9706 0.9713 0.6538 1.0 0.7907 0.7756 0.7959

9 0.9724 0.9700 0.6756 1.0 0.8064 0.7922 0.8099

Mean 0.9717 0.9709 0.6669 1.0 0.8001 0.7855 0.8042

Std 0.0012 0.0013 0.0138 0.0 0.0099 0.0104 0.0088

==================================================