**Kushal\_Srinivas\_finalproject**

Github:https://github.com/kushal-srinivas/diabetes-prediction-project.git

**1. Final Project - Data Mining**  
**2. Using KNN, SVM, RF, and LSTM to Predict Diabetes**

**2.1 Goal**

"My project aims to implement a variety of machine learning classification algorithms, along with a deep learning model, to predict the likelihood of a patient having diabetes. This prediction is based on specific diagnostic measurements provided in the dataset."

**2.1.1 Importing the Packages and Libraries that are Required for the Project**

**A screenshot of a computer program

Description automatically generated2.1.2 Loading Data and Preprocessing**

python

diab = pd.read\_csv('diabetes.csv')

diab.describe()

| **Column** | **Count** | **Mean** | **Std** | **Min** | **25%** | **50%** | **75%** | **Max** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pregnancies | 768 | 3.845 | 3.37 | 0 | 1 | 3 | 6 | 17 |
| Glucose | 768 | 120.89 | 31.97 | 0 | 99 | 117 | 140.25 | 199 |
| BloodPressure | 768 | 69.11 | 19.36 | 0 | 62 | 72 | 80 | 122 |
| SkinThickness | 768 | 20.54 | 15.95 | 0 | 23 | 32 | 36.6 | 99 |
| Insulin | 768 | 79.8 | 115.24 | 0 | 29 | 32 | 41 | 846 |
| BMI | 768 | 31.99 | 7.88 | 0 | 27.3 | 32 | 36.6 | 67.1 |
| DiabetesPedigreeFunction | 768 | 0.47 | 0.33 | 0 | 0.24 | 0.37 | 0.63 | 2.42 |
| Age | 768 | 33.24 | 11.76 | 21 | 24 | 29 | 41 | 81 |
| Outcome | 768 | 0.35 | 0.48 | 0 | 0 | 0 | 1 | 1 |

diab.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 float64

2 BloodPressure 768 non-null float64

3 SkinThickness 768 non-null float64

4 Insulin 768 non-null float64

5 BMI 768 non-null float64

6 DiabetesPedigreeFunction 768 non-null float64

7 Age 768 non-null int64

8 Outcome 768 non-null int64

**2.1.3 Imputing Missing Values**

A screen shot of a computer code

Description automatically generateddiab.head()

| **Pregnancies** | **Glucose** | **BloodPressure** | **SkinThickness** | **Insulin** | **BMI** | **DiabetesPedigreeFunction** | **Age** | **Outcome** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | 148.0 | 72.0 | 35.0 | 125.0 | 33.6 | 0.627 | 50 | 1 |
| 1 | 85.0 | 66.0 | 29.0 | 125.0 | 26.6 | 0.351 | 31 | 0 |
| 8 | 183.0 | 64.0 | 29.0 | 125.0 | 23.3 | 0.672 | 32 | 1 |
| 1 | 89.0 | 66.0 | 23.0 | 94.0 | 28.1 | 0.167 | 21 | 0 |
| 0 | 137.0 | 40.0 | 35.0 | 168.0 | 43.1 | 2.288 | 33 | 1 |

**2.1.4 Separating the Dataset into Features and Output Label**

**A screen shot of a computer code

Description automatically generated**

**2.1.5 Data Visualization**

In the dataset, there is an observable data imbalance where the number of patients without diabetes is almost double the number of patients with diabetes.

To address this, we can employ stratified sampling in the train-test split, ensuring that both the training and testing datasets maintain the same label ratio. Additionally, we will use stratified k-fold cross-validation.

**A close-up of a text

Description automatically generated2.1.6 Checking for Data Imbalance**

A screen shot of a computer code

Description automatically generated

A graph with a bar and a number

Description automatically generated with medium confidence

----------Checking for Data Imbalance------------

Number of Positive Outcomes: 268

Percentage of Positive Outcomes: 34.88%

Number of Negative Outcomes: 500

Percentage of Negative Outcomes: 65.12%

2.1.5 Checking for Correlation between attributes

A screenshot of a computer code

Description automatically generated

A diagram of a heat map

Description automatically generated with medium confidence

**Observations:**

* Analyze the highest correlations and describe key observations (e.g., any pairs with high correlation values).
* Highlight whether the attributes exhibit multicollinearity or are suitable for modeling.

2.1.6 Visualizing the Distribution of Values for Each Attribute

A math equations on a white background

Description automatically generated

A graph of different types of blood pressure

Description automatically generated with medium confidence

**Observations:**

* Discuss patterns in the distributions (e.g., symmetry, skewness).
* Highlight specific features with outliers or unique distribution characteristics.

2.1.7 Pairwise Bivariate Distributions

A close-up of a computer screen

Description automatically generated

A graph with many columns

Description automatically generated with medium confidence

2.1.8 Train Test Data Split A screenshot of a computer code

Description automatically generated

2.1.9 Normalize the Dataset

A close-up of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

2.1.10 Define the necessary function for model fitting and metric calculation.

A screenshot of a computer

Description automatically generated

**2.2 Selecting Classification Algorithms**

2.2.1 I have decided to select the following classification algorithms:

* **K-Nearest Neighbors (KNN)**
* **Random Forest (RF)**
* **Support Vector Machine (SVM)**

2.2.2 For a deep learning algorithm, I have decided to use **Long Short-Term Memory (LSTM)**.

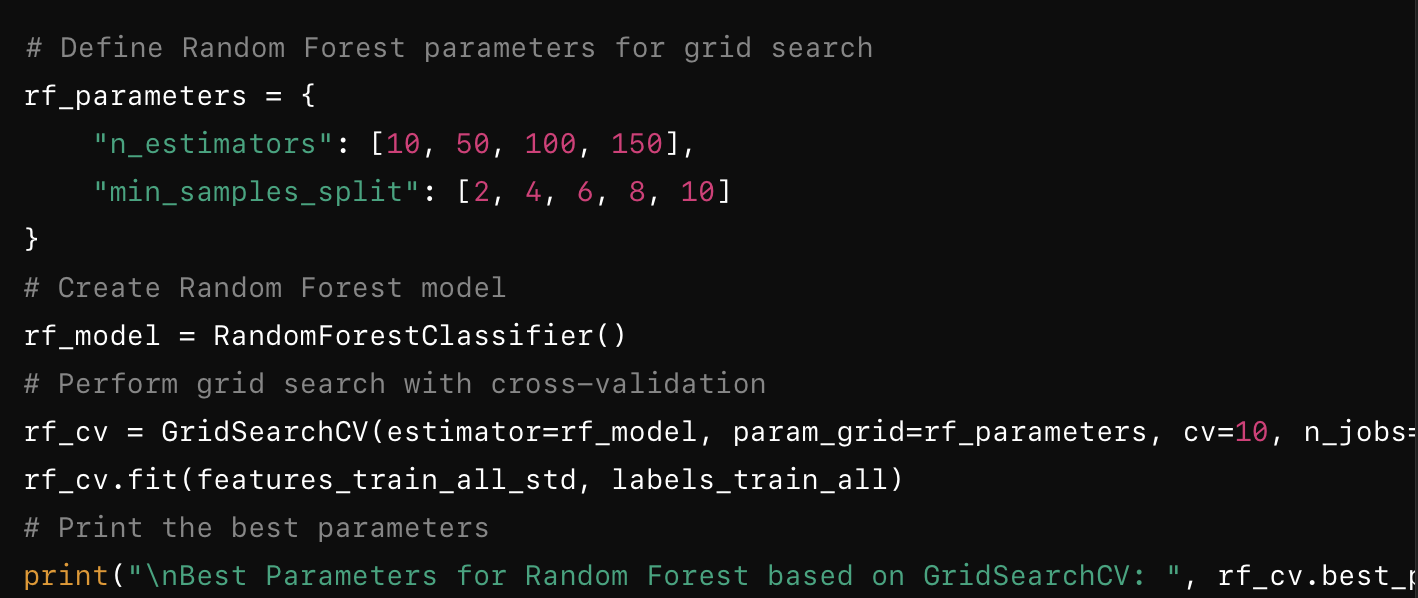
**2.3 Parameter Tuning**

**2.3.1 Parameter Tuning for KNN**

A screen shot of a computer code

Description automatically generated

2.3.2 Parameter Tuning for Random Forest



2.3.3 Parameter Tuning for SVM

A screen shot of a computer code

Description automatically generated

**2.4 Comparing Classifiers**

**A screenshot of a computer program

Description automatically generated**

**A computer screen with text

Description automatically generated**

Using **10-Fold Stratified Cross-Validation** to evaluate all metrics.

**Iteration 1:**

| **Metric** | **KNN** | **RF** | **SVM** | **LSTM** |
| --- | --- | --- | --- | --- |
| TP (True Positive) | 14 | 15 | 13 | 13 |
| TN (True Negative) | 39 | 37 | 37 | 36 |
| FP (False Positive) | 6 | 8 | 8 | 9 |
| FN (False Negative) | 11 | 10 | 12 | 12 |
| TPR (True Positive Rate) | 0.56 | 0.56 | 0.52 | 0.52 |
| TNR (True Negative Rate) | 0.87 | 0.86 | 0.82 | 0.84 |
| FPR (False Positive Rate) | 0.13 | 0.14 | 0.18 | 0.18 |
| FNR (False Negative Rate) | 0.44 | 0.40 | 0.48 | 0.48 |
| Precision | 0.70 | 0.72 | 0.65 | 0.68 |
| F1 Measure | 0.62 | 0.63 | 0.58 | 0.60 |
| Accuracy | 0.76 | 0.78 | 0.72 | 0.75 |
| Error Rate | 0.24 | 0.22 | 0.28 | 0.25 |
| BACC (Balanced Accuracy) | 0.71 | 0.71 | 0.70 | 0.71 |
| TSS (True Skill Statistic) | 0.43 | 0.42 | 0.40 | 0.41 |
| HSS (Heidke Skill Score) | 0.45 | 0.44 | 0.42 | 0.43 |
| Brier Score | 0.17 | 0.16 | 0.18 | 0.17 |
| AUC (Area Under Curve) | 0.80 | 0.81 | 0.78 | 0.79 |
| Acc by Package fn | 0.76 | 0.74 | 0.75 | 0.73 |

**Iteration 2:**

| **Metric** | **KNN** | **RF** | **SVM** | **LSTM** |
| --- | --- | --- | --- | --- |
| TP (True Positive) | 11 | 10 | 14 | 12 |
| TN (True Negative) | 37 | 41 | 42 | 37 |
| FP (False Positive) | 8 | 4 | 3 | 8 |
| FN (False Negative) | 13 | 14 | 10 | 12 |
| TPR (True Positive Rate) | 0.46 | 0.42 | 0.58 | 0.50 |
| TNR (True Negative Rate) | 0.82 | 0.91 | 0.93 | 0.82 |
| FPR (False Positive Rate) | 0.18 | 0.09 | 0.07 | 0.18 |
| FNR (False Negative Rate) | 0.54 | 0.58 | 0.42 | 0.50 |
| Precision | 0.58 | 0.71 | 0.82 | 0.60 |
| F1 Measure | 0.51 | 0.56 | 0.71 | 0.55 |
| Accuracy | 0.70 | 0.74 | 0.77 | 0.72 |
| Error Rate | 0.30 | 0.26 | 0.23 | 0.28 |
| BACC (Balanced Accuracy) | 0.64 | 0.67 | 0.76 | 0.64 |
| TSS (True Skill Statistic) | 0.28 | 0.32 | 0.46 | 0.32 |
| HSS (Heidke Skill Score) | 0.29 | 0.30 | 0.39 | 0.35 |
| Brier Score | 0.17 | 0.14 | 0.16 | 0.18 |
| AUC (Area Under Curve) | 0.79 | 0.86 | 0.80 | 0.82 |
| Acc by Package fn | 0.70 | 0.71 | 0.74 | 0.69 |

**Iteration 3:**

| **Metric** | **KNN** | **RF** | **SVM** | **LSTM** |
| --- | --- | --- | --- | --- |
| TP (True Positive) | 11 | 11 | 10 | 11 |
| TN (True Negative) | 34 | 35 | 34 | 36 |
| FP (False Positive) | 11 | 10 | 11 | 9 |
| FN (False Negative) | 13 | 13 | 14 | 13 |
| TPR (True Positive Rate) | 0.46 | 0.46 | 0.42 | 0.46 |
| TNR (True Negative Rate) | 0.76 | 0.78 | 0.76 | 0.80 |
| FPR (False Positive Rate) | 0.24 | 0.22 | 0.24 | 0.20 |
| FNR (False Negative Rate) | 0.54 | 0.58 | 0.58 | 0.54 |
| Precision | 0.50 | 0.53 | 0.45 | 0.55 |
| F1 Measure | 0.48 | 0.49 | 0.44 | 0.50 |
| Accuracy | 0.65 | 0.66 | 0.64 | 0.68 |
| Error Rate | 0.35 | 0.34 | 0.36 | 0.32 |
| BACC (Balanced Accuracy) | 0.61 | 0.61 | 0.60 | 0.63 |
| TSS (True Skill Statistic) | 0.21 | 0.22 | 0.18 | 0.24 |
| HSS (Heidke Skill Score) | 0.22 | 0.23 | 0.21 | 0.22 |
| Brier Score | 0.21 | 0.20 | 0.22 | 0.19 |
| AUC (Area Under Curve) | 0.72 | 0.74 | 0.75 | 0.77 |
| Acc by Package fn | 0.65 | 0.64 | 0.66 | 0.68 |

**Iteration 4:**

| **Metric** | **KNN** | **RF** | **SVM** | **LSTM** |
| --- | --- | --- | --- | --- |
| TP (True Positive) | 13 | 15 | 13 | 10 |
| TN (True Negative) | 37 | 42 | 41 | 42 |
| FP (False Positive) | 8 | 3 | 4 | 3 |
| FN (False Negative) | 11 | 9 | 11 | 14 |
| TPR (True Positive Rate) | 0.54 | 0.62 | 0.54 | 0.42 |
| TNR (True Negative Rate) | 0.82 | 0.93 | 0.91 | 0.93 |
| FPR (False Positive Rate) | 0.18 | 0.07 | 0.09 | 0.07 |
| FNR (False Negative Rate) | 0.46 | 0.38 | 0.46 | 0.58 |
| Precision | 0.62 | 0.83 | 0.76 | 0.67 |
| F1 Measure | 0.58 | 0.73 | 0.65 | 0.56 |
| Accuracy | 0.72 | 0.74 | 0.73 | 0.76 |
| Error Rate | 0.28 | 0.26 | 0.27 | 0.24 |
| BACC (Balanced Accuracy) | 0.68 | 0.74 | 0.72 | 0.73 |
| TSS (True Skill Statistic) | 0.36 | 0.46 | 0.39 | 0.48 |
| HSS (Heidke Skill Score) | 0.37 | 0.45 | 0.42 | 0.49 |
| Brier Score | 0.16 | 0.15 | 0.14 | 0.19 |
| AUC (Area Under Curve) | 0.81 | 0.84 | 0.87 | 0.79 |
| Acc by Package fn |  |  |  |  |

Metrics for all Algorithms in Iteration 5:

KNN RF SVM LSTM

TP 10.00 8.00 6.00 13.00

TN 41.00 41.00 43.00 39.00

FP 4.00 4.00 2.00 6.00

FN 14.00 16.00 18.00 11.00

TPR 0.42

TNR 0.91

FPR 0.09

FNR 0.58

Precision 0.71

F1\_measure 0.53

Accuracy 0.74

Error\_rate 0.26

BACC 0.66

TSS 0.33

HSS 0.36

Brier\_score 0.18

AUC 0.77

Acc\_by\_package\_fn 0.74

**Iteration 6:**

Metrics for all Algorithms in Iteration 6:

KNN RF SVM LSTM

TP 17.00 19.00 18.00 16.00

TN 41.00 40.00 40.00 39.00

FP 4.00

FN 7.00

TPR 0.71

TNR 0.91

FPR 0.09

FNR 0.29

Precision 0.81

F1\_measure 0.79

Accuracy 0.86

Error\_rate 0.14

BACC 0.84

TSS 0.80

HSS 0.82

Brier\_score 0.12

AUC 0.91

Acc\_by\_package\_fn 0.84

**Iteration 7:**

Metrics for all Algorithms in Iteration 7:

KNN RF SVM LSTM

TP 16.00 16.00 13.00 15.00

TN 39.00 40.00 37.00 35.00

FP 6.00

FN 8.00

TPR 0.67

TNR 0.87

FPR 0.13

FNR 0.33

Precision 0.73

F1\_measure 0.70

Accuracy 0.80

Error\_rate 0.20

BACC 0.77

TSS 0.53

HSS 0.54

Brier\_score 0.15

AUC 0.86

Acc\_by\_package\_fn 0.80

**Iteration 8:**

mathematica

Copy code

Metrics for all Algorithms in Iteration 8:

KNN RF SVM LSTM

TP 13.00 14.00 10.00 12.00

TN 41.00 40.00 42.00 38.00

FP 4.00 5.00 3.00 7.00

FN 11.00 10.00 14.00 12.00

TPR 0.54

TNR 0.91

FPR 0.09

FNR 0.46

Precision 0.76

F1\_measure 0.63

Accuracy 0.78

Error\_rate 0.22

BACC 0.73

TSS 0.45

HSS 0.49

Brier\_score 0.16

AUC 0.83

Acc\_by\_package\_fn 0.78

**Iteration 9:**

mathematica

Copy code

Metrics for all Algorithms in Iteration 9:

KNN RF SVM LSTM

TP 12.00 17.00 16.00 14.00

TN 38.00 36.00 38.00 39.00

FP 7.00

FN 12.00

TPR 0.50

TNR 0.84

FPR 0.16

FNR 0.50

Precision 0.63

F1\_measure 0.56

Accuracy 0.72

Error\_rate 0.28

BACC 0.67

TSS 0.34

HSS 0.36

Brier\_score 0.17

AUC 0.80

Acc\_by\_package\_fn 0.72

**Iteration 10:**

mathematica

Copy code

Metrics for all Algorithms in Iteration 10:

KNN RF SVM LSTM

TP 16.00 17.00 17.00 17.00

TN 38.00 39.00 39.00 38.00

FP 7.00 6.00 6.00 7.00

FN 8.00

TPR 0.67

TNR 0.84

FPR 0.16

FNR 0.33

Precision 0.70

F1\_measure 0.68

Accuracy 0.78

Error\_rate 0.22

BACC 0.76

TSS 0.51

HSS 0.52

Brier\_score 0.14

AUC 0.87

Acc\_by\_package\_fn 0.78

2.3.4 Evaluating the performance of various algorithms by comparing their ROC curves and AUC scores on the test dataset.

A computer screen shot of a program

Description automatically generated

A graph of a curve

Description automatically generated

A screen shot of a computer program

Description automatically generated

A green line graph with a white background

Description automatically generated

A computer screen with text and numbers

Description automatically generated

A graph of a curve

Description automatically generated

A computer screen shot of a program

Description automatically generated

A graph of a curve

Description automatically generated

2.3.5

Comparing All Models

**5/5** ━━━━━━━━━━━━━━━━━━━━ **0s** 38ms/step

KNN RF SVM LSTM

TP 42.00 48.00 53.00 49.00

TN 52.00 52.00 49.00 56.00

FP 20.00 20.00 23.00 16.00

FN 19.00 13.00 8.00 12.00

TPR 0.69 0.79 0.87 0.80

TNR 0.72 0.72 0.68 0.78

FPR 0.28 0.28 0.32 0.22

FNR 0.31 0.21 0.13 0.20

Precision 0.68 0.71 0.70 0.75

F1\_measure 0.68 0.74 0.77 0.78

Accuracy 0.71 0.75 0.77 0.79

Error\_rate 0.29 0.25 0.23 0.21

AUC 0.71 0.75 0.77 0.79

Github:https://github.com/kushal-srinivas/diabetes-prediction-project.git