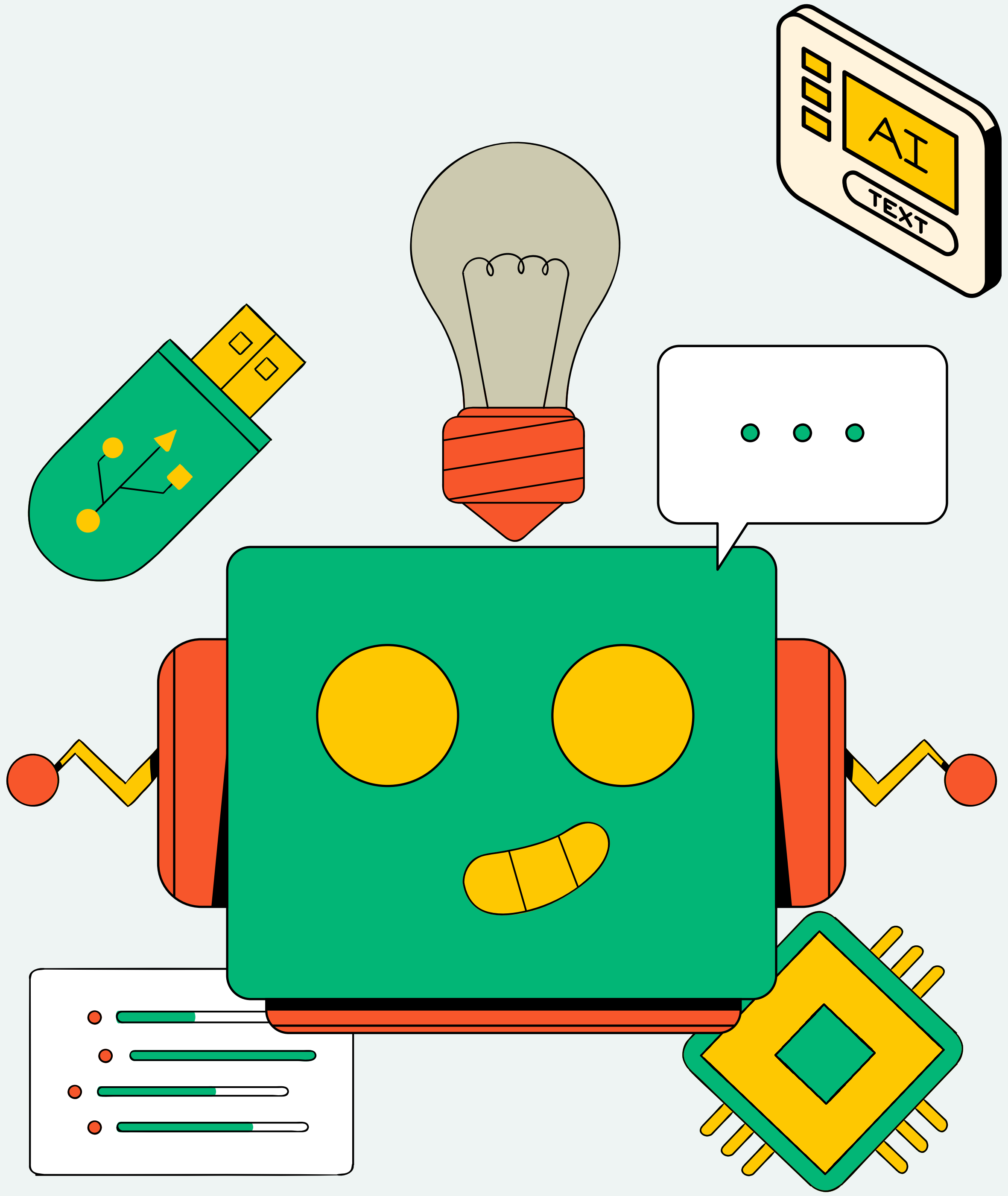
THYNK UNLIMITED

WE LEARN FOR THE FUTURE

DIABETES PREDICTION USING MACHINE LEARNING



PRESENTED BY:

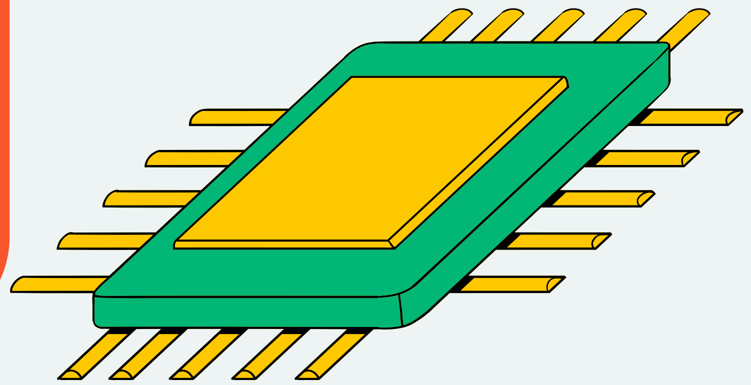
VINIT VIJAYKUMAR ADKE

YASH SANJAYKUMAR PARDESHI

DHIRRAJ SURESH KUMAR

JASWANTH GIRIDHARAGOPALAN

VINEETH KRISHNA



PRESENTATION OUTLINE

- Business Problem
- Data Understanding
- Data Preparation
- Modelling
- Evaluation
- Deployment

BUSINESS PROBLEM

Diabetes affects over 500M people worldwide & the nos keep increasing. However, early diagnosis remains uncommon, as it often relies on reactive testing once symptoms appear.

Our Solution:
We propose a machine learning- based diagnostic tool that predicts diabetes risk using basic health indicators(such as glucose levels, BMI, Age and BP). The model enables preventive care rather than reactive treatment.

Value to Stakeholders:
Doctors & Healthcare Providers: Detect at-risk patients, enable personalized interventions. Reduce unnecessary testings. Improve diagnostic accuracy.

Patients & Caregivers: Receive personalized alerts. Feel more informed, proactive in managing their health

Why it mattes:
Clinical Impact: Reduce complications like heart disease, kidney failure.
Economic Value: Early detection cuts long-term healthcare cost
Scalability: With minimal input data, this model can be integrated into routine screenings globally.

WHY USE DATA MINING?

1. Early Risk Prediction
2. Evidence Based Decision Making
3. Personlized Healthcare
4. Resource Optimization
5. Scalability and Accessibility

DATA UNDERSTANDING

Dataset:
It contains dignostic health info from female patients aged 21 and above

Target Variable: 0 – No diabetes , 1 – Diabetes

Input Features:
Pregnancies – Number of times the patient has been pregnant
Glucose – Plasma glucose concentration
Blood Pressure – Diastolic blood pressure measurement
Skin Thickness– Triceps skinfold thickness
Insulin – 2-hour post-load serum insulin level
BMI – Body Mass Index (weight relative to height)
Diabetes Pedigree Function– Score indicating genetic predisposition to diabetes
Age – Patient's age in years

Pre-processing Steps:
1. **Handling Missing Values**
2. **Feature Scaling**
3. **Train-Test Split**

MODEL EXPLORATION

Decision Tree

Logistic Regression

SVM

Neural Network

Naive Bayes

Ensemble

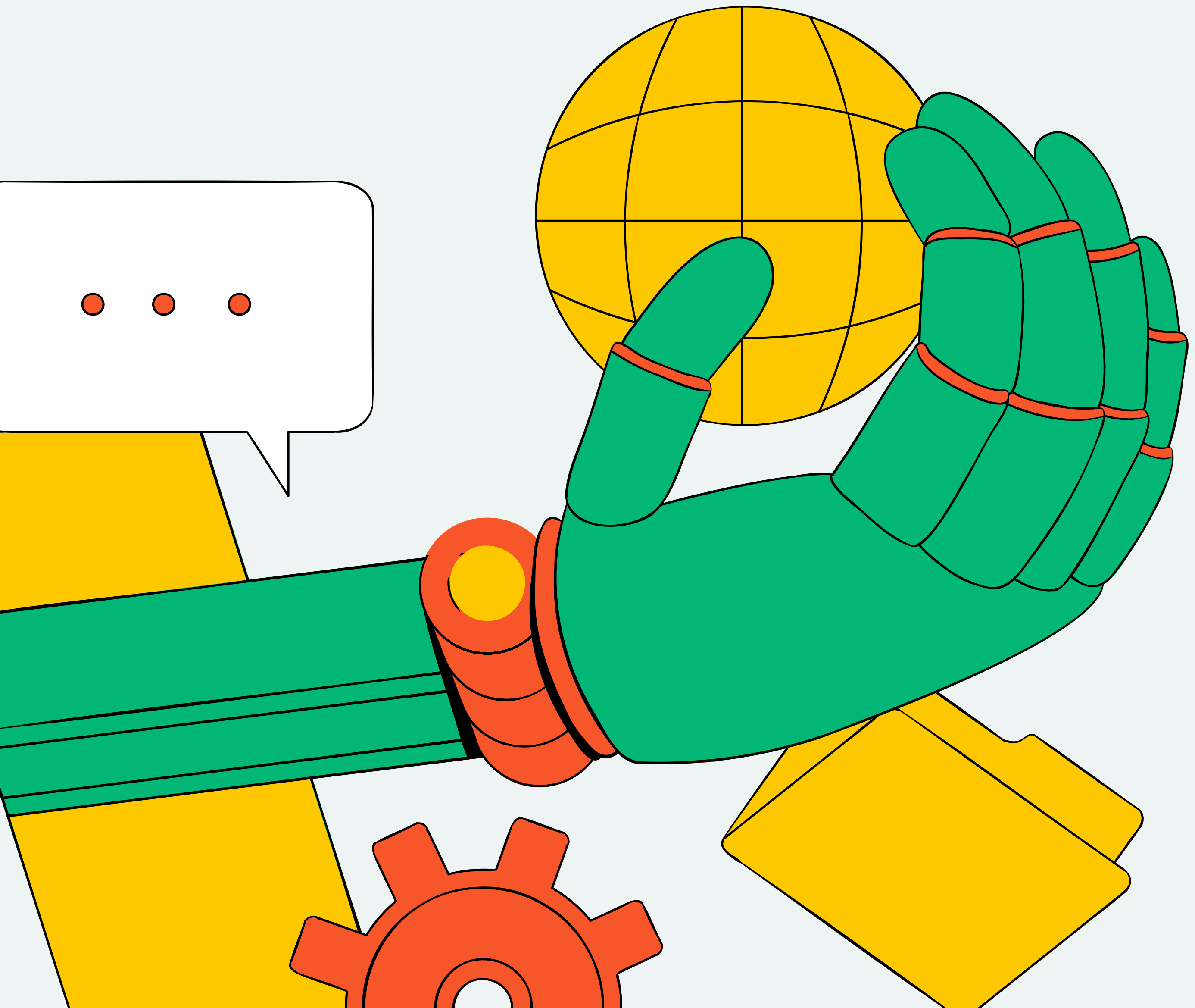
Random Forest

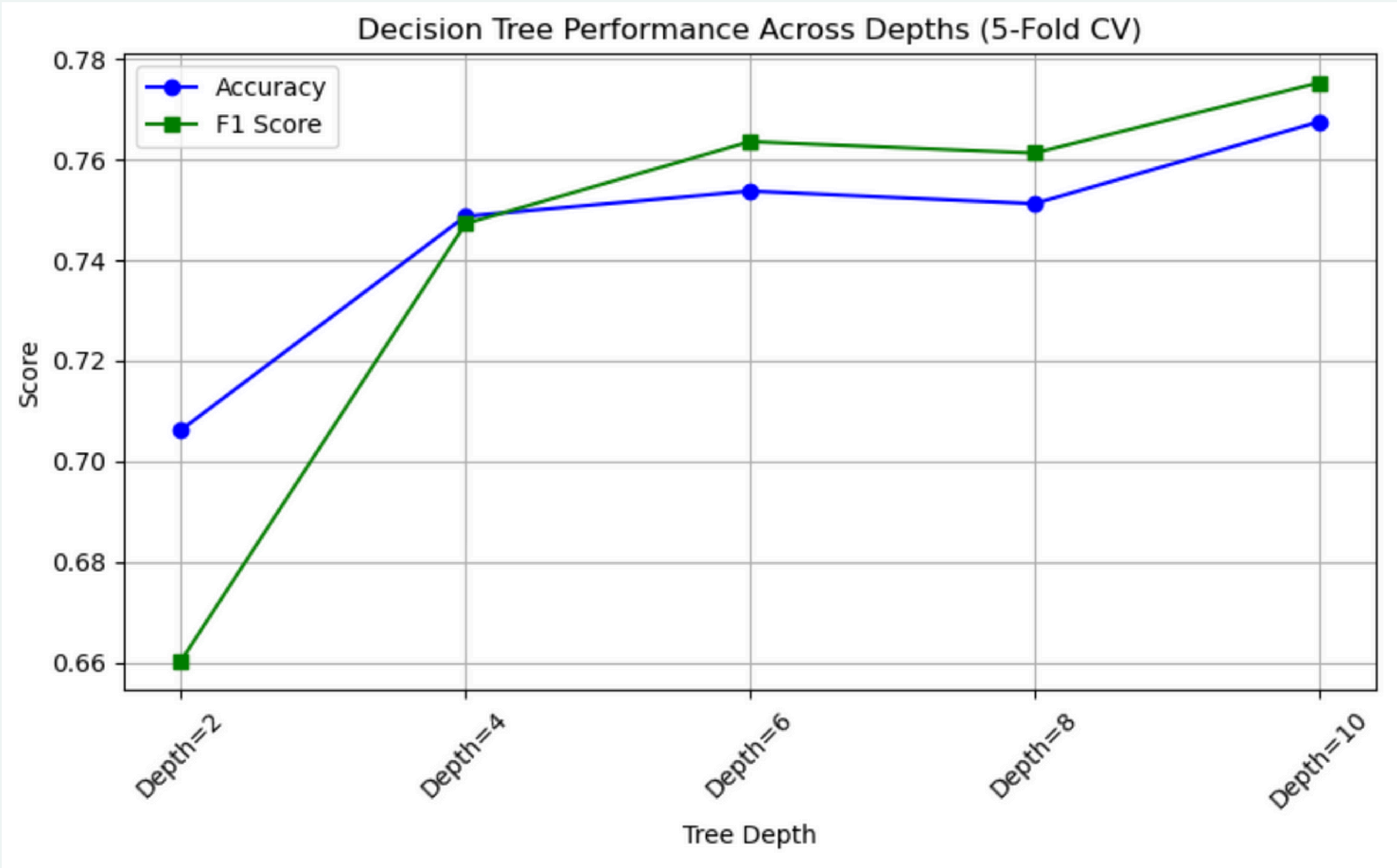
XGBoost

KNN

Hierarchial Clustering

DECISION TREE

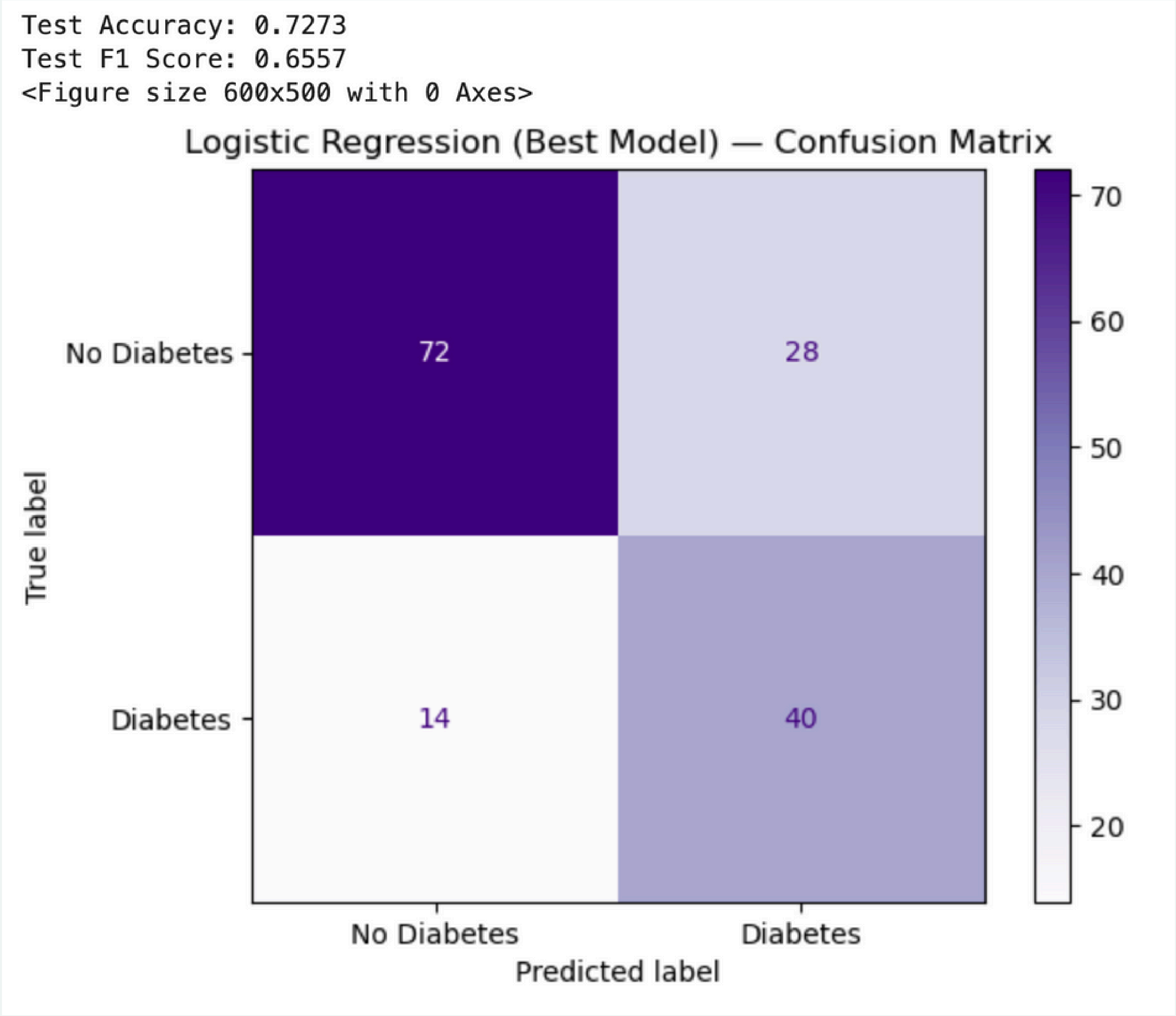


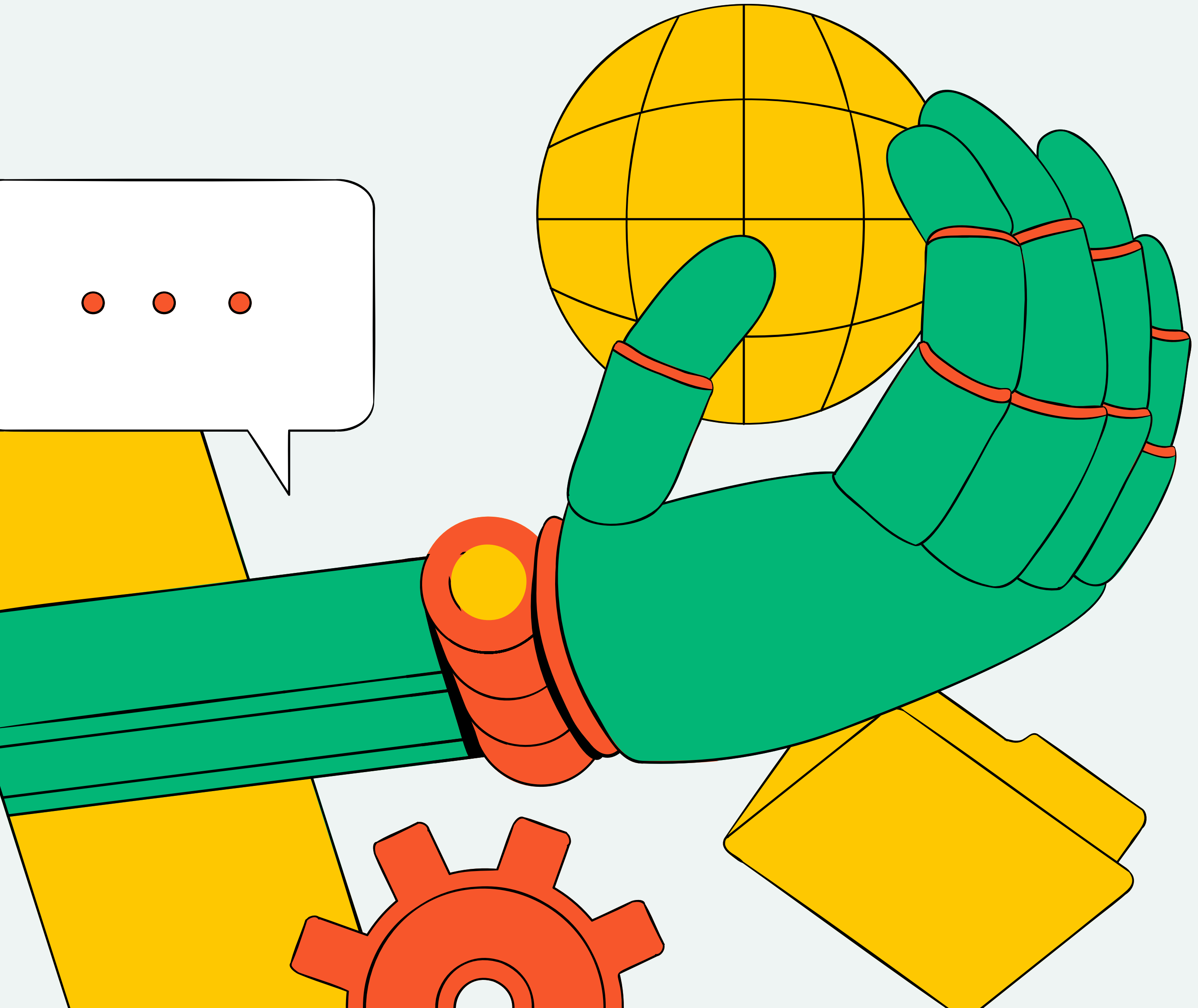


Performance stabilizes around depth 6 to 10
Depth 10 has the highest F1 score

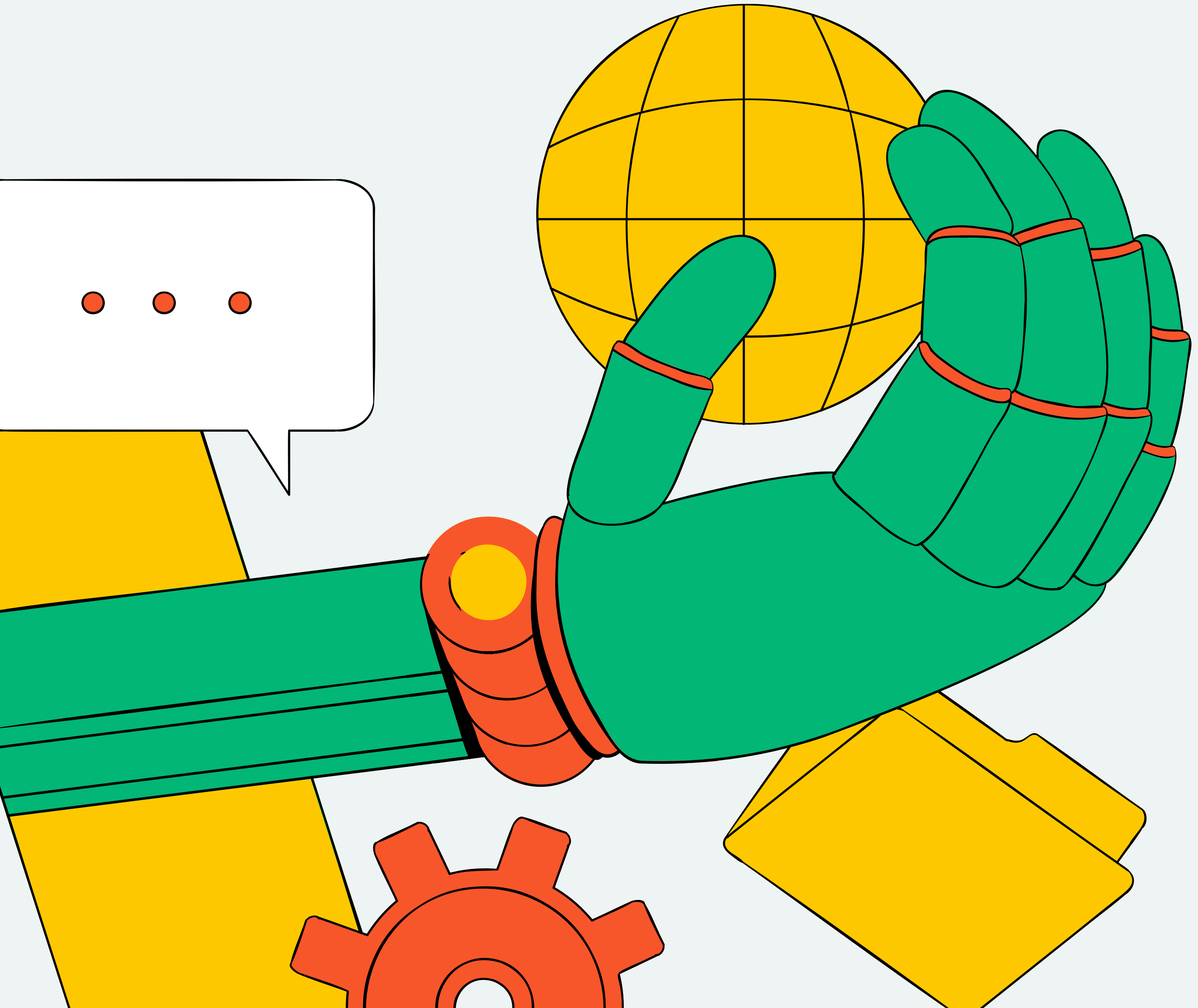
LOGISTIC REGRESSION

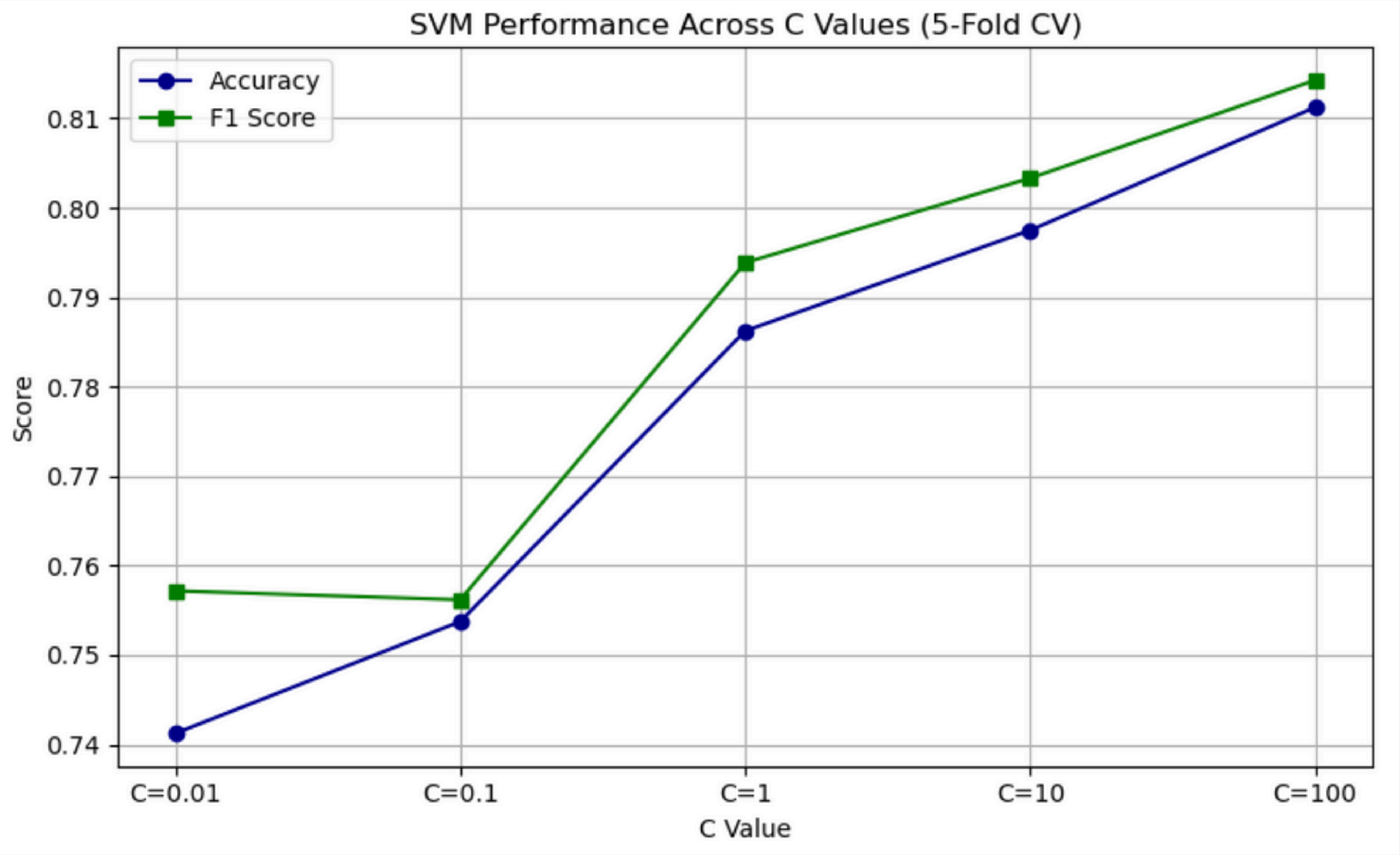
Backward Selected Features: ['Pregnancies', 'Glucose', 'Insulin', 'BMI', 'DiabetesPedigreeFunction']





SUPPORT VECTOR MACHINE





• C = 1 provided the best overall balance between accuracy and generalization.

• Very low C (0.01) led to severe underfitting, resulting in high bias and poor classification.

• Very high C (100) led to overfitting and poorer generalization on unseen data.

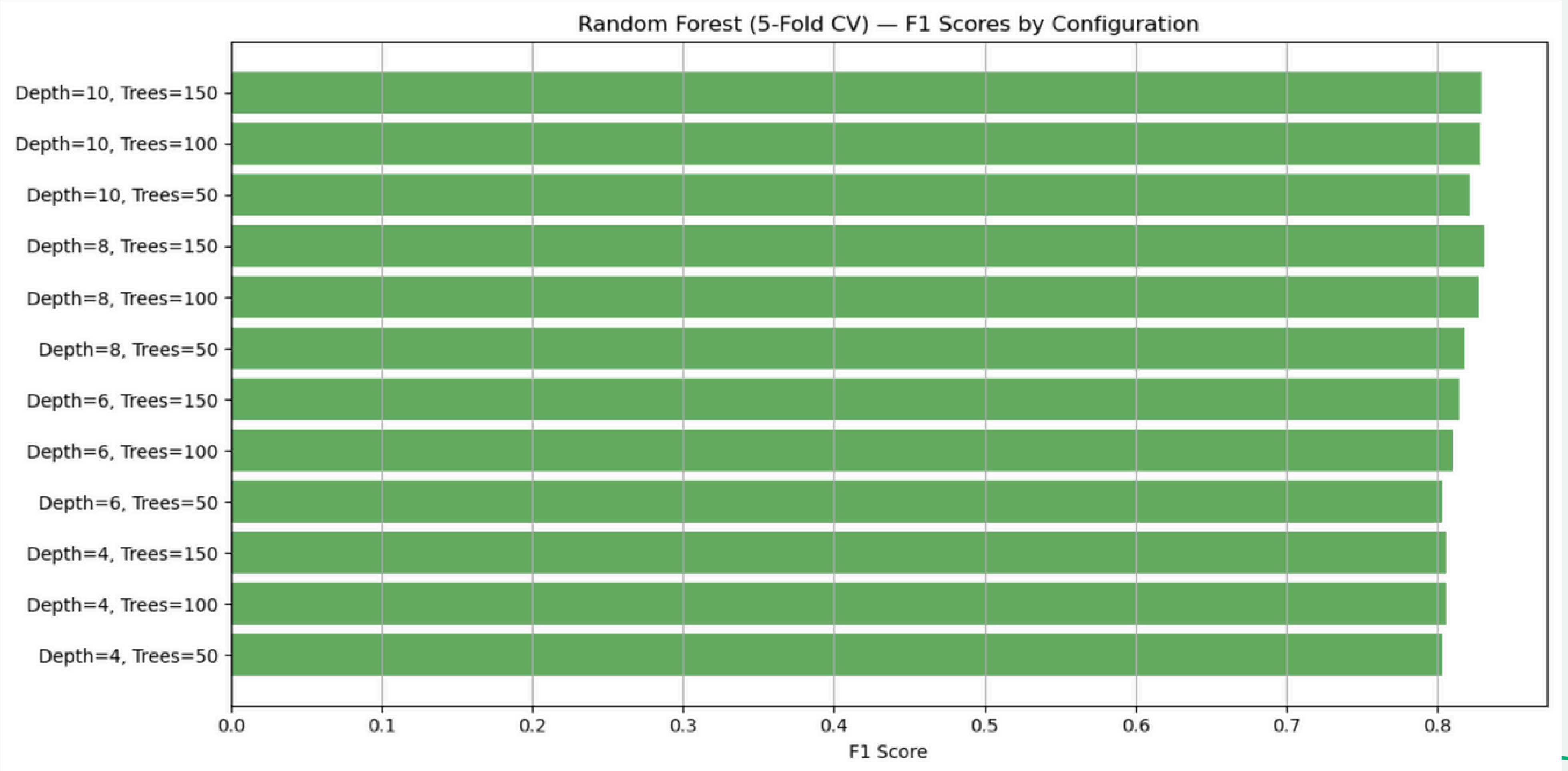
•C= 1 with RBF kernel selected as the final model configuration. Achieved the highest mean cross-validation accuracy (~76%).

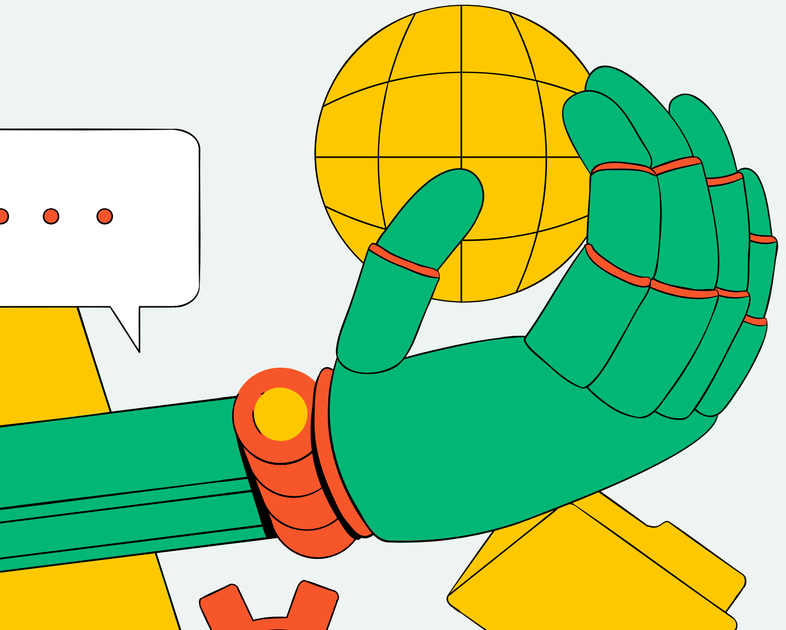
RANDOM FOREST

Random Forest (5-Fold CV) Results:

Configuration	Accuracy	F1 Score
Depth=8, Trees=150	0.8213	0.8314
Depth=10, Trees=150	0.8225	0.8299
Depth=10, Trees=100	0.8213	0.8291
Depth=8, Trees=100	0.8187	0.8282
Depth=10, Trees=50	0.8125	0.8213
Depth=8, Trees=50	0.8087	0.8179
Depth=6, Trees=150	0.8037	0.8146
Depth=6, Trees=100	0.7987	0.8103
Depth=4, Trees=150	0.7950	0.8061
Depth=4, Trees=100	0.7937	0.8057
Depth=4, Trees=50	0.7913	0.8037
Depth=6, Trees=50	0.7925	0.8036

Depths between 8–10 combined with more trees (100–150) consistently produced the best results.

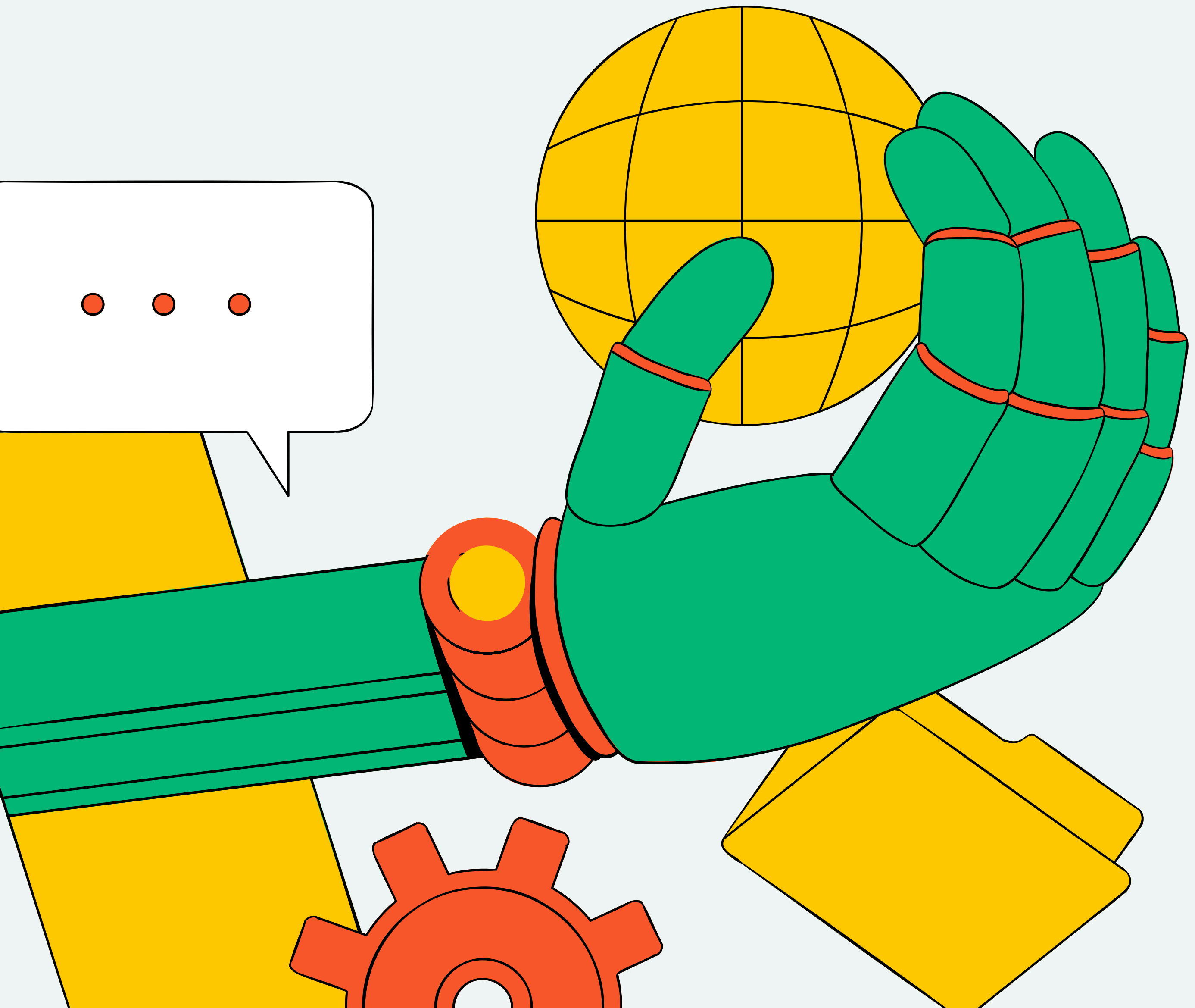


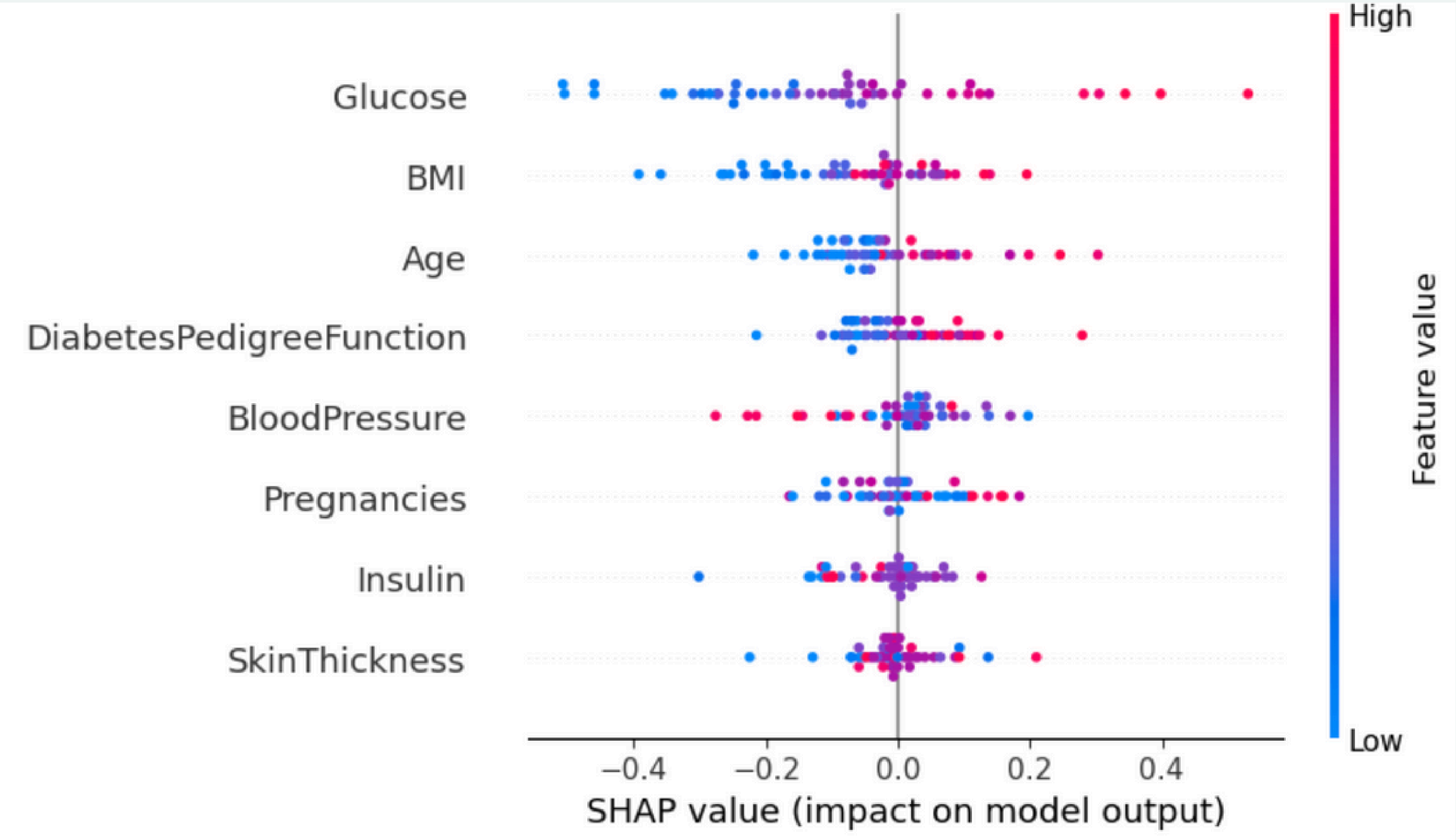


NEURAL NETWORK

Neural Network (5-Fold CV) Results (Sorted by F1 Score):

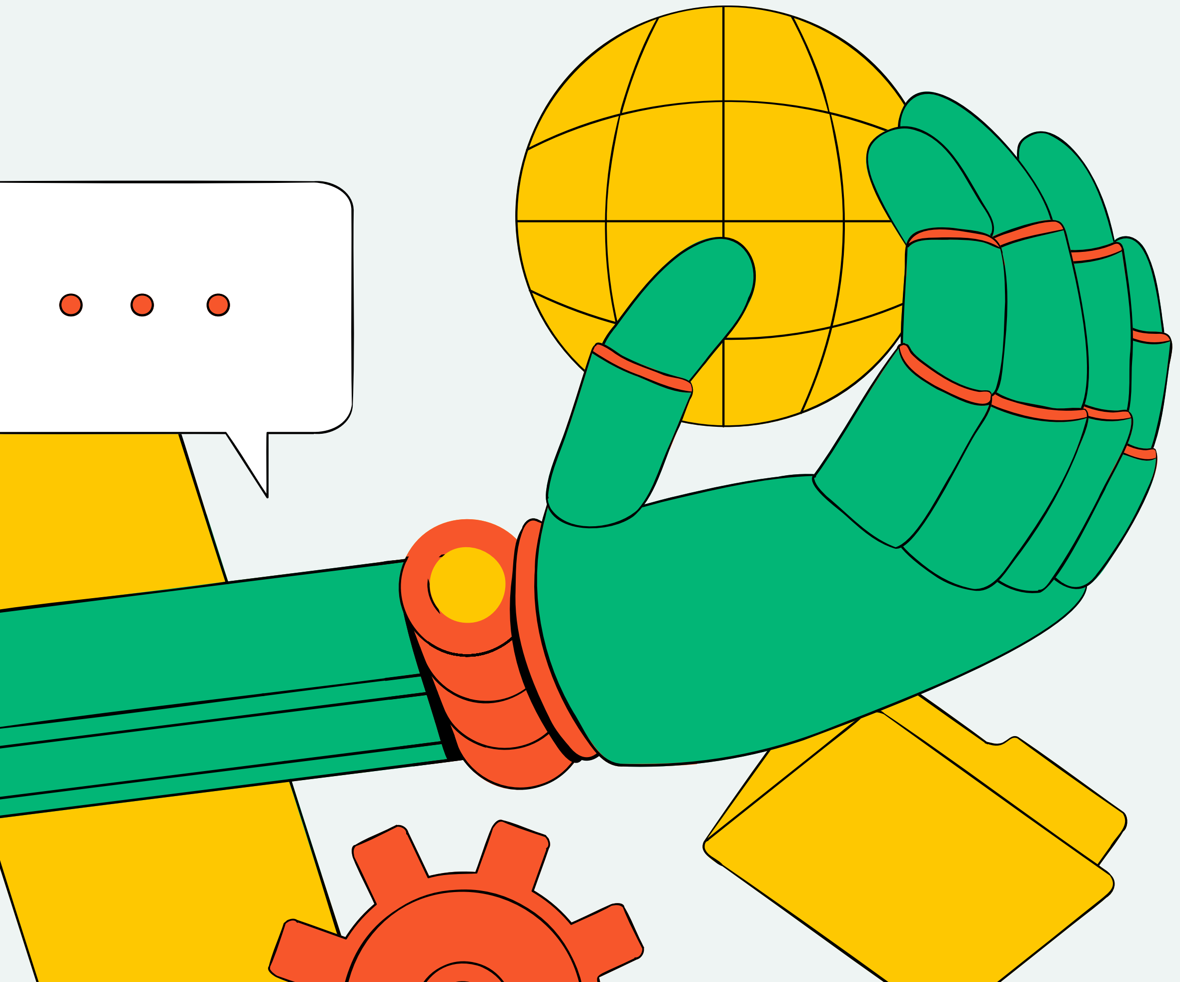
Configuration	Accuracy	F1 Score
Act=relu, Layers=(100, 50, 25)	0.8337	0.8387
Act=tanh, Layers=(100, 50, 25)	0.8250	0.8307
Act=relu, Layers=(50, 50)	0.8213	0.8263
Act=tanh, Layers=(50, 50)	0.8125	0.8221
Act=relu, Layers=(10,)	0.7575	0.7592
Act=tanh, Layers=(10,)	0.7550	0.7586
Act=logistic, Layers=(10,)	0.7450	0.7415
Act=logistic, Layers=(50, 50)	0.7375	0.7396
Act=logistic, Layers=(100, 50, 25)	0.7325	0.7394





ENSEMBLE

```
Model                          Accuracy    F1 Score
-------------------------------------------------------
Voting Different Models        0.7987      0.8022
 → Combination: Decision Tree (Best) + Random Forest (Best) + XGBoost (Best)
-------------------------------------------------------
Voting RF Different Configs    0.8200      0.8304
 → Combination: Random Forest (depth=6, trees=100) + RF (depth=8, trees=150) + RF (depth=10, trees=200)
-------------------------------------------------------
Bagging RF Same Config         0.8100      0.8214
 → Combination: Random Forest (depth=8, trees=150) – 10 Random Samples
-------------------------------------------------------
```

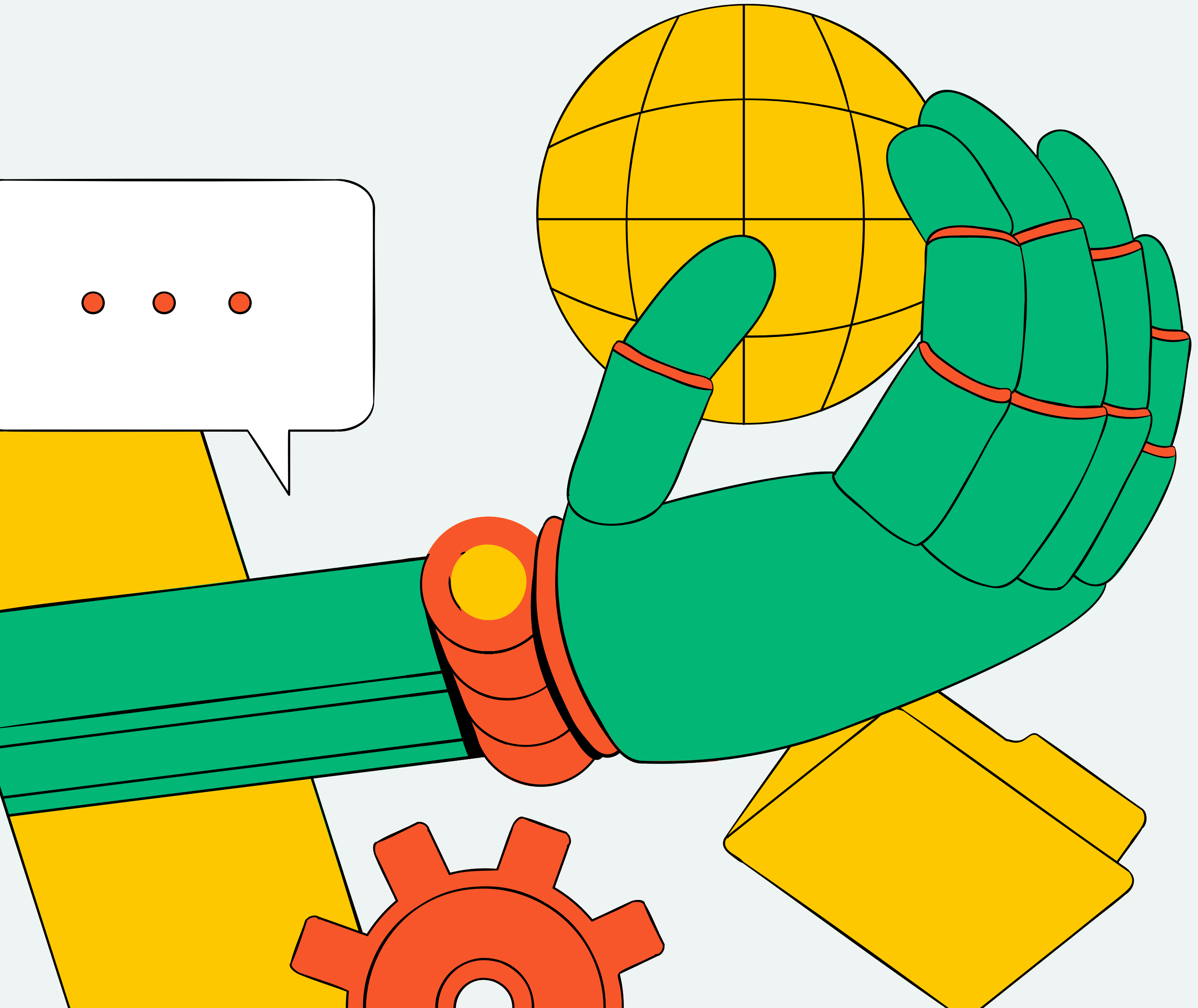


Parameter diversity inside the same model family can yield better calibration and stability.

Bagging aims to reduce variance and stabilize predictions by averaging multiple models trained on varied samples.

NAIVE BAYES

```
Naive Bayes (CV) Results:
Mean Accuracy:  0.75
Mean Precision: 0.67
Mean Recall:    0.58
Mean F1 Score:  0.62
```



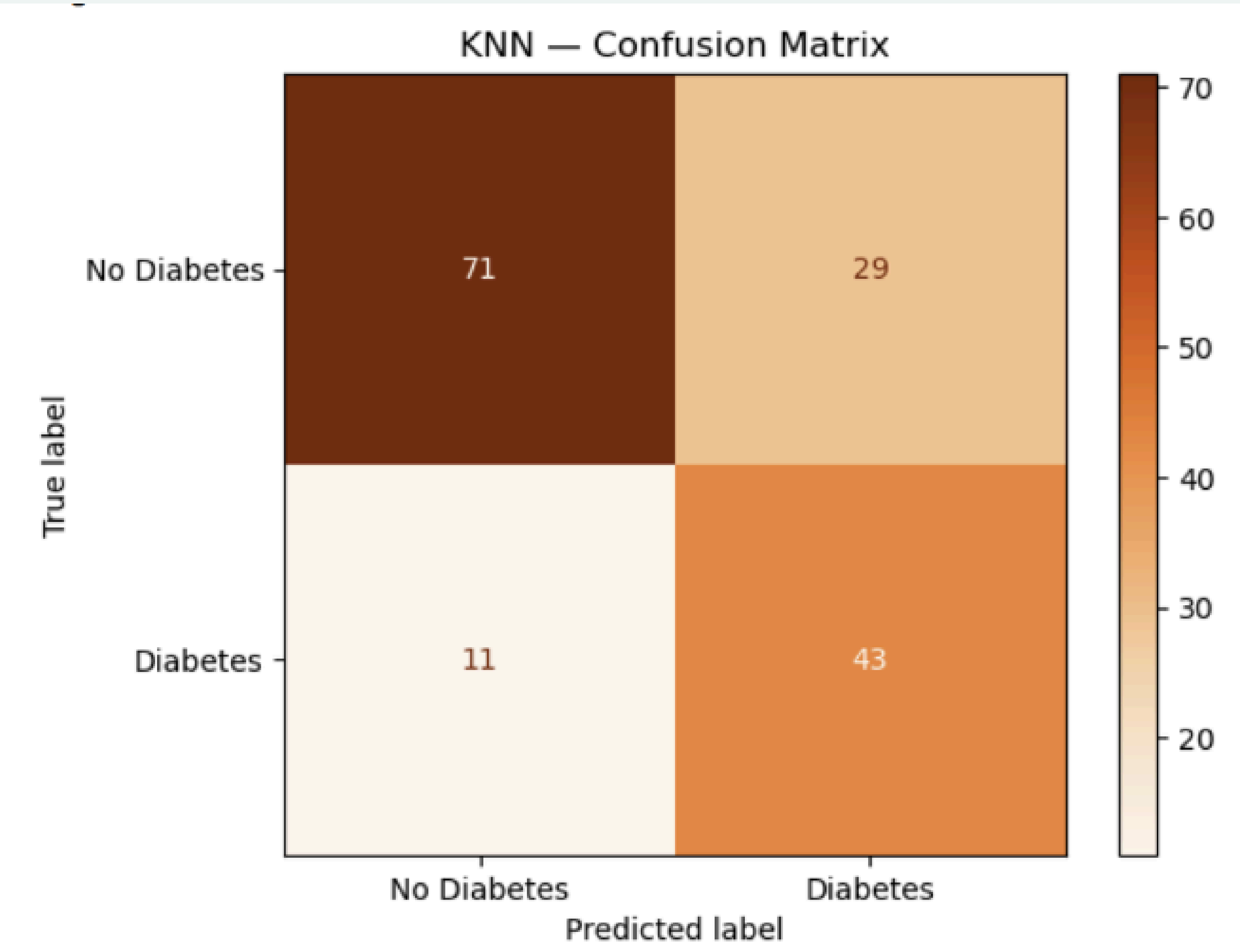
- Performs well despite simplicity — strong baseline performance
- High precision → fewer false positives
- Lower recall → more false negatives, a concern in healthcare
- Very fast to train and easy to interpret

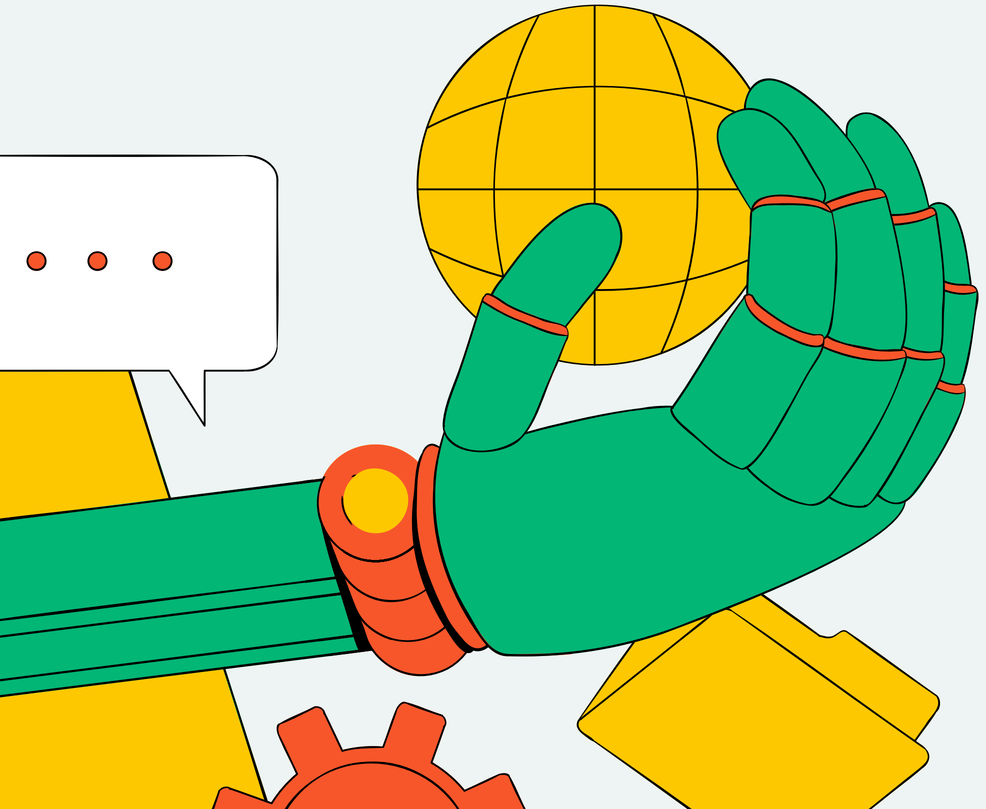
K-NEAREST NEIGHBORS

```
Best Parameters for KNN: {'metric': 'euclidean', 'n_neighbors': 11, 'weights': 'distance'}
Best F1 Score (CV): 0.8377505103692691
Test Accuracy: 0.7403
Test F1 Score: 0.6825
```

Weighting by distance often improves KNN performance especially when some neighbors are much closer than others.

Choosing 11 neighbors gives a good tradeoff b
etween noise smoothing and local decision-making.





XGBOOST MODEL

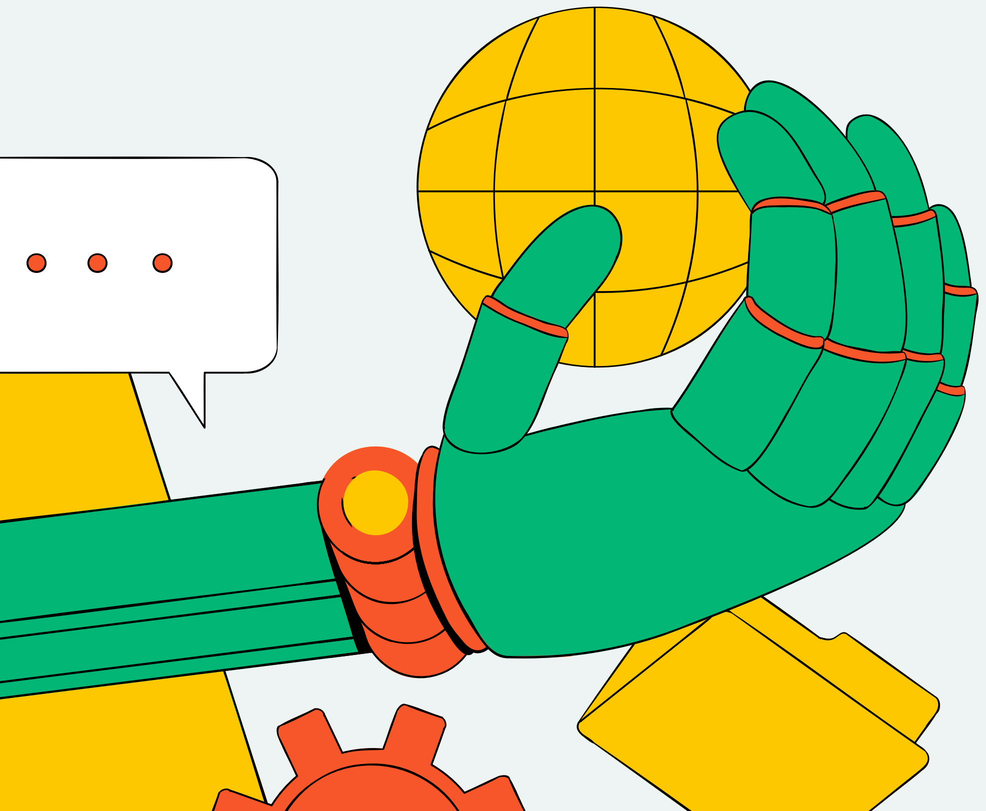
```
Fitting 5 folds for each of 27 candidates, totalling 135 fits
✅ Best Parameters for XGBoost: {'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 150}
✅ Best F1 Score (CV): 0.8168348319012061
```

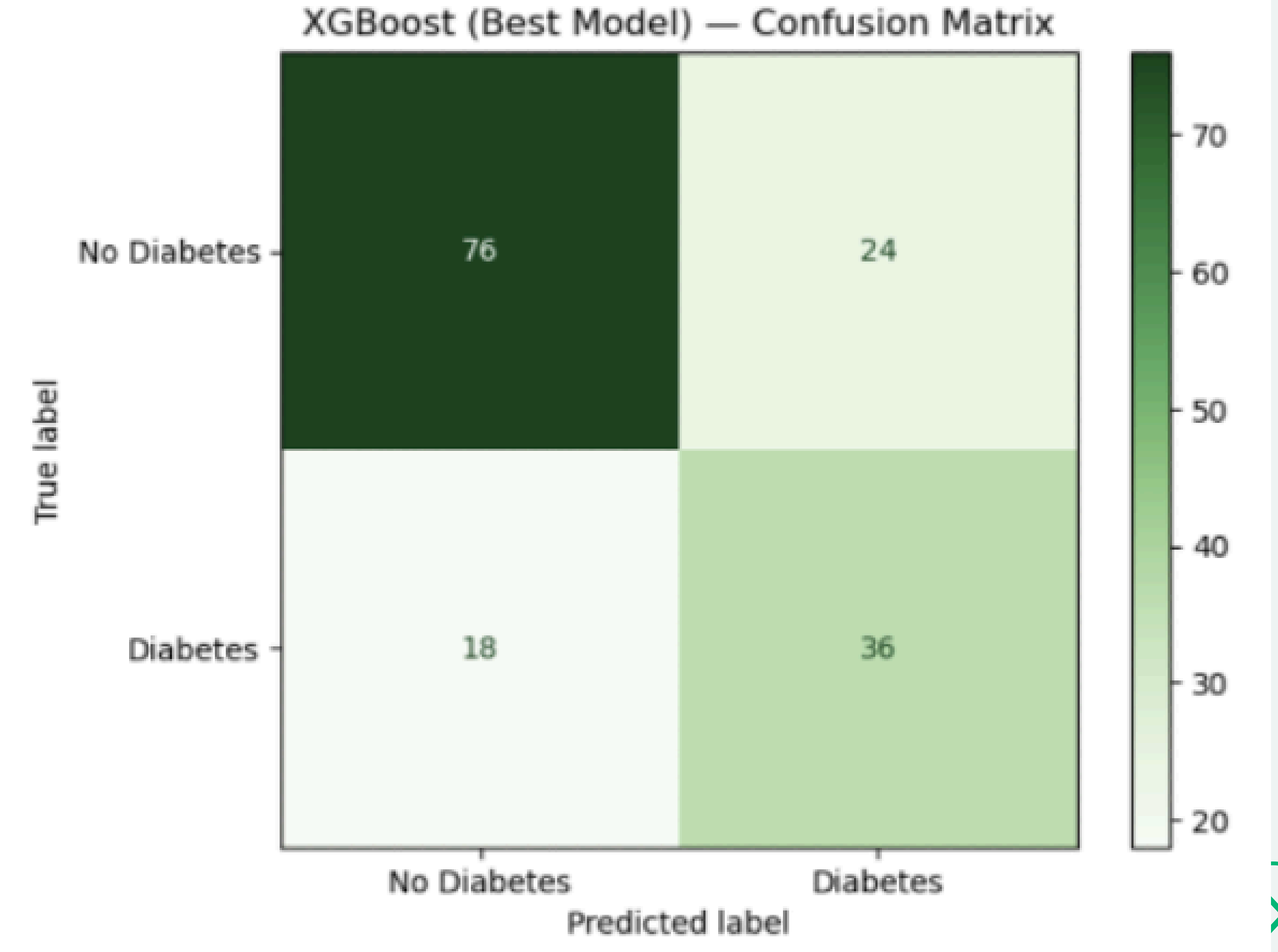
🔧 Parameters Tuned:

- `n_estimators` : [50, 100, 150]
- `learning_rate` : [0.01, 0.1, 0.2]
- `max_depth` : [3, 5, 7]

✅ Best Configuration:

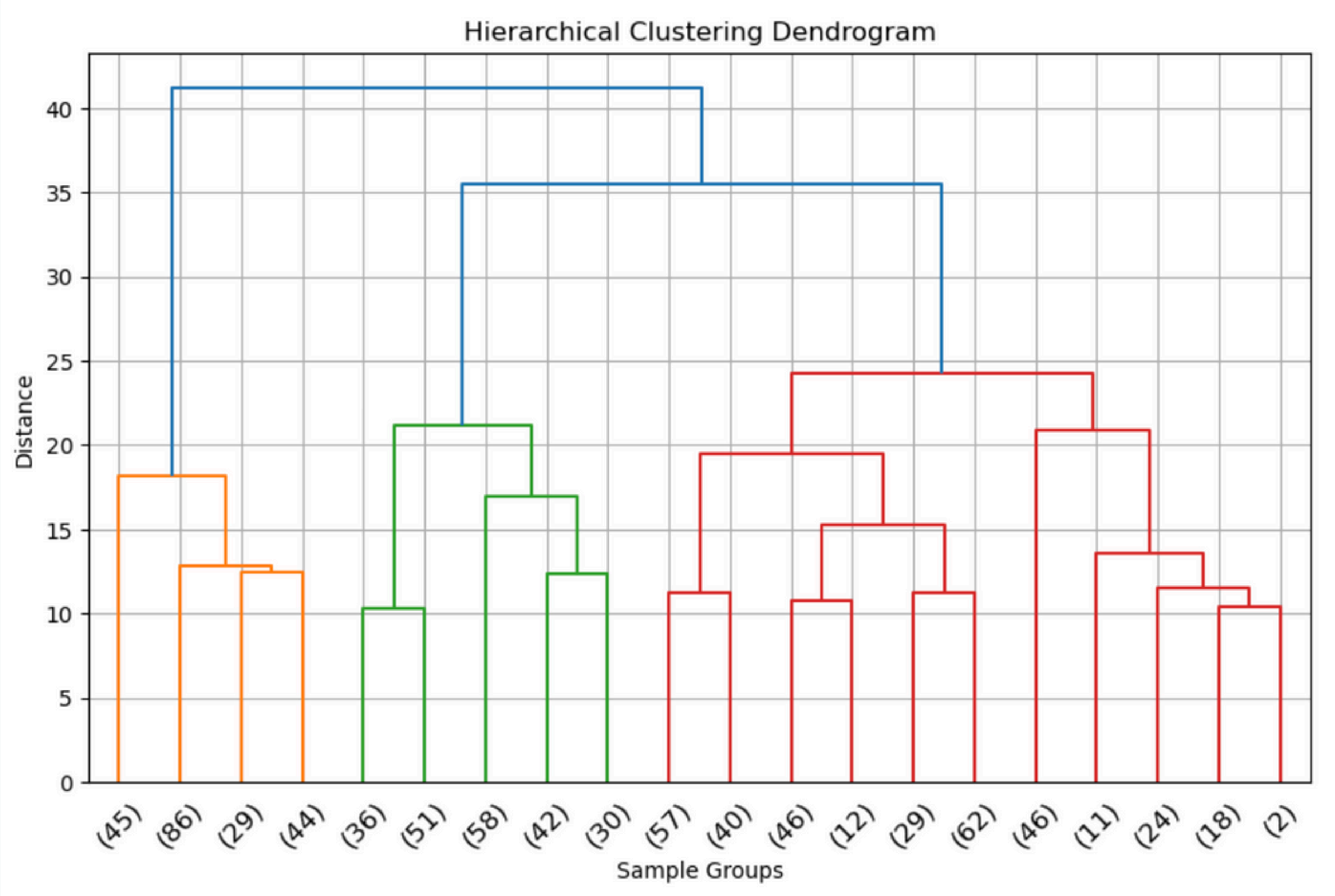
- `n_estimators` : 150
- `learning_rate` : 0.1
- `max_depth` : 7

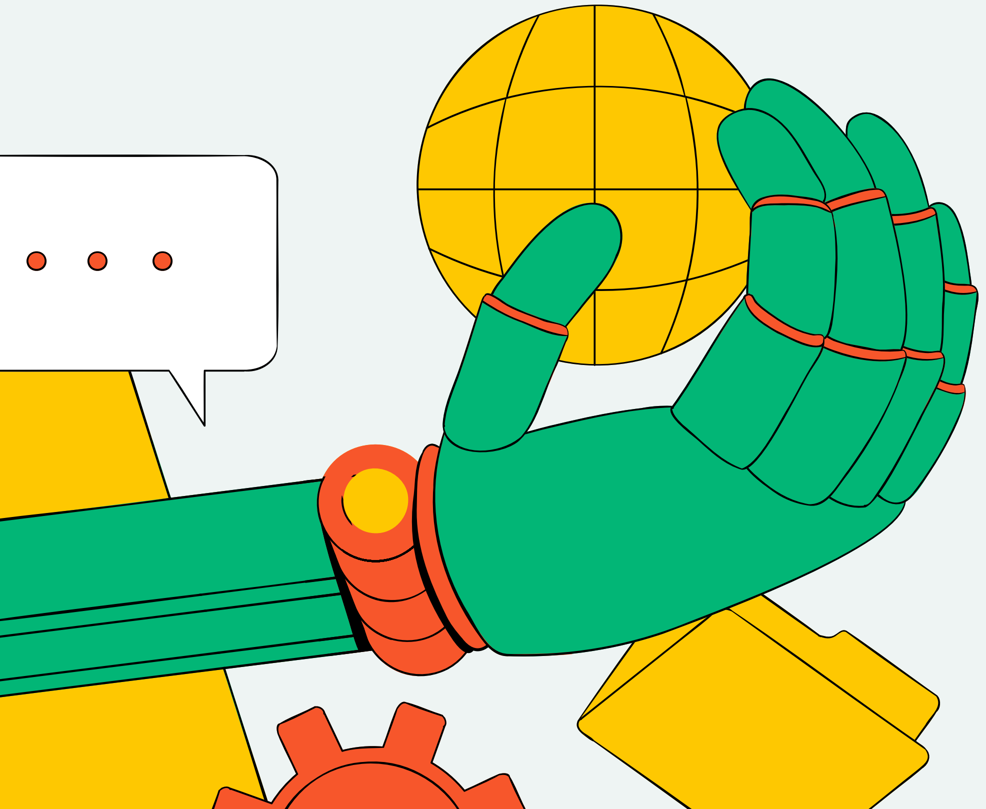




HIERARCHICAL CLUSTERING

- The dendrogram shows clear separation into 3–4 main clusters.
- These clusters may represent distinct patient profiles:
  - High-risk diabetics with elevated glucose, BMI, and age.
  - Younger, low-BMI individuals with low glucose levels.
  - Borderline or intermediate-risk patients.

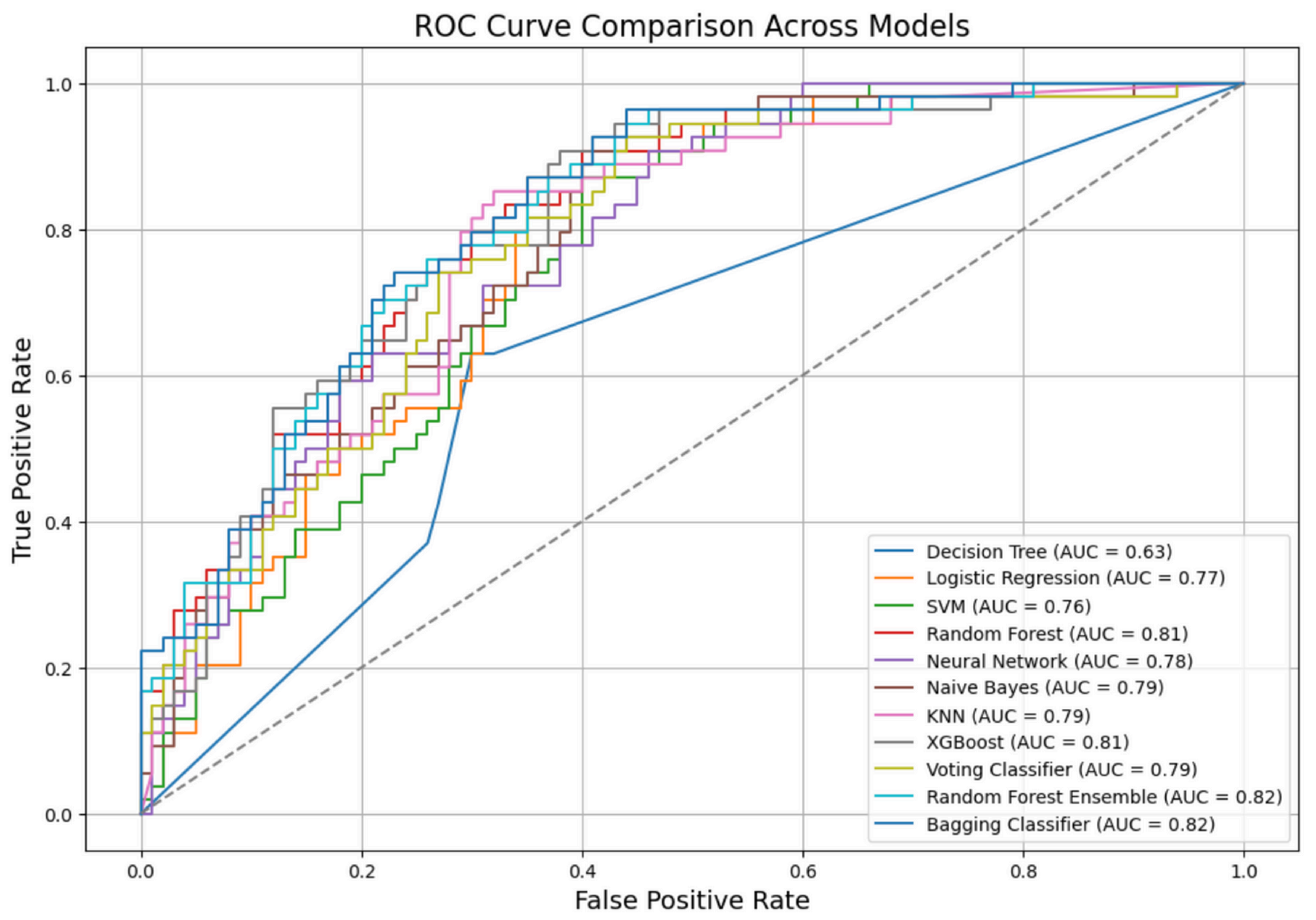


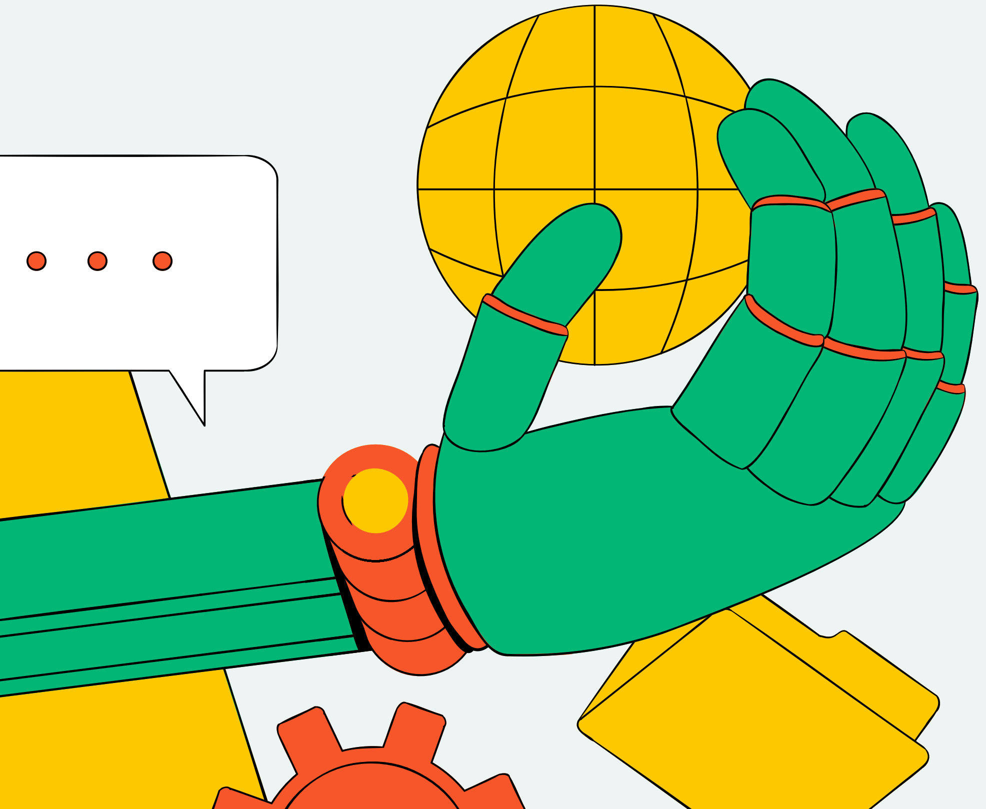


ROC CURVE

To assess the ability of each model to distinguish between diabetic and non-diabetic patients, we plotted ROC

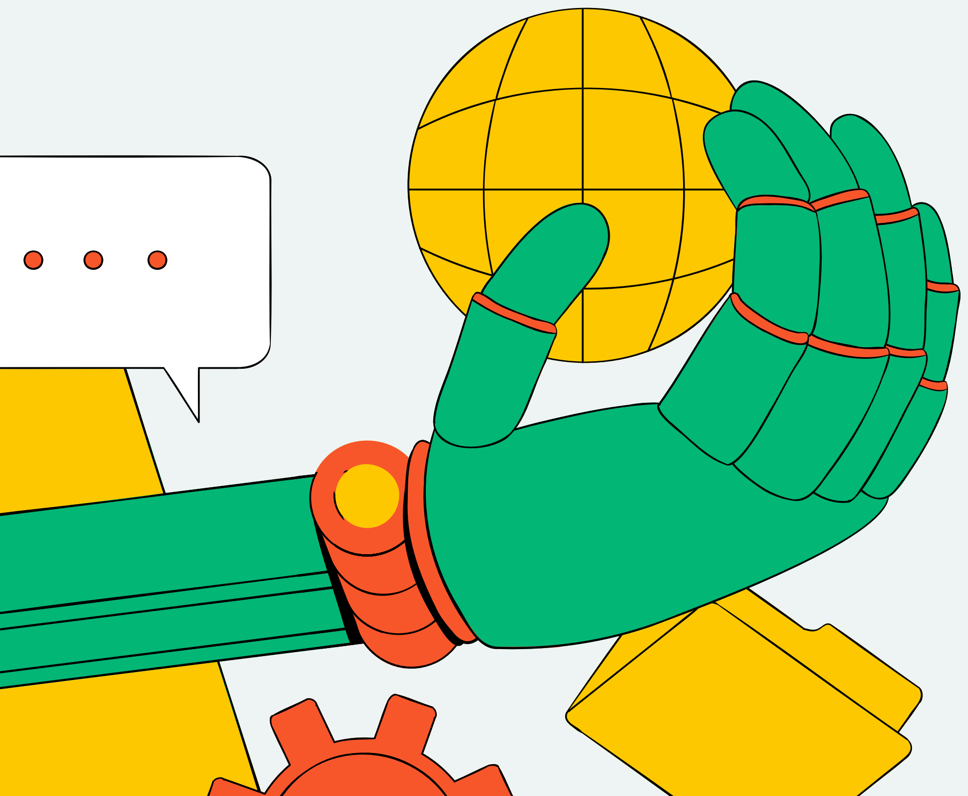
This analysis clearly confirms that e nsemble models like Bagging Classifier, \Random Forest Ensemble and XGBoost are superior not only in F1-Score but also in overall probability calibration and ROC-AUC performance.



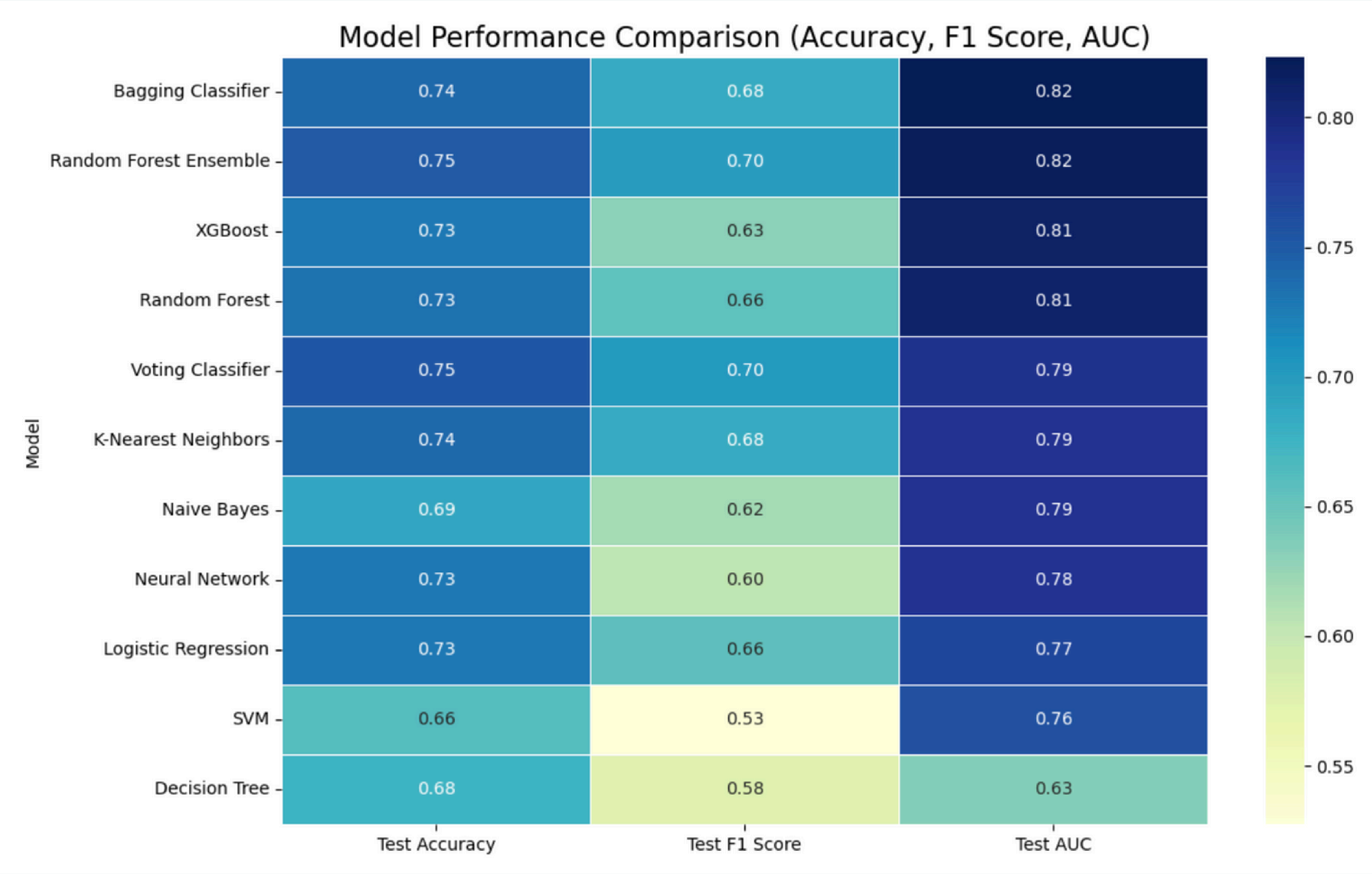


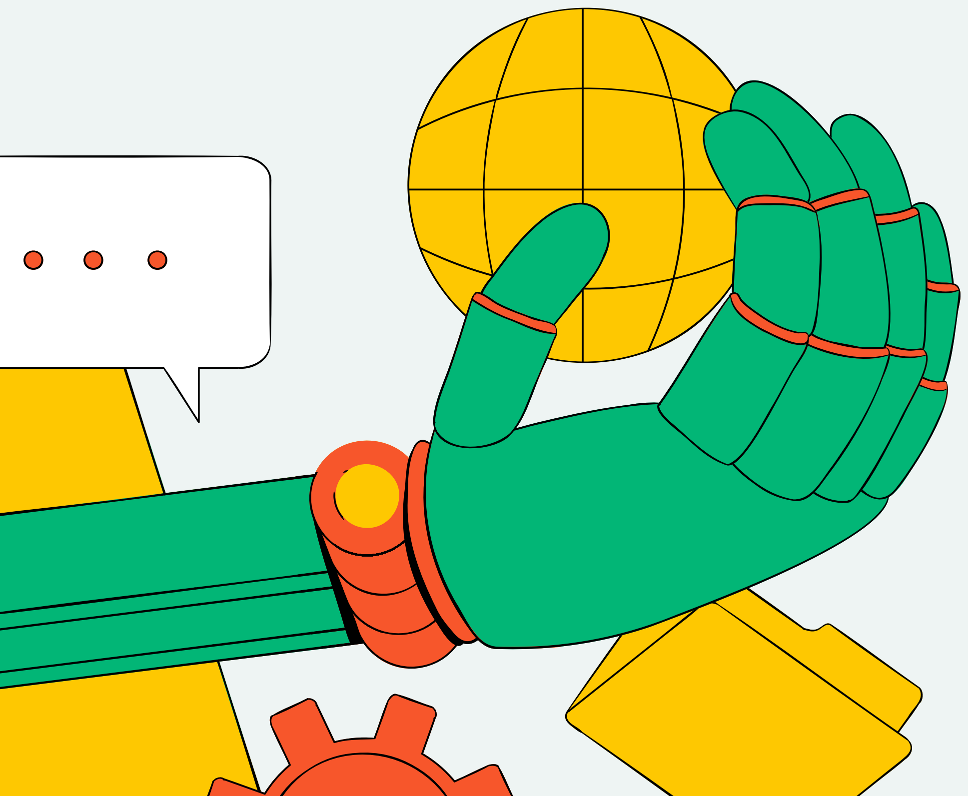
SUMMARY

	Model	Best Hyperparameters	Test Accuracy	Test F1 Score	Test AUC
0	Bagging Classifier	Random Forest(depth=8, n=150), Bagging(n_estim...	0.7403	0.6825	0.8231
1	Random Forest Ensemble	RF(depth=6, n=100) + RF(depth=8, n=150) + RF(d...	0.7500	0.6980	0.8187
2	XGBoost	n_estimators=150, learning_rate=0.1, max_depth=7	0.7273	0.6316	0.8135
3	Random Forest	n_estimators=100, max_depth=10, max_features='...	0.7338	0.6555	0.8106
4	Voting Classifier	DT (best) + RF (best) + XGB (best), voting='soft'	0.7532	0.7018	0.7880
5	K-Nearest Neighbors	n_neighbors=11, metric='euclidean', weights='d...	0.7403	0.6825	0.7871
6	Naive Bayes	Default (GaussianNB)	0.6883	0.6190	0.7870
7	Neural Network	hidden_layer_sizes=(100, 50), activation='relu...	0.7273	0.6038	0.7841
8	Logistic Regression	penalty='l2', C=0.01, solver='liblinear'	0.7273	0.6557	0.7676
9	SVM	C=100, kernel='rbf', gamma='auto'	0.6623	0.5273	0.7581
10	Decision Tree	max_depth=10, criterion='gini', min_samples_sp...	0.6753	0.5763	0.6346



SUMMARY

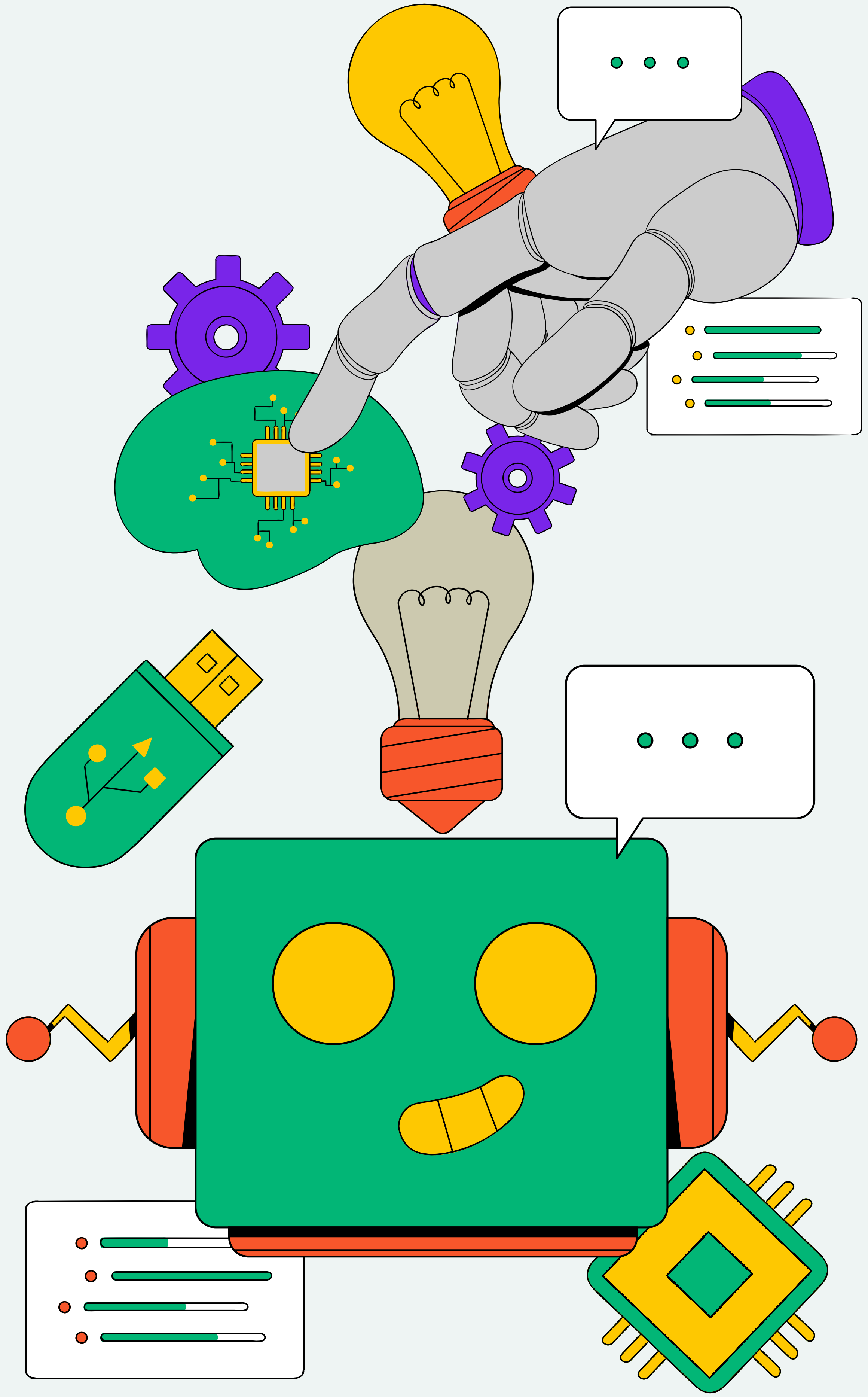

Model Performance Comparison (Accuracy, F1 Score, AUC)
Model
Bagging Classifier
0.74
0.68
0.82
Random Forest Ensemble
0.75
0.70
0.82
XGBoost
0.73
0.63
0.81
Random Forest
0.73
0.66
0.81
Voting Classifier
0.75
0.70
0.79
K-Nearest Neighbors
0.74
0.68
0.79
Naive Bayes
0.69
0.62
0.79
Neural Network
0.73
0.60
0.78
Logistic Regression
0.73
0.66
0.77
SVM
0.66
0.53
0.76
Decision Tree
0.68
0.58
0.63
Test Accuracy
Test F1 Score
Test AUC
0.80
0.75
0.70
0.65
0.60
0.55



MODEL EVALUATION

The evaluation primarily uses F1 score because the business goal requires a balance between precision & recall.

Additionally, ROC-AUC score is used to access the model’s ability to ddistinguish between classes (0 - No Diabetes, 1 - Diabetes)



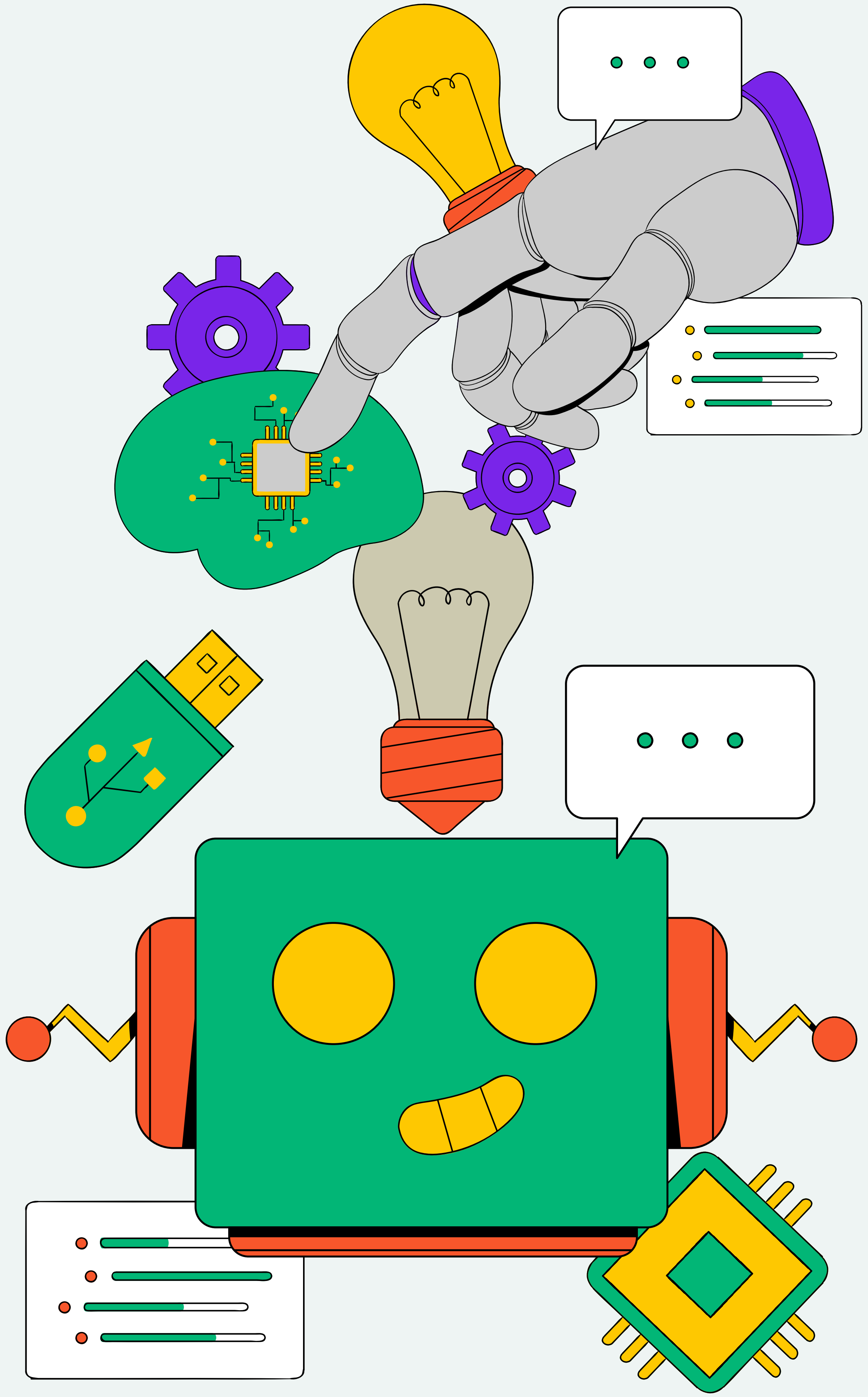
MODEL DEPLOYEMENT

•Deployment Option 1:

Deploy the Random Forest Ensemble as a batch model, processing customer/patient data every few hours or daily.

•Deployment Option 2:

Deploy it as a real-time microservice API using frameworks like FastAPI or Flask, hosted on cloud infrastructure (AWS, Azure, GCP).



THANK YOU

QUESTIONS?