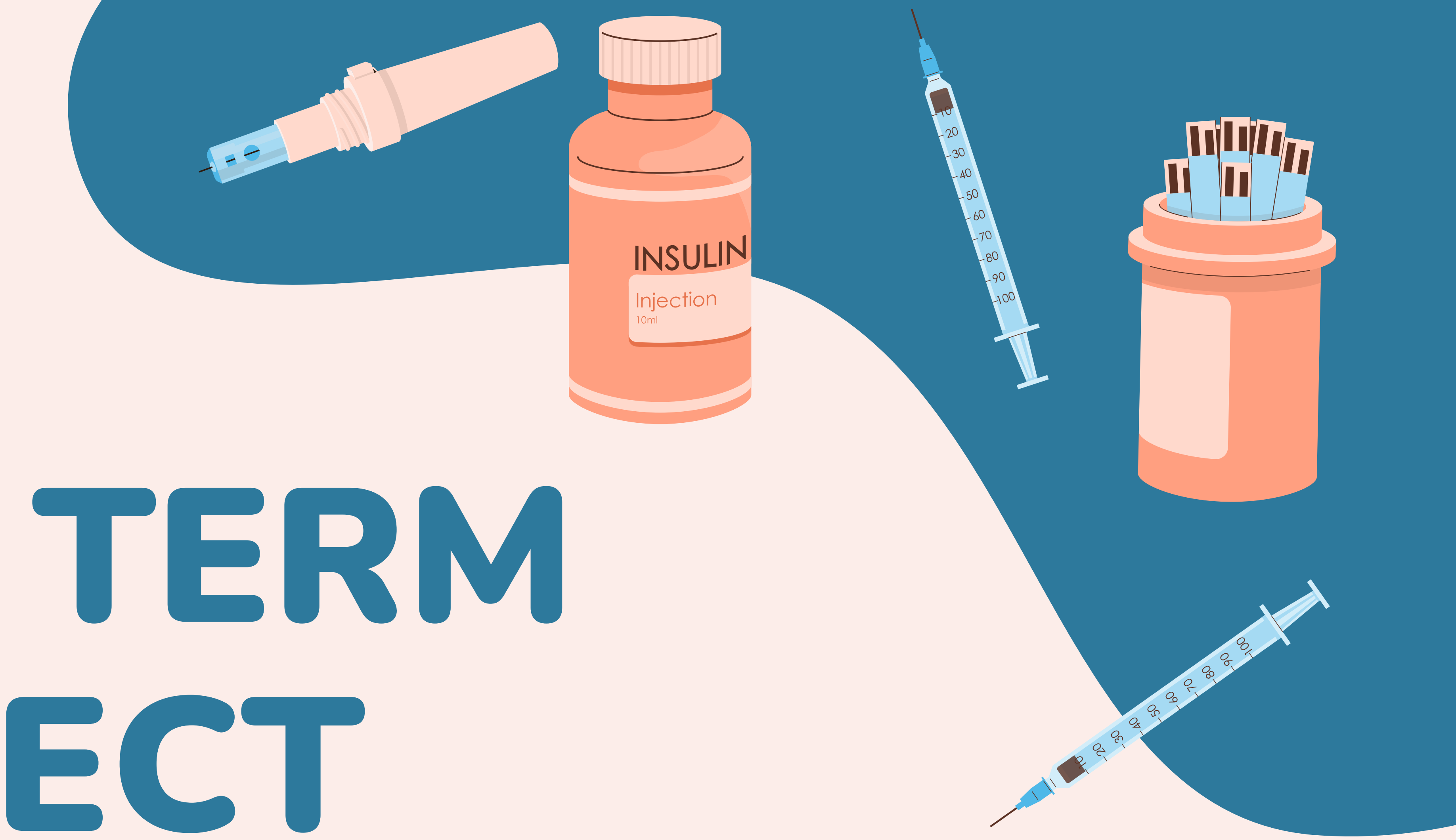
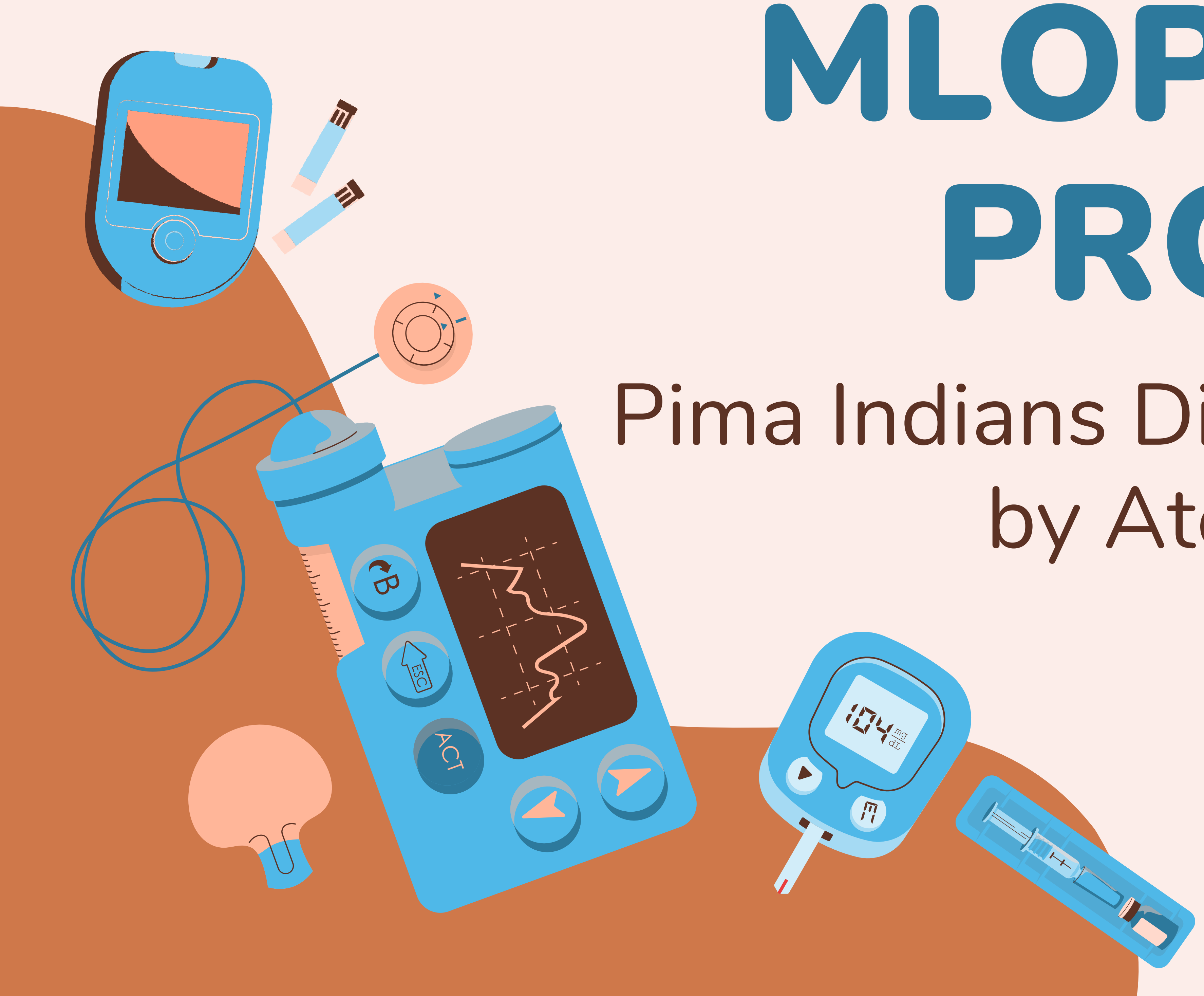


MLOPS TERM PROJECT

Pima Indians Diabetes Prediction System
by Atena Jafari Parsa



- Problem:

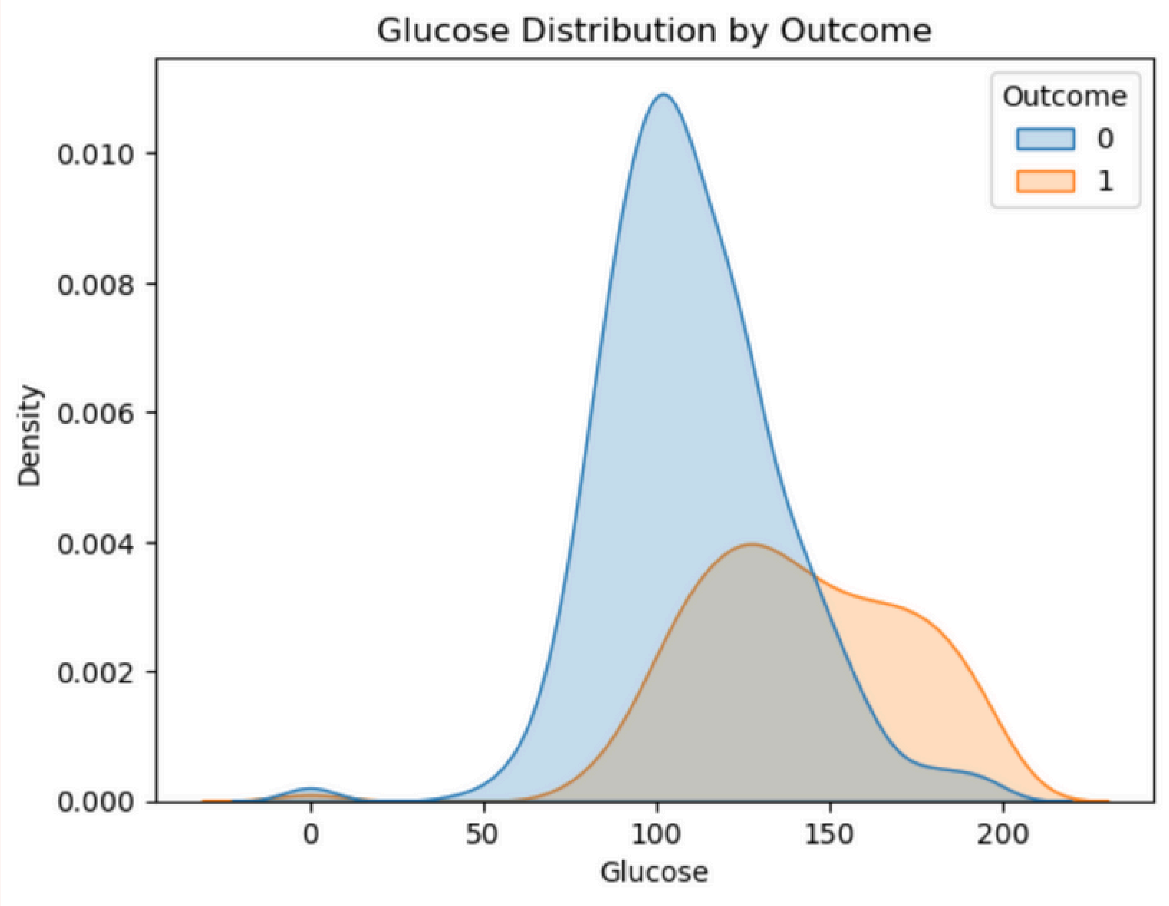
Predict whether a person has diabetes based on health indicators

- Dataset:

Pima Indians Diabetes Dataset (Kaggle)

    - 768 samples
    - 8 input features
    - Target: Outcome → 0 (No diabetes), 1 (Diabetes)

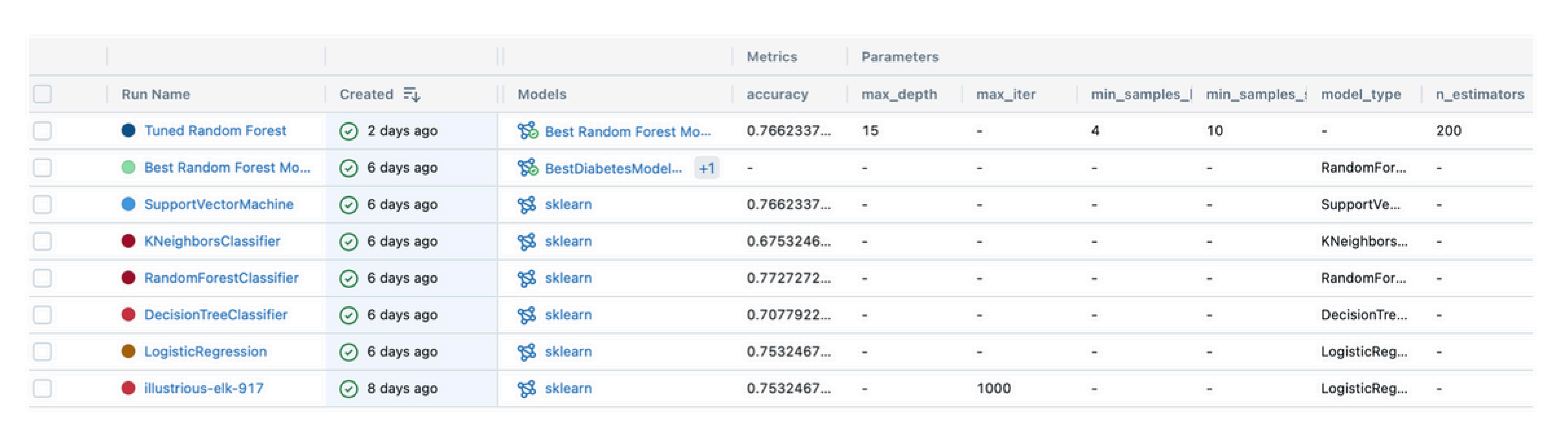
- Why this dataset?
    - Structured, clean, interpretable
    - Real-world healthcare relevance
    - Ideal for quick experimentation and lifecycle demos



Exploratory Data Analysis (EDA)

⚙️ MLOps Pipeline with MLflow

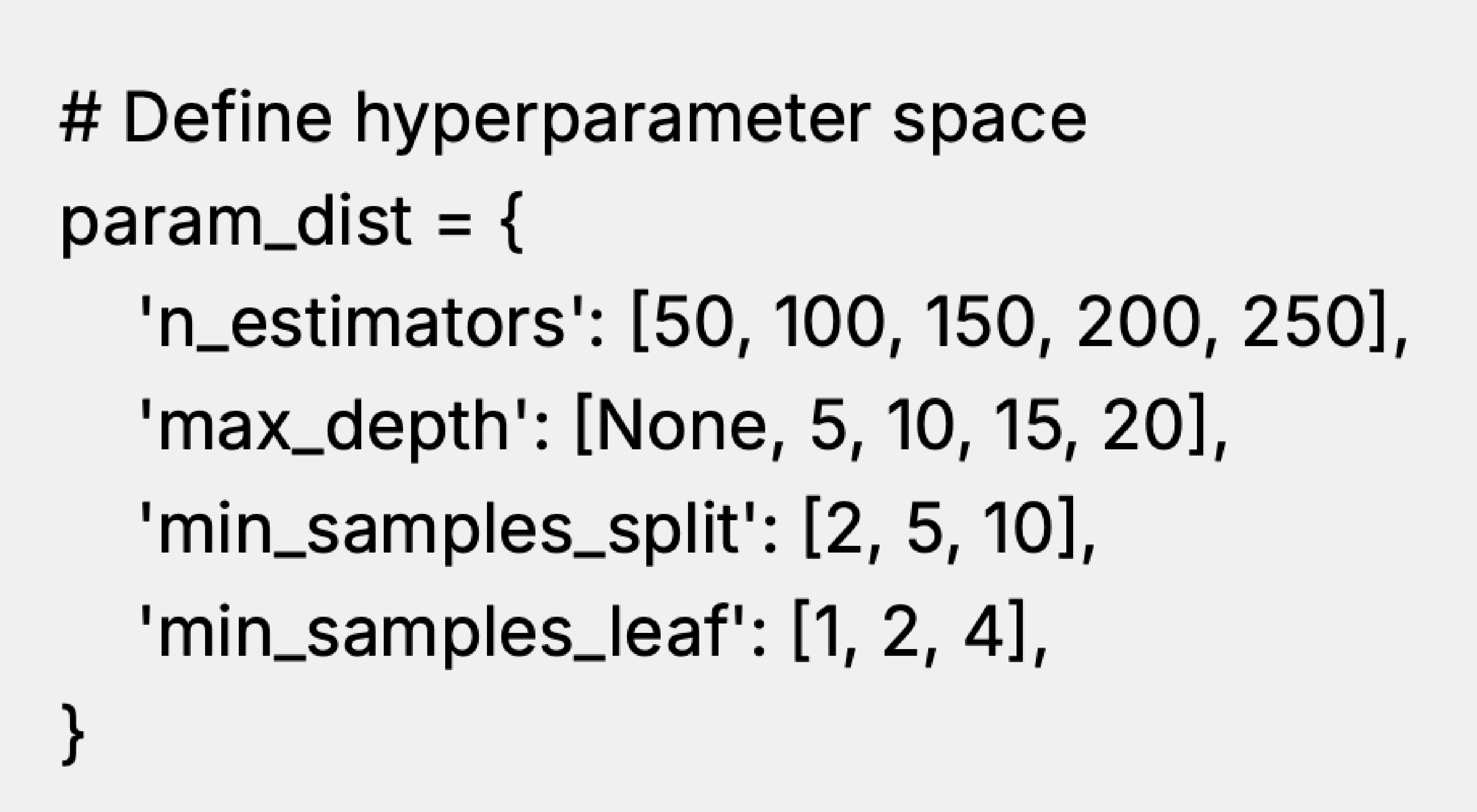
- Training Process:
  - Used train_model.py for baseline (Logistic Regression)
  - Used train_multiple_models.py to train 5 models: Random Forest, SVM, Logistic Regression, Decision Tree, KNN
- MLflow Tracking:
  - Logged parameters, metrics, and models
  - Visualized runs and comparisons in MLflow UI
- Model Registry:
  - Registered best model (Random Forest) as BestDiabetesModel
  - Promoted version 1 to Production
- Hyperparameter Tuning:
  - Tried tuning RF with tune_hyperparams.py
  - But original model had higher accuracy, so tuning result was not promoted



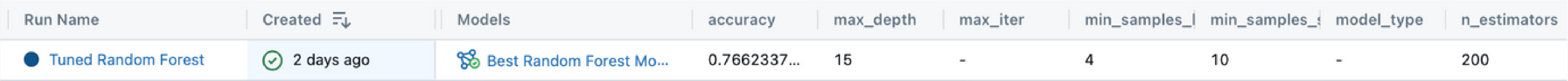
				Metrics	Parameters					
☐	Run Name	Created	Models	accuracy	max_depth	max_iter	min_samples_l	min_samples_	model_type	n_estimators
☐	● Tuned Random Forest	✓ 2 days ago	Best Random Forest Mo...	0.7662337...	15	-	4	10	-	200
☐	● Best Random Forest Mo...	✓ 6 days ago	BestDiabetesModel... +1	-	-	-	-	-	RandomFor...	-
☐	● SupportVectorMachine	✓ 6 days ago	sklearn	0.7662337...	-	-	-	-	SupportVe...	-
☐	● KNeighborsClassifier	✓ 6 days ago	sklearn	0.6753246...	-	-	-	-	KNeighbors...	-
☐	● RandomForestClassifier	✓ 6 days ago	sklearn	0.7727272...	-	-	-	-	RandomFor...	-
☐	● DecisionTreeClassifier	✓ 6 days ago	sklearn	0.7077922...	-	-	-	-	DecisionTre...	-
☐	● LogisticRegression	✓ 6 days ago	sklearn	0.7532467...	-	-	-	-	LogisticReg...	-
☐	● illustrious-elk-917	✓ 8 days ago	sklearn	0.7532467...	-	1000	-	-	LogisticReg...	-

Hyperparameters Tuning

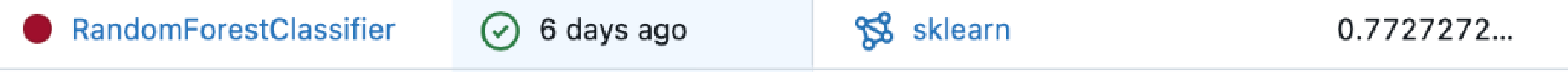
- 10 different combination were randomly tried (n_iter=10) and for each one, the training was split into 3 parts. 2 parts used for training and one for testing (**3-fold cross validation)**
- The best version with the highest accuracy (the best combination from the hyperparameters space) was picked. The hyperparameters combination in the registered tune model, gave the highest accuracy comparing to the other 9 random hyperparameter combinations.



```
# Define hyperparameter space
param_dist = {
    'n_estimators': [50, 100, 150, 200, 250],
    'max_depth': [None, 5, 10, 15, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
}
```



Run Name	Created	Models	accuracy	max_depth	max_iter	min_samples_l	min_samples_s	model_type	n_estimators
● Tuned Random Forest	✓ 2 days ago	Best Random Forest Mo...	0.7662337...	15	-	4	10	-	200



● RandomForestClassifier	✓ 6 days ago	sklearn	0.7727272...

Why Tuning made it worse?

- Scikit-learn’s default RandomForestClassifier() already uses good defaults: n_estimators=100, max_features='sqrt', etc.
- So tuning didn’t have much room to improve — and could easily **overfit.**
- Tuning adds randomness and with only 10 combinations (n_iter=10), there’s no guarantee that better settings are tested.
- Small dataset = high variance → overfitting on cross-validation folds.
- Very deep trees (max_depth=20)
- Fewer samples per split (min_samples_split=2)
- These can lead to complex trees that overfit on small data.

📈 Results & Monitoring

- **Best Model:** Random Forest Classifier

Accuracy: 77.27%

Deployment: Served via Flask API (deploy_model.py)

Accessible at localhost: 5003/predict

Promoted to Production Stage as BestDiabetesModel

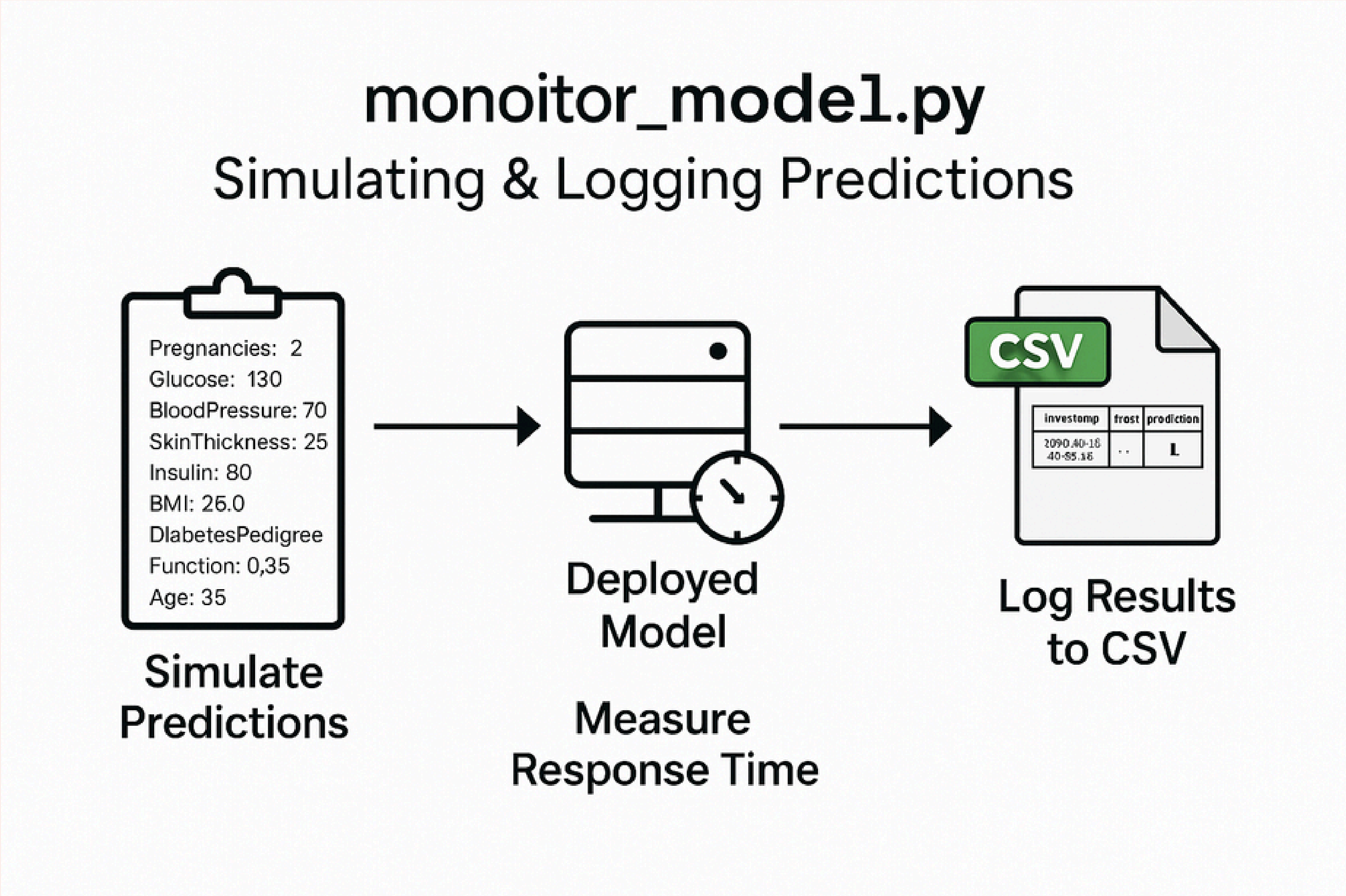
- **Monitoring:**

Simulated 5 live requests using monitor_model.py

Logged:

- Input data
- Prediction result
- Latency (response time)
- Timestamp

Saved logs in prediction_monitoring_log.csv



monoitor_model.py

Simulating & Logging Predictions

Pregnancies: 2
Glucose: 130
BloodPressure: 70
SkinThickness: 25
Insulin: 80
BMI: 26.0
DIabetesPedigree
Function: 0,35
Age: 35

My Github Repository:

https://github.com/AtenaJP22/MLOps-Term-Project

My Notion Report:

https://www.notion.so/MLOps-Term-Project-1de9e2cf8572801cbe01d3383159545f?pvs=4

Dataset Retrieved from:

https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database