

MILANO, i-DAYS 2023

dAib
*Personalized
Diabetes Prevention*

Milano, 7-8-9 Novembre 2023



Co-funded by the
European Union



Our Team

BSc in Mathematics and AI



Federico Russo



Changchen Yu



Lorenzo Tondini



Giovanni De Muri

BSc in Biomedical and Computer Engineering



Maria Elisabetta Granieri



Silvia Giada Frisullo



Nicolò Venturini

Problem

Number of people who suffer from **diabetes**



Cost of care per patient

3000 €/year

Value Proposition

"A personalized machine learning **algorithm**, based on **statistical** and **clinical data**, capable of **predicting** the **insurgence** of **type II diabetes** in healthy patients in order to prevent its development.

The algorithm is integrated in the **healthcare** management apps provided by insurance companies and diagnostic centers. "

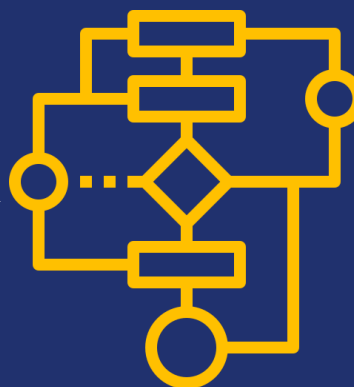
How it Works

Patient data



Input

ML algorithm



Output

Probability of diabetes



Diagnostic Exams

Anamnesis

Learns the **statistical relationship** between patient data and diabetes

Recognizes patterns on new **unseen** data

Prototype

	Diabetes_012	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	Fruits	...	AnyHealthcare	NoDocbcCost	Ger
0	0.0	1.0	1.0	1.0	40.0	1.0	0.0	0.0	0.0	0.0	...	1.0	0.0	
1	0.0	0.0	0.0	0.0	25.0	1.0	0.0	0.0	1.0	0.0	...	0.0	1.0	
2	0.0	1.0	1.0	1.0	28.0	0.0	0.0	0.0	0.0	1.0	...	1.0	1.0	
3	0.0	1.0	0.0	1.0	27.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0	
4	0.0	1.0	1.0	1.0	24.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0	
...	
253675	0.0	1.0	1.0	1.0	45.0	0.0	0.0	0.0	0.0	1.0	...	1.0	0.0	
253676	2.0	1.0	1.0	1.0	18.0	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	
253677	0.0	0.0	0.0	1.0	28.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0	
253678	0.0	1.0	0.0	1.0	23.0	0.0	0.0	0.0	0.0	1.0	...	1.0	0.0	
253679	2.0	1.0	1.0	1.0	25.0	0.0	0.0	1.0	1.0	1.0	...	1.0	0.0	

Prototype

Model

```
#model = LogisticRegression(multi_class = "multinomial", max_iter = 500)
model = KNeighborsClassifier(n_neighbors = 10)
#model = SVC()
```

```
model = xgboost.XGBClassifier(max_depth = 3, learning_rate = 0.4, n_estimators = 600, objective = "binary:logistic")
```

```
import statsmodels.api as sm
sd_model = sm.Logit(y, sm.add_constant(X)).fit(dispatch = 0)
print(sd_model.pvalues)
sd_model.summary()
```

const	0.000000e+00
HighBP	8.834547e-304
HighChol	7.121084e-213
CholCheck	7.316015e-63
BMI	0.000000e+00
Smoker	9.290749e-01
Stroke	7.496383e-05
HeartDiseaseorAttack	6.323628e-19
PhysActivity	1.197257e-01
Fruits	7.818480e-02
Veggies	8.883898e-03
HvyAlcoholConsump	2.414335e-53
AnyHealthcare	1.972641e-01
NoDocbcCost	5.774518e-01
GenHlth	0.000000e+00
MentHlth	6.884790e-04
PhysHlth	2.937376e-12
DiffWalk	8.809206e-06
Sex	2.896713e-44
Age	0.000000e+00
Education	3.159242e-04

Prototype

Data:

- BRFSS 2015 survey on 253680 US patients
- 22 parameters for every patient

83%
ROC on unseen data

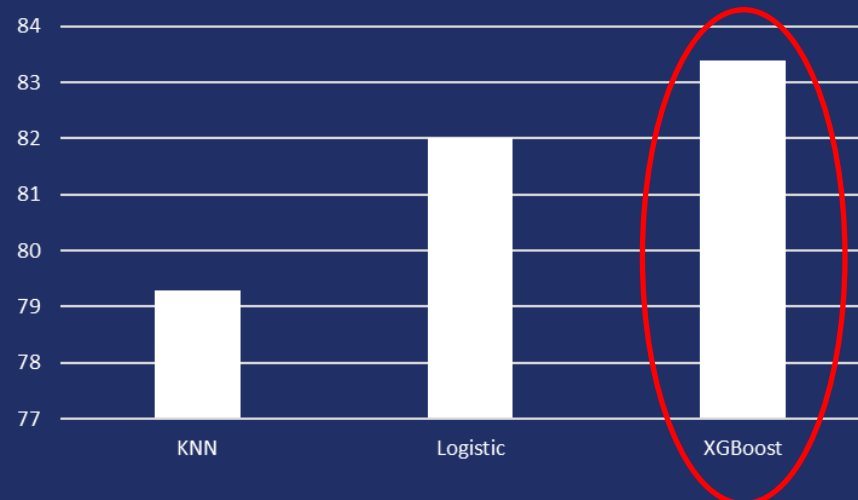
Case study (23 year old female):

high blood pressure, high cholesterol,
BMI of 25, previous heart disease, no
physical exercise



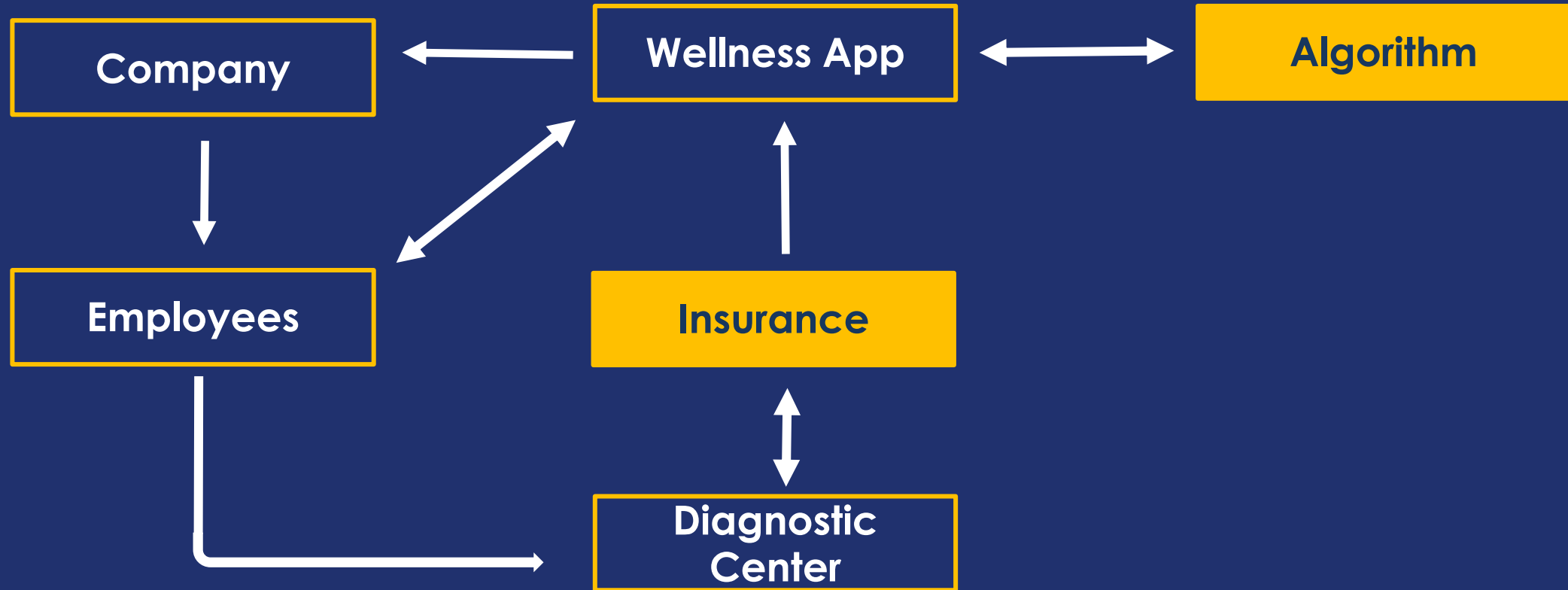
84%
probability of diabetes

Validation ROC

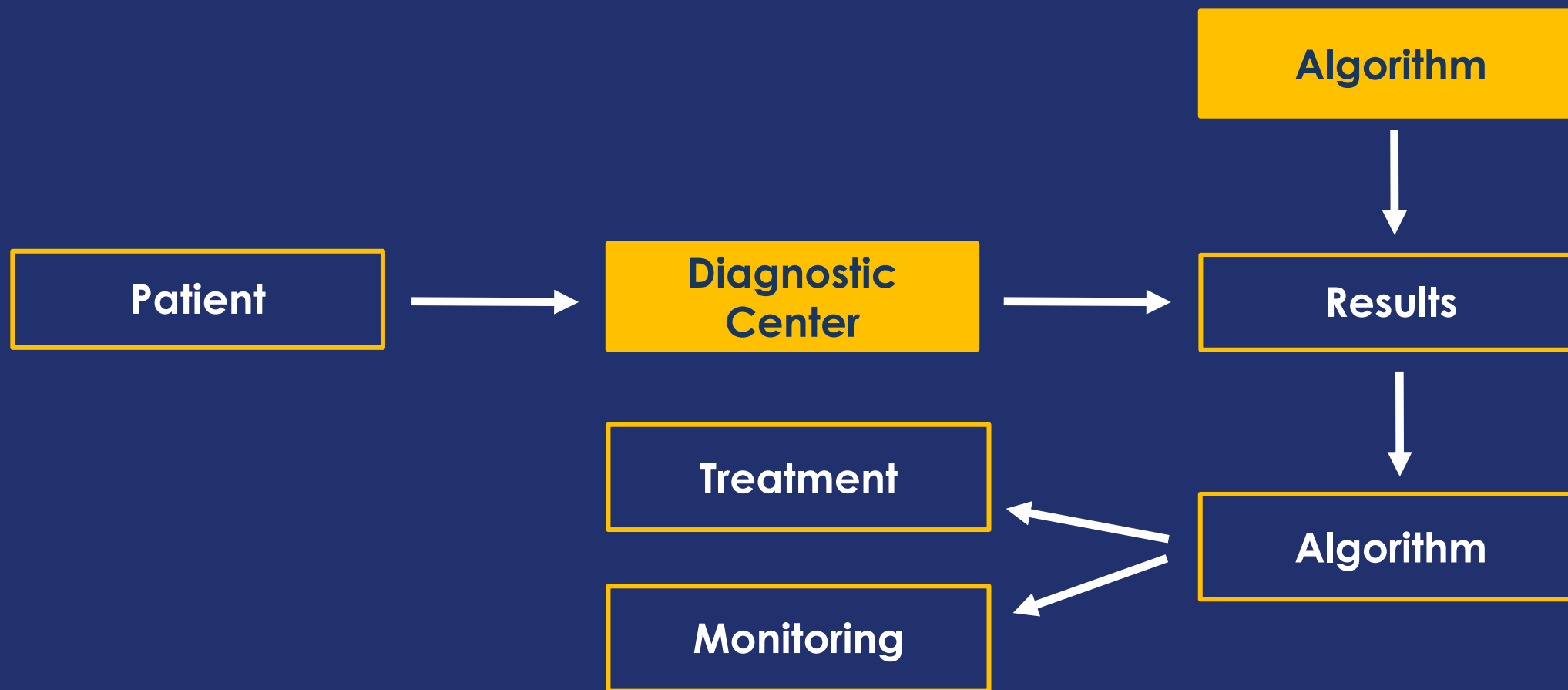


*BRFSS: Behavioral Risk Factor Surveillance

Business Model - Insurance



Business Model - Diagnostic Center



Revenue Model

- Integration costs due to the implementation of the algorithm
- Fixed **usage fee** to cover expenses
- Maintenance and upkeep costs

Insurance company

≈ 0.1 euro / commission

Diagnostic center

≈ 10% / test

Indirect Competitors

"TEMPUS



Companies that use AI for **medical imaging** and **real-time simulations**



Diagnostic centers

Roadmap

