Guggenberger, Tyler

Utilizing Digital Twins for Type II Diabetes:

Leveraging Data and AI for Predictive Healthcare

Abstract

This paper provides a framework for developing a digital twin system for the prediction of Type II Diabetes. Here, we demonstrate what is needed to construct a minimum viable product and further instruct how this framework may be expanded in future iterations to drive greater accuracy and enhanced clinical utility. The methods utilized within follow industry best practices around health information system architecture, data warehousing and model deployment. The preliminary results indicate that digital twin systems can be utilized for the prediction of Type II Diabetes and aid in clinical decision support.

Introduction

Digital Twin Technology and Healthcare

As technologies around Big Data and Artificial Intelligence (AI) have rapidly evolved, numerous breakthroughs, applications and use cases have emerged across a variety of industries, including healthcare. One such technology is the concept of a Digital Twin, which refers to a virtual representation of physical object, person, system or process that is continuously updated using real-time data for purposes of simulation, machine-learning and, ultimately, enhanced decision-making (McKinsey, 2024)(IBM, 2021).

In the healthcare industry, digital twin technology has emerged as a transformative force with the ability to help predict disease progression, identify high-risk individuals and recommend preventative care interventions. Digital twins enable healthcare providers to gather and analyze a wealth of patient data from various sources, including electronic health records, wearables, medical devices, genetic information, and social determinants of health. This holistic view provides a much richer portrait of health, which in turn allows digital twins to identify patterns, correlations, and anomalies, as well as different risk strata across patient cohorts. As Vallee (2023) notes, “Digital twins provide insights into which patients are most likely to benefit from preventative measures, early screenings, lifestyle modifications, or specific interventions. This targeted approach improves the allocation of healthcare resources, reduces costs, and enhances patient outcomes.” (p3).

With many different use cases and the continually evolving state of digital twins in healthcare, it’s important that health organizations understand how they may implement this concept into their

larger Health Information System and data environments. To demonstrate, this paper narrows the scope of utilizing digital twins to predict the likelihood of patients developing Type II diabetes, covering an end-to-end prototype of what this may look like in practice.

Type II Diabetes

Diabetes is a chronic disease that is estimated to affect over 800 million people worldwide, with Type II diabetes accounting for approximately 98% of all diabetes diagnoses (Robertson & Lipska, 2025). In the United States alone, the national cost of diabetes in 2022 was estimated to exceed $400 billion and was the 8th leading cause of death (ADA, n.d.). Diabetes complications are very serious and can lead to blindness, kidney failure, heart attacks and stroke, so early detection is critical to patient safety and improved clinical outcomes.

The World Health Organization notes that while Type II diabetes is often preventable, the onset of symptoms can be mild and may take several years to develop. As a result, the time to clinical diagnosis is often delayed until after complications have already arisen, which in turn mitigates the efficacy of lifestyle interventions alone (WHO, 2024). Thus, there is a need for earlier detection methods. We can assess the current state of digital twin technology as it relates to Type II Diabetes before transitioning into the design stage.

Current Research

There are several publications that discuss the concept of applying digital twins to Type II Diabetes, which provide very useful context and further reinforce the value proposition. For instance, Shamanna et al. (2024) analyzed one-year outcomes of a digital twin intervention for Type II diabetes and demonstrated significant improvement to glycemic control, reducing the need for anti-diabetic medication. They concluded that digital twins "could play a vital role in the future of T2D [Type II Diabetes] management by offering a comprehensive, personalized, and technologically advanced solution" (p.9). Another example can be observed in the work outlined by Zhang et al. (2024), which introduces a digital twin framework for T2D by integrating machine learning with multiomic data, knowledge graphs, and mechanistic models. This study demonstrates how digital twins can be used to virtually represent a patient's disease and facilitate real-time monitoring, analysis and simulation. In their conclusion, they note that the appropriate next steps would be "to move beyond purely statistical modeling by integrating [models] into the pipeline" (p.8).

As we can see, there is a very bullish sentiment around this technology and its ability to help predict chronic disease and progress personalized medicine. However, much of the current research is focused on outcomes and does not provide explicit instruction around how health organizations may design and deploy digital twins for themselves. To answer that question, we must turn to fundamental principles of Health Information system design.

Conceptual Design

As Levinger (2025) notes, once a business need is defined, in this case a working digital twin system, the next phase is to design a solution. The design stage is a crucial component that brings clarity to critical decisions and juncture points. It is during design that we can map out the technical architecture, outline performance requirements, and answer questions around data

inputs, hardware, software, and lay the foundation for successful project planning. For this system and its prediction of Type II Diabetes, we need to understand the architecture, data inputs, data warehousing and model deployment – we examine each below.

Architecture

One of the first decision points around architecture is the determination of utilizing a cloud-based or on-premises solution. While both options carry pros and cons, the benefits of a cloud-based environment are valuable enough in this use-case to merit selection. This is largely because cloud-based digital twins have greater implementation and operational elasticity, present a shorter time to market, possess greater interoperability, better facilitate the inclusion of new technology and offer edge computing capabilities (Wong, 2023). In the manufacturing world, where digital twins are more prevalent, many of the industry leaders rely on cloud-based digital twin systems, including Bosch, Honeywell, BMW, Rolls-Royce and Procter & Gamble (DataDynamics, n.d.). For this research, we have utilized cloud-based architecture via Databricks. This environment can host our ETL processes, store the data and execute the machine learning model.

Data Inputs: Overview

Another critical decision is around data sources and inputs. That is, what data is needed to facilitate the digital twin system and how to ingest it. The prediction of Type II Diabetes is a complex equation consisting of many variables that span multi-modal data such as clinical, genetic, environmental, social factors and more (Kaput & Dawson, 2007). As such, for a minimum viable product, this system will need to integrate clinical data from the Electronic Health Record (EHR), Internet of Things (IoT) data for wearables and biometric sensors, as well as demographic and social determinants of health (SDOH) data. Future iterations may expand the scope of the data, but baseline functionality will require the incorporation of these modalities.

Data Inputs: EHR / Clinical Data

From a clinical perspective, EHR systems have become an invaluable tool for health organizations, as they provide a centralized environment to capture information around patients, procedures, providers, labs and imaging, medications, decision support, radiology and more. The many advantages of EHRs are well-documented, but some of the key benefits include improved quality of care, better clinical outcomes, improved operational efficiencies and lower costs, as well as better clinical decision-making (CMS, 2024). As such, incorporating EHR-derived clinical data is a necessary aspect of the digital twin system.

Data Inputs: IoT / Wearables

Due to advances in technology, health related IoT data covers many disciplines and focus areas. IoT devices contain sensors that can measure and transmit data capturing temperature, blood pressure, heart rate, glucose levels, respiration and more. These sensors have been integrated into wearables like smart watches and step counters, medical equipment like continuous glucose monitors and even implantable or ingestible devices. According to one study, it is estimated that by the end of 2025, there will be approximately 75 billion connected devices worldwide (Li et al., 2024). And as Li et al. (2024) reinforce, “AI algorithms have emerged as powerful tools in

healthcare, particularly when combined with data from IoT devices. By leveraging the vast amount of data these devices generate, AI algorithms can provide real-time insights into patient health and enable personalized healthcare interventions" (p.4). This supports the hypothesis of this paper; therefore, we have included IoT data in the digital twin system.

Data Inputs: Social Determinants of Health / Environmental

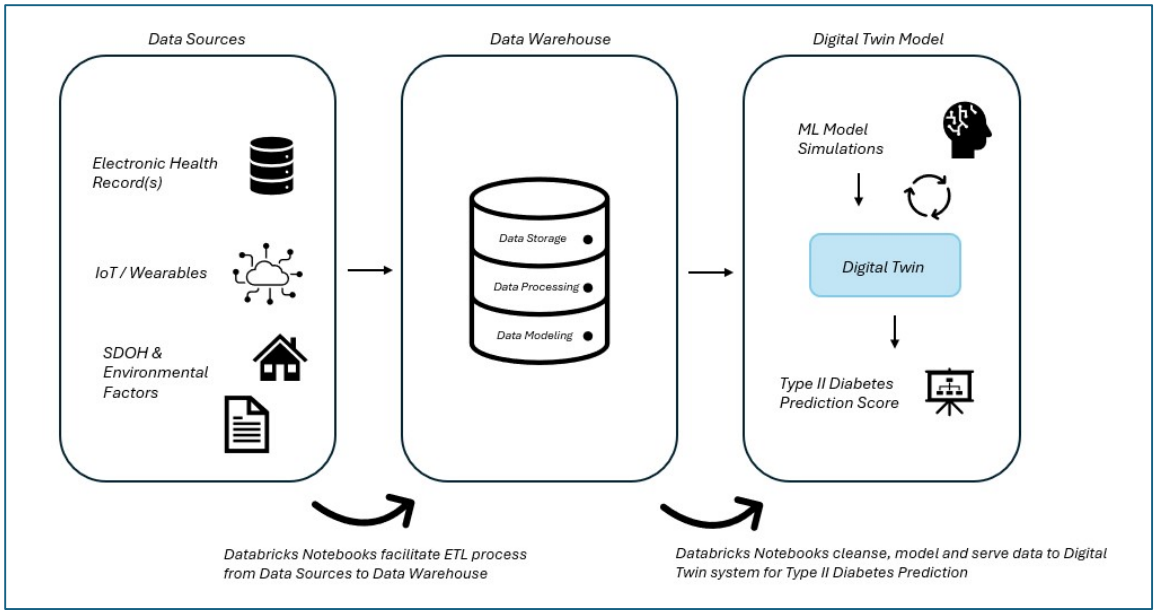
SDOH data encompasses the conditions in the environments where people are born, live, learn, work, play, worship, and age that affect a wide range of health, functioning, and quality-of-life outcomes and risks. The US Department of Health and Human Services groups SDOH factors into 5 distinct domains: Economic Stability, Education Access & Quality, Health Care Access & Quality, Neighborhood Environment, and Social & Community Context (HealthyPeople, n.d.). Together, these factors act as a major driver of health and well-being and can be used to aid in the prediction of chronic disease and progression, as well as help assuage health disparities and inequities (Murphy et al., 2024)(Seeger et al., 2022). Leveraging SDOH data in our digital twin system will help to improve accuracy and real-world applicability.

Date Warehousing & Model Deployment

Because our digital twin will utilize many different data inputs, it's critical that we have a consolidated repository available for storage, data synthesis and downstream consumption via model deployment. Here, we rely on our cloud-based Databricks environment to function as an Enterprise Data Warehouse, which carries the benefit of being able to handle unstructured, semi-structured and structured data, as well as robust checks around data quality, consistency and accuracy (AWS, n.d.). This is especially important in healthcare, where data is very often stored in a variety of formats leading to interoperability challenges. This environment is ideal for proof-of-concepts and production pipelines alike, as we have complete control over scalability, compute resources, role-based access, monitoring and data sharing (Databricks, n.d.). Building a digital twin system in a cloud-based platform is an efficient approach for health organizations looking to adopt the framework proposed herein.

Conceptual Design Figure

Synthesizing the above together, Figure 1 below depicts what this design looks like from an end-to-end perspective. This figure covers the ingestion, storage and serving of data in our cloud-based Databricks environment. Code examples for each step are covered in the 'Prototype section of this report.



Prototype & Development

Prototyping

Agile project management and software development philosophy emphasizes a focus on continuous release, feedback and iteration. As the name suggests, this methodology allows for a more dynamic and responsive approach to building products compared to the more traditional waterfall approach, where linear dependency and finalized deliverables are prioritized (Levinger, 2025). Adhering to the agile methodology, prototyping our digital twin system will allow us the opportunity to fail fast and iterate forward. Put simply, our focus in the early stages here is to generate a functional product that lays the foundation for future development.

So, using the above conceptual framework, we can now start building our prototype. Suppose we have the following variables for each data input:

Input	**Variable**
EHR / Clinical	Patient ID, Age, BMI
IoT	Patient ID, Daily Steps, Avg. Glucose, HR Variability
SDOH	Patient ID, Income Level, Smoker

Using these variables, we can generate a test dataset, construct and train a diabetes risk-score model, and test the model with a sample patient. In this prototype case study, we examine a physical twin patient that is a 55-year-old smoker with a BMI of 28. This patient averages 6500 steps a day, maintains an average glucose of 140 mg/dL, exhibits an average HR variability of 50 milliseconds, and has an annual household income of $65,000. The below code snippets demonstrate how to programmatically build this out accordingly.

```python
#Digital Twin Prototype
#Sample dataset for Physical Twin

import pandas as pd

# physical twin sample dataframe
ehr_sample = pd.DataFrame({
    'pat_id': [112233],
    'age': [55],
    'BMI': [28],
})

iot_sample = pd.DataFrame({
    'pat_id': [112233],
    'steps': [6500],
    'avg_glucose': [140],  # mg/dL
    'hr_variability': [50], #milliseconds
})

sdoh_sample = pd.DataFrame({
    'pat_id': [112233],
    'hh_income': [65000], #annual household income
    'smoker': [1],  # 1=smoker, 0=nonsmoker
})


# Merge data to unified dataset
physical_twin = pd.merge(iot_sample, sdoh_sample, on='pat_id')
physical_twin = pd.merge(physical_twin, ehr_sample, on='pat_id')
physical_twin.head()
```

	pat_id	steps	avg_glucose	hr_variability	hh_income	smoker	age	BMI
0	112233	6500	140	50	65000	1	55	28

Create physical twin sample record and generate proxy data for model training

```python
#Digital Twin Prototype
#Risk score model training

import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Generated Proxy Data for Model Training
data = pd.DataFrame({
    'steps': [6500, 4000, 10000, 3000, 8000, 4500, 12000, 2000, 7000, 5000],
    'avg_glucose': [140, 180, 120, 200, 130, 170, 110, 210, 125, 160],
    'hr_variability': [50, 40, 60, 30, 55, 45, 65, 25, 58, 43],
    'hh_income': [65000, 30000, 75000, 25000, 70000, 35000, 80000, 20000, 72000, 40000],
    'smoker': [1, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'age': [55, 60, 45, 65, 50, 58, 40, 70, 48, 63],
    'BMI': [28, 35, 24, 38, 26, 32, 23, 40, 25, 34],
    'diabetes_risk': [1, 1, 0, 1, 0, 1, 0, 1, 0, 1]  # 1=high risk, 0=low risk
})

# Label and Train Model
X = data[['steps', 'avg_glucose', 'hr_variability', 'hh_income', 'smoker', 'age', 'BMI']]
y = data['diabetes_risk']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print(f'Model Accuracy: {accuracy_score(y_test, y_pred):.2f}')
```

```
Model Accuracy: 1.00
```

Train model and score physical twin record

```python
#Digital Twin Prototype
#Risk Score for Physical Twin

physical_twin_score = physical_twin[['steps', 'avg_glucose', 'hr_variability', 'hh_income', 'smoker', 'age', 'BMI']]

probability = model.predict_proba(physical_twin_score)[0][1]  # probability of high-risk
risk_label = "High" if probability > 0.5 else "Low"

print(f"Predicted Risk Category: {risk_label}")
print(f"Predicted Risk Probability: {probability:.2%}")
```

```
Predicted Risk Category: High
Predicted Risk Probability: 86.00%
```

As we can see, this patient is categorized as 'High' risk for Type II Diabetes and carries a risk probability of 86%. Using a digital twin version of this patient, we can adjust some of the variables to see how it may alter their risk category and probability. Suppose we leap forward, and this patient is now 58 years old, has quit smoking, reduced their BMI to 26 and increased both their daily steps to 8500 and income to $75000 – with all other variables held constant. The digital twin code for this simulation is as follows:

```python
#Digital Twin Prototype
#Sample dataset for Digital Twin

import pandas as pd

# digital twin sample dataframe
ehr_sample_digital = pd.DataFrame({
    'pat_id': [112233],
    'age': [58],
    'BMI': [26],
})

iot_sample_digital = pd.DataFrame({
    'pat_id': [112233],
    'steps': [8500],
    'avg_glucose': [140],  # mg/dL
    'hr_variability': [50], #milliseconds
})

sdoh_sample_digital = pd.DataFrame({
    'pat_id': [112233],
    'hh_income': [75000], #annual household income
    'smoker': [0],  # 1=smoker, 0=nonsmoker
})


# Merge data to unified dataset
digital_twin = pd.merge(iot_sample_digital, sdoh_sample_digital, on='pat_id')
digital_twin = pd.merge(digital_twin, ehr_sample_digital, on='pat_id')
digital_twin.head()
```

	pat_id	steps	avg_glucose	hr_variability	hh_income	smoker	age	BMI
0	112233	8500	140	50	75000	0	58	26

Adjust variables to depict intervention and score digital twin

```python
#Digital Twin Prototype
#Risk Score for Digital Twin

digital_twin_score = digital_twin[['steps', 'avg_glucose', 'hr_variability', 'hh_income', 'smoker', 'age', 'BMI']]

probability = model.predict_proba(digital_twin_score)[0][1]  # probability of high-risk
risk_label = "High" if probability > 0.5 else "Low"

print(f"Predicted Risk Category: {risk_label}")
print(f"Predicted Risk Probability: {probability:.2%}")
```

```
Predicted Risk Category: Low
Predicted Risk Probability: 49.00%
```

Based on these changes, the digital twin risk category is now classified as 'Low' and the Risk Probability has been reduced to 49%. Of course, this is just one hypothetical simulation that arbitrarily adjusted the variables, so it is important to regard this as a proof-of-concept and not a production-ready deliverable. However, because this model does allow us to begin quantifying risk

levels based on a diverse set of variables, additional testing and feedback is warranted. To continue enhancing this digital twin system, we can assess the opportunities for improvement and conduct a review of this first prototype.

Prototype 1 Iteration and Feedback

As noted above in the agile discussion, our focus is to refine and iterate. So, what can we improve? Firstly, in our prototype, the model is trained using sample data for demonstration purposes only and at just 10 records, both the sample size and quality of data is a major issue. In practice, it is suboptimal to rely on proxy data like this and it would be much improved to utilize valid data covering larger volumes of records. In addition, this prototype does not contain any type of visualization or plotted output, which may strengthen the usability and utility of the information being presented. To iterate, we should consider enriching our dataset wherever possible and layering in a visualization component to aid in presentation.

Prototype 2

Addressing the visualization concern, consider the below output, which plots patient risk score against a predefined threshold – in this case 50%. The threshold can, of course, be adjusted up or down depending on clinical preference, but if we consider how this may look across the aggregate, we could quickly and easily identify which patients in a Primary Care Provider's panel or purview may warrant additional follow-up. It's feasible to conceptualize how something like this may eventually be layered into a Patient Chart or existing EMR module for quick review during clinical encounters.

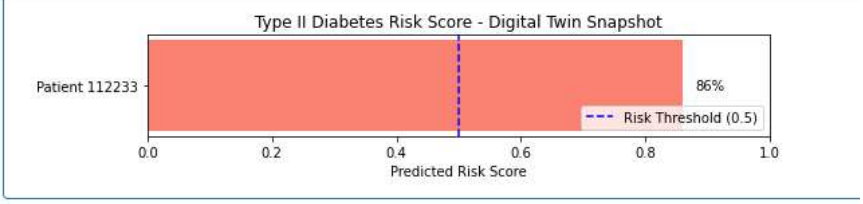
```
import matplotlib.pyplot as plt
import numpy as np

# Physical Twin Risk Score
risk_score = 0.86
threshold = 0.5

fig, ax = plt.subplots(figsize=(8, 2))
ax.barh(['Patient 112233'], [risk_score], color='salmon')
ax.axvline(threshold, color='blue', linestyle='--', label='Risk Threshold (0.5)')

ax.text(risk_score + 0.02, 0, f'{risk_score:.0%}', va='center', fontsize=10)
ax.set_xlim(0, 1)
ax.set_xlabel('Predicted Risk Score')
ax.set_title('Type II Diabetes Risk Score - Digital Twin Snapshot')
ax.legend(loc='lower right')

plt.tight_layout()
plt.show()
```



Set threshold to desired level and plot risk score for review

As for enriching our inputs, though we do not have access to live, production environment, we can still improve our model by leveraging data that has been published to the public domain. The National Institute of Diabetes and Digestive and Kidney Diseases has made available a dataset that covers 768 patients and includes several of our variables: age, glucose and BMI, as well as a Diabetes Outcome Indicator to train the model on. That said, because this data does not include Steps, HR Variability, Income, Smoking Status or a Patient Identifier, we will still need to proxy in those values. It should also be noted that this data is specifically filtered down female patients at least 21 years of age and of Pima Indian Heritage, so further iterations of the digital twin will want to expand beyond this specified cohort, but it is sufficient for the sake of progressing our digital twin model forward (UCI, 2016).

The next iteration phase loads in the raw data and manually generates proxy values for the missing variables. It then re-trains the earlier model using the larger dataset and again applies the digital twin concept against a physical twin example. Code snippets for these steps are covered below:

```
#Load in National Institute of Diabetes dataset exactly as it comes from source
file_location = "/FileStore/tables/diabetes.csv"
file_type = "csv"

infer_schema = "true"
first_row_is_header = "true"
delimiter = ","

df_pima = spark.read.format(file_type) \
  .option("inferSchema", infer_schema) \
  .option("header", first_row_is_header) \
  .option("sep", delimiter) \
  .load(file_location)

display(df_pima)
```

▸ (3) Spark Jobs

▸ df_pima: pyspark.sql.dataframe.DataFrame = [Pregnancies: integer, Glucose: integer ... 7 more fields]

Table +

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
17	0	118	84	47	230	45.8	0.551	31	1
18	7	107	74	0	0	29.6	0.254	31	1
19	1	103	30	38	83	43.3	0.183	33	0

```
#Refine down to relevant columns for prototype digital twin system
df_pima_filtered = df_pima['Glucose', 'BMI', 'Age', 'Outcome']
display(df_pima_filtered)
```

▸ (1) Spark Jobs

▸ df_pima_filtered: pyspark.sql.dataframe.DataFrame = [Glucose: integer, BMI: double ... 2 more fields]

Table +

	Glucose	BMI	Age	Outcome
1	148	33.6	50	1
2	85	26.6	31	0
3	183	23.3	32	1
4	89	28.1	21	0
5	137	43.1	33	1
6	116	25.6	30	0
7	78	31	26	1
8	115	35.3	29	0
9	197	30.5	53	1

Load in expanded dataset and filter down to relevant variables

```python
#Manually generate missing variables and rename glucose column to match earlier prototype

from pyspark.sql.functions import monotonically_increasing_id, concat, lpad, lit, rand, round, when, floor, col

df_pima_filtered = df_pima_filtered.withColumn("row_num", monotonically_increasing_id())
df_pima_filtered = df_pima_filtered.withColumn("pat_id", concat(lit("P"), lpad(df_pima_filtered["row_num"].cast("string"), 5, "0")))
df_pima_filtered = df_pima_filtered.withColumn("steps", floor(rand() * 8000 + 4000))
df_pima_filtered = df_pima_filtered.withColumn("hr_variability", round(rand() * 80 + 20, 0))
df_pima_filtered = df_pima_filtered.withColumn("hh_income", floor(rand() * 100000 + 20000))
df_pima_filtered = df_pima_filtered.withColumnRenamed("Glucose", "avg_glucose")

df_pima_filtered = df_pima_filtered.drop("row_num")
display(df_pima_filtered)
```

▸ (1) Spark Jobs

▸ df_pima_filtered: pyspark.sql.dataframe.DataFrame = [avg_glucose: integer, BMI: double ... 7 more fields]

Table +

	avg_glucose	BMI	Age	Outcome	pat_id	steps	hr_variability	hh_income	smoker
1	148	33.6	50	1	P00000	4828	91	57308	0
2	85	26.6	31	0	P00001	10217	64	36192	1
3	183	23.3	32	1	P00002	9276	52	103390	0
4	89	28.1	21	0	P00003	7763	81	62319	0

```python
df_final = df_pima_filtered.toPandas()
```

▾ (1) Spark Jobs
▾ Job 38 View (Stages: 1/1)
Stage 39 1/1 succeeded View

Augment expanded dataset with generated values for missing variables and re-train model

▶ ✓ 03:41 PM (<1s) 12

```python
#train model against expanded dataset

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report

X = df_final.drop(columns=['Outcome', 'pat_id'])
y = df_final['Outcome']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# print summary of model results
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.76      0.82      0.79        99
           1       0.62      0.55      0.58        55

    accuracy                           0.72       154
   macro avg       0.69      0.68      0.69       154
weighted avg       0.71      0.72      0.72       154
```

Apply intervention and score physical twin against digital twin

```python
# select physical twin from dataset
physical_twin = X_test.iloc[0].copy()

# leverage model for physical twin
physical_prob = model.predict_proba([physical_twin])[0][1]

# apply Digital Twin by increasing daily steps and lowering average glucose
digital_twin = physical_twin.copy()
digital_twin['steps'] += 2000
digital_twin['avg_glucose'] -= 15

# leverage model for digital twin
digital_prob = model.predict_proba([digital_twin])[0][1]

# print comparison
print("=== Digital Twin Simulation ===")
print(f"Physical Twin Risk: {physical_prob:.2f}")
print(f"Digital Twin Risk (improved lifestyle): {digital_prob:.2f}")
```

```
=== Digital Twin Simulation ===
Physical Twin Risk: 0.60
Digital Twin Risk (improved lifestyle): 0.31
```

Prototype 2 Results & Discussion

In reviewing the results of the second prototype, consider the classification report that was generated for the latest model results. As indicated by the support value and test_size, this model was trained against 20%, or 154, of the 768 records. The other 80% were scored using the model. Under this sample size, the model yielded an accuracy of 72%, which demonstrates reasonable predictive power but leaves room for improvement (Google, n.d.).

For each of the other metrics, we have one value pertaining to the No Diabetes Outcome class and another for the Has Diabetes Outcome class. In this summary, the Precision represents all the True Positives / (True Positives + False Positives), so among the Has Diabetes class, this model was correct 62% of the time it scored a patient as having diabetes. Conversely, among the No Diabetes class, the model was better performing at 76% accuracy when scoring patients non-diabetic. Recall, on the other hand, is capturing all the True Positives / (True Positives + False Negatives). Of the No Diabetes class, this model did correctly identify 82% of all non-diabetic patients, but only 55% of the actual diabetic patients. And finally, the F1-score is a measure of (2 x Precision x Recall) / (Precision + Recall), which represents a harmonic mean of precision and recall meant to serve as a single metric covering both false positives and false negatives and a balance between the trade-off associated with precision and recall (Acharya, 2024). The No Diabetes class shows strong overall performance with its F1-score, while the Has Diabetes class lacks utility as it misses almost half of all true positives.

Our second prototype also demonstrated the impact of lifestyle intervention on diabetes risk score via the digital twin model. We increased the daily steps of the physical twin by 2000 and lowered glucose by 15 mg/dL, which corresponded to an almost 30% reduction in diabetes risk. Given our classification report, we cannot confidently draw a meaningful conclusion from this intervention, however, as a framework for continued research and development, this digital twin proves again to be both insightful and useful.

Prototype 2 Iteration & Feedback

Overall, the second prototype demonstrates considerable progress over the first iteration, but there are still improvements that warrant follow-up. The primary issue remains sample size and dataset quality. While we were able to locate a larger and valid dataset with variables around age, BMI and glucose, that dataset was restricted to a specific and pre-filtered population subset. We also still had to proxy in the additional variables around SDOH and IOT. As locating publicly available datasets was a notable challenge, future iterations may consider purchasing datasets to enrich the model and continue progressing forward. Or, for health organizations, leveraging an existing internal EDW would be a fantastic way to accelerate this forward. In either case, continuous feedback, iteration and development are the recommended routes to continue this research, as they embody the philosophy of agile development and allow for fast but effective progress.

Future iterations may also consider conducting a more robust variable selection process and explore additional predictive modeling methods. As the purpose of this research was to provide a framework around building a digital twin system, there was less emphasis placed on those aspects of the project, however, in practice, it would be recommended to formally review the specific inputs and calculation methods to optimize results and yield maximum utility.

Conclusion

The research provided here serves as a schematic blueprint to design and develop a working digital twin system, which is something healthcare organizations may look to adopt in the interest of driving improved clinical outcomes and patient safety. Digital twins have a very promising future in healthcare and medicine, as they sit on the juncture point between big data and artificial intelligence. Demonstrated here was a digital twin system designed to predict Type II diabetes and quantify the impact of various interventions on risk, but the real-world applications are vast and expansive. As Katsoulakis et al. (2024) note, the ever-evolving interactions between the physical entity and the digital twin could traverse from microscopic to macroscopic scale and last from birth to death (conclusion1). This emphasizes the power and utility that digital twins may have on personalized and precision medicine, especially as technology and research continues to evolve. It is an exciting time for health informatics and health data professionals, and digital twin technology is just one example that captures the power of these disciplines.

References

McKinsey & Company. (2024, August 26). *What is digital-twin technology?* https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-digital-twin-technology

IBM. (2021, August 05). *What is a digital twin?* https://www.ibm.com/think/topics/what-is-a-digital-twin

Vallée A. (2023). *Digital twin for healthcare systems*. Frontiers in digital health, 5, 1253050. https://doi.org/10.3389/fdgth.2023.1253050

Robertson, P., Lipska, K. (2025). *Type 2 Diabetes Mellitus: Prevalence and Risk Factors*. https://www.uptodate.com/contents/type-2-diabetes-mellitus-prevalence-and-risk-factors/print

American Diabetes Association. (n.d.). *Examine the Facts*. https://diabetes.org/about-diabetes/statistics

World Health Organization. (2024, November 14). *Diabetes*. https://www.who.int/news-room/fact-sheets/detail/diabetes

Shamanna, P., Erukulapati, R.S., Shukla, A. et al. *One-year outcomes of a digital twin intervention for type 2 diabetes: a retrospective real-world study*. Sci Rep 14, 25478 (2024). https://doi.org/10.1038/s41598-024-76584-7

Zhang, Y., Qin, G., Aguilar, B., Rappaport, N., Yurkovich, J. T., Pflieger, L., Huang, S., Hood, L., & Shmulevich, I. (2024). A framework towards digital twins for type 2 diabetes. Frontiers in digital health, 6, 1336050. https://doi.org/10.3389/fdgth.2024.1336050

Wong, YK. (2023, December 12). *Digital Twins Built on a Cloud Platform Bring Much-Needed Benefits for Businesses*. ABI Research. https://www.abiresearch.com/blog/cloud-based-digital-twin-benefits

DataDynamics. (n.d.). *Amplifying Manufacturing Excellence: Unveiling five advantages of cloud-based digital twin technology*. https://www.datadynamicsinc.com/quick-bytes-amplifying-manufacturing-excellence-unveiling-five-advantages-of-cloud-based-digital-twin-technology/

Kaput, J., & Dawson, K. (2007). *Complexity of type 2 diabetes mellitus data sets emerging from nutrigenomic research: a case for dimensionality reduction?*. Mutation research, 622(1-2), 19–32. https://doi.org/10.1016/j.mrfmmm.2007.02.033

Centers for Medicare & Medicaid Services. (2024). *Electronic Health Records*. https://www.cms.gov/priorities/key-initiatives/e-health/records

Li, C., Wang, J., Wang, S., Zhang, Y. (2024). A review of IoT applications in healthcare. Neurocomputing, Volume 565. https://doi.org/10.1016/j.neucom.2023.127017.

Healthy People 2030, U.S. Department of Health and Human Services, Office of Disease Prevention and Health Promotion. https://health.gov/healthypeople/objectives-and-data/social-determinants-health

Murphy, B., Nam, Y., McClelland, R., Acquah, I., Cainzos-Achirica, M., Nasir, K., Post, W., Aldrich, M., DeFilippis, A. (2024). *Addition of Social Determinants of Health to Coronary Heart Disease Risk Prediction: The Multi-Ethnic Study of Atherosclerosis*. Journal of the American Heart Association, vol. 13, no. 14. https://www.ahajournals.org/doi/abs/10.1161/JAHA.123.033651

Segar MW, Hall JL, Jhund PS. (2022). *Machine Learning–Based Models Incorporating Social Determinants of Health vs Traditional Models for Predicting In-Hospital Mortality in Patients With Heart Failure*. *JAMA Cardiol.* 7(8):844–854. https://jamanetwork.com/journals/jamacardiology/fullarticle/2793728

Amazon Web Services. (n.d.). *What is a Data Warehouse?* https://aws.amazon.com/what-is/data-warehouse/

Databricks (n.d.). *The Best Data Warehouse is a Lakehouse*. https://www.databricks.com/product/databricks-sql

UCI Machine Learning. *Pima Indians Diabetes Database*. (2016). https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database?select=diabetes.csv

Google. (n.d.). *Classification: accuracy, recall, precision and related metrics*. https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall

Acharya, N. (2024). *Understanding Precision, Recall, F1-Score, and Support in Machine Learning Evaluation*. https://medium.com/@nirajan.acharya777/understanding-precision-recall-f1-score-and-support-in-machine-learning-evaluation-7ec935e8512e

Katsoulakis, E., Wang, Q., Wu, H. (2024).*Digital twins for health: a scoping review*. *npj Digit. Med.* **7**, 77. https://doi.org/10.1038/s41746-024-01073-0