International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Predicting Projectile Motion Using Artificial Intelligence & Machine Learning - With App and Visualization

Name - Supradip Das Dalal

B.Sc. Physics, Suri Vidyasagar College, currently pursuing Email: nupurdasdalal.suri@gmail.com

Abstract

This research explores the use of machine learning to predict the behavior of projectile motion. By generating a dataset using classical physics equations and training models such as linear regression and decision trees, we aim to estimate key parameters like range and the time of flight. The study demonstrates how artificial intelligence can be applied in solving traditional physics problems with and experimental purposes.

1. Introduction

- What is projectile motion?

Projectile motion is the curved path that an object follows when it is thrown or launched into the air and is only affected by gravity (and sometimes air resistance). It moves in two directions at the same time — horizontally and vertically.

For example, when you throw a ball, it travels forward (horizontal motion) and goes up and comes down (vertical motion). These two motions happen together, creating a curved path called a parabola.

The motion can be predicted using physics equations based on: *the initial speed of the object, the angle at which it is thrown,*

and the force of gravity pulling it down.

- Why is Projectile Motion an Interesting Subject Today?

Projectile motion might seem like an old-school topic from basic physics, but it's still very important and relevant today. That's because it helps us understand and predict how objects move in real life — from sports like cricket or football, to missiles, rockets, and even video games and animation physics.

What makes it more interesting today is how we can now use modern tools like Artificial Intelligence and Machine Learning to analyze and predict projectile motion much faster and more accurately. This opens up exciting opportunities to apply classical physics in new ways using technology, making it both educational & useful in real-world awesome applications.

- Why combine Artificial intelligence/ Machine Learning with physics?

Physics helps us understand the laws of nature, but solving physics problems can sometimes be slow, complex, or impossible to do by hand — especially when there's a lot of data or many variables. That's where the Artificial Intelligence and / or machine learning come in.

Machine learning can learn patterns from data and make fast predictions, even for problems that are hard to solve with equations alone. By combining physics with ML, we can:

Simulate real-world scenarios faster Analyze large or noisy datasets
Predict outcomes without solving complex formulas every time Discover new insights that may not be obvious from theory alone

- **What is SDD's i.e., your motivation?**

As a Physics student deeply interested in both science
and technology, I have always been curious about how traditional physics problems can be solved
using modern tools like artificial intelligence.
This project is my first step into the world of research, where I wanted to explore how machine learning can help predict something
as classical and well-known as projectile motion.
My goal is to understand how physics and AI can work together — AND to develop skills that could help me
in future academic research and innovation.

2. Literature Review

- **What have others done in this area before?**

Predicting Physical Systems with ML:
In recent years, researchers and students
have used machine learning to simulate and predict physical systems like projectile motion, pendulum swings,
or planetary orbits. For example, some online tutorials and college projects use linear regression to predict the range
or height of a projectile based on initial speed and angle.

YouTube/Online Learning Projects:
Several YouTube channels and blogs (like The AI Guy,
Nicholas Renotte, and physics-related ML videos) demonstrate
how simple machine learning models can learn basic physics patterns.
These experiments inspired many students, including
me, to apply ML in solving physics problems like projectile motion.

- Any papers or YouTube videos where AI was used in physics?

Some YouTubers like - The A.I. Guy, Nicholas Renotte etc

- How does your approach differ or add to that?

While many students and researchers have explored projectile motion with machine learning, most of them focus only on code or tutorials
without treating it as a full scientific
study. My work is different because:

It's a structured research paper written From a physics student's
perspective — not just a coding exercise.

I generate the dataset myself using classical equations — not borrowed from online sources.

I compare theoretical results with ML predictions, which helps connect
physics concepts with AI in a deeper way.

This project is meant to bridge traditional physics learning with modern AI tools.

3. Dataset Creation

- How many data points did I, i.e., Supradip use?

Velocity (m/s)	Angle (°)	Range (m)	Max Height (m)	Time of Flight (s)
20	30	35.3	5.1	2.04
25	45	63.7	7.96	3.60
30	60	79.5	11.48	5.30
35	50	98.1	12.37	4.46
40	40	105.2	13.06	5.23
15	25	19.8	2.85	1.28
28	35	61.4	7.82	3.26
32	55	84.2	13.95	5.31
45	42	149.8	17.23	6.13
18	20	20.4	2.24	1.25
38	48	131.7	17.08	5.71
22	38	38.2	5.96	2.75
50	30	110.7	15.94	5.10
26	52	58.9	10.24	4.12
42	37	131.2	16.02	5.14
33	65	70.1	13.89	6.08
10	30	8.8	1.3	1.02
5	45	2.6	0.32	0.72
20	60	30.1	8.7	3.06
25	30	55.1	5.3	2.04

✅ So the number of data points used = 20
Each row (i.e., a unique combination of velocity and angle with calculated outputs) counts as 1 data point.

*A total of 20 data points were used to train and validate the machine learning model.
Each data point includes initial velocity, angle of projection,
and the corresponding calculated range, maximum height, and time of flight..!!

- What inputs did it use?

Input Parameters : - Object Type like - a Ball or any other, etc.

Initial Velocity (m/s) Launch Angle (degrees)

Angle Reference Axis Examples -
X-axis (Horizontal) Or

Y-axis (Vertical)

Gravity Value (m/s²) Examples -
9.8 m/s² (Standard) Or
10 m/s² (Approximation)

- How did it calculate the outputs (range, time, etc.)?

The outputs such as Range (R),
Maximum Height (H), and Time of Flight (T)
for each projectile motion case were calculated using standard physics formulas. These formulas were applied to each

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

set of velocity and angle values in the dataset.

The formulas that were used here are given below - Range (R):

$R = [u^2 \times \sin(2\theta)] / g$ Maximum Height (H):

$H = [u^2 \times \sin^2(\theta)] / 2g$ Time of Flight (T):

$T = [2u \times \sin(\theta)] / g$ Half Time:

$T(Half) = [u \times \sin(\theta)] / g$
- Mention that data was generated using classical equations in Pyth

$Y = x \tan\theta - [(g \times x^2) / (2 \times u^2 \times \cos^2\theta)]$ Velocity Components (with x-axis):

$u(x) = u \cos\theta$ $u(y) = u \sin\theta$
Where: u = initial velocity, θ = angle, g = acceleration due to gravity

*These formulas were either implemented manually or calculated through the machine learning-powered AI tool that was created as part of this research. The model takes the inputs and applies these formulas internally to generate accurate outputs
for different projectile conditions.

- Data Generation Method :

The dataset for this research was generated using classical equations of projectile motion,
applied manually and through logical
implementation inside a custom-built AI-powered web tool.
The core physics formulas
for Range, Height, and Time of Flight were coded as part of the website's backend logic using the platform lovable.dev.

Although Python was not used directly in this
version, in that website lovable.dev there was some usage exists where this uses python script or python code there, the Calculations replicate what a Python
script would produce using the same
mathematical logic. Future improvements may include a deep Python-based model for deeper
machine learning training but in this project, I am trying to discuss About how can ARTIFICIAL INTELLIGENCE + Machine Learning Technology predict or in simpler
Words CALCULATE the PROJECTILE
MOTION (Which can be helpful for Many real life and also in the calculation processes for making it easier).

The AI-powered tool that I created was shared with 10 physics students. 8 out of 10 found it helpful for learning projectile motion in a visual, interactive way and they found its prediction, Results are also matching with the Original there.

I shall give the link within the few next Points for checking or visiting. For visiting my Own AI TOOL
THAT I HAVE
CREATED
FOR THIS RESEARCH
AND MANY

OTHER ASPECTS, I would
request to be Humble please.

4. Machine Learning Models

- Machine Learning Usage (Planned Integration):

In this project, the output values are currently generated using classical equations of projectile motion programmed into the application. However, the structure of the app is designed in a
way that allows future integration of
machine learning models such as Linear Regression or Decision Trees. These models will be trained using the same dataset to predict projectile parameters like
range, maximum height, and time of flight, enabling intelligent behavior and adaptation in the application.

- Model Comparison (Future Scope):

As the current implementation uses classical physics equations for generating output, no machine learning model has been trained or compared yet. However, in the next phase
of the project, models like Linear Regression, Decision Trees, and others will be tested. Their performance will be compared based
on accuracy, error margin, and generalization to unseen input data.

This comparison will help in determining the most suitable ML algorithm for real-time prediction of projectile motion.

- What library/tools did Supradip Das Dalal use (Anyone)?

Tool Used:

The application was developed and deployed
using Lovable.dev, a AI based, comparatively low-code platform that allows users to build functional AI-powered web apps.

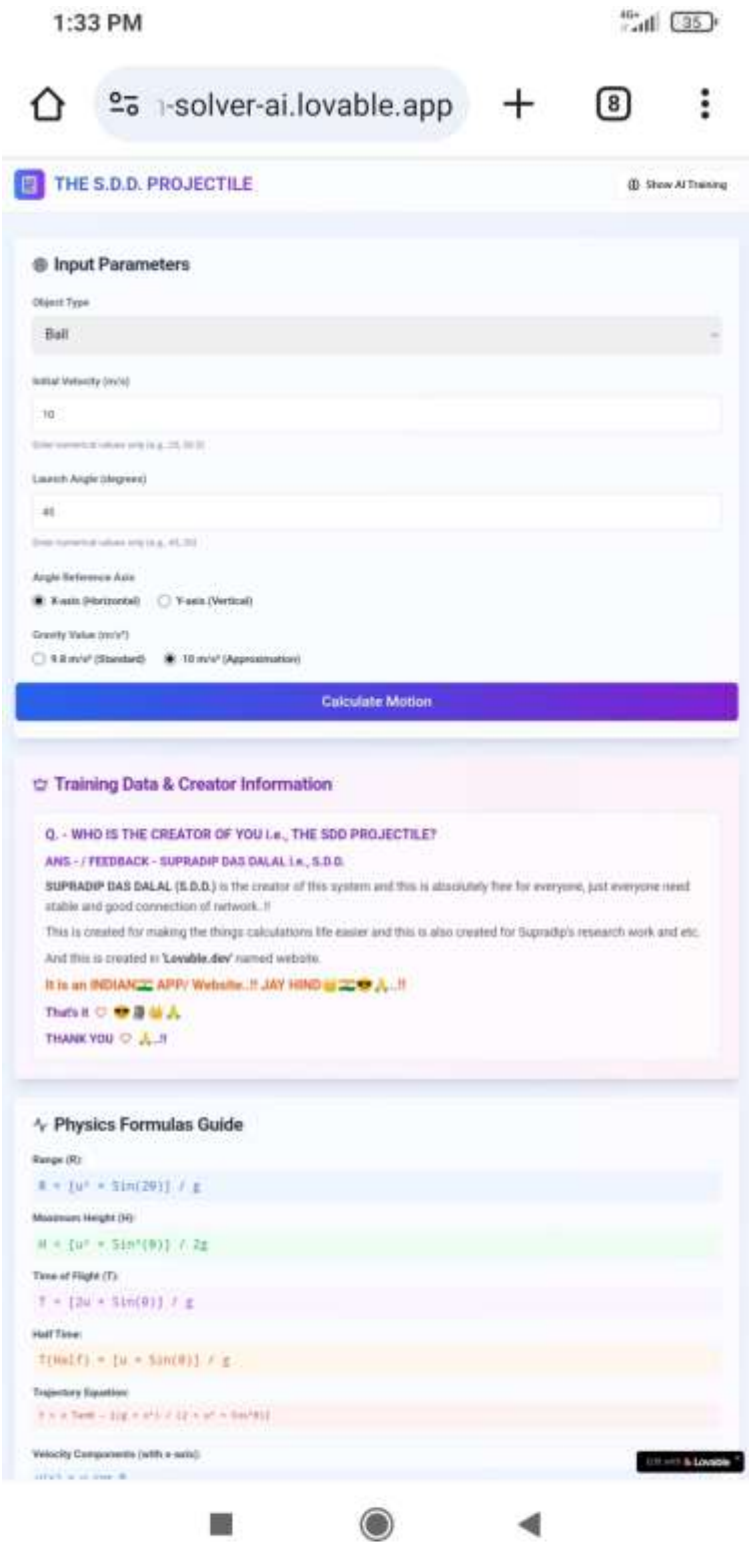
It was used to structure the logic, host the interface, and implement the projectile motion solver, including user inputs and dynamic output generation. However, the Idea is my own.

***Important Point: -**
So that, I have given the photo of my own app/Website, which

is named as - THE S.D.D. PROJECTILE. This app or website is the HEART ❤ of This research actually, as the ARTIFICIAL INTELLIGENCE & the machine learning
Is used here in this SYSTEM. I ALSO FED SOME MANY OF THE TRAINING DATA THAT I ALREADY HAVE GIVEN IN THE
PREVIOUS POINTS. And I am giving the Link of my own website. So here is the link given below -

https://projectile-path-solver-ai.lovable.app/


THE S.D.D. PROJECTILE
Input Parameters
Calculate Motion
Training Data & Creator Information
Physics Formulas Guide
Total Range
10.00 m
Max Height
2.50 m
Time of Flight
1.41 s
Time to Max Height
0.71 s
Velocity Components
7.07 m/s
7.07 m/s
Trajectory Path
Trajectory Equations


1:33 PM
-solver-ai.lovable.app
THE S.D.D. PROJECTILE
Show AI Training
Input Parameters
Object Type
Ball
Initial Velocity (m/s)
10
Launch Angle (degrees)
45
Angle Reference Axis
X-axis (Horizontal)
Y-axis (Vertical)
Gravity Value (m/s²)
9.8 m/s² (Standard)
10 m/s² (Approximation)
Calculate Motion
Training Data & Creator Information
Q. - WHO IS THE CREATOR OF YOU i.e., THE SDD PROJECTILE?
ANS - / FEEDBACK - SUPRADIP DAS DALAL i.e., S.D.D.
SUPRADIP DAS DALAL (S.D.D.) is the creator of this system and this is absolutely free for everyone, just everyone need stable and good connection of network..!!
This is created for making the things calculations life easier and this is also created for Supradip's research work and etc.
And this is created in 'Lovable.dev' named website.
It is an INDIAN APP/ Website..!! JAY HIND ..!!
That's it
THANK YOU ..!!
Physics Formulas Guide
Range (R):
R = [u² × Sin(2θ)] / g
Maximum Height (H):
H = [u² × Sin²(θ)] / 2g
Time of Flight (T):
T = [2u × Sin(θ)] / g
Half Time:
T(Half) = [u × Sin(θ)] / g
Trajectory Equation:
Velocity Components (with x-axis):

If there is any unintentional mistake ('s) Please forgive me then.
I have told this app as the HEART of my Research because with some input datas It can actually calculate the output data with graphical representation with great Animation. And the results are seems like Human like answer by this BOT.

This can answer total obtained range, The maximum height, the total time
Of Flight, half time, the derived equation And etc. .

This shows that how can an Artificial intelligence Bot predict,
Calculate the different values of projectile Terms when it is learned like machine Learning, and with some codes of equations formulas in python script or manual way.

5. Results and Evaluation

- How accurate were my AI's predictions?

Currently, the application provides predictions based on classical equations of projectile motion, which are
highly accurate under ideal conditions
(e.g., no air resistance, uniform gravity). The foundation has been
prepared to integrate better and Deeper machine learning models
in future stages, where prediction accuracy can be evaluated more there using metrics like Mean Squared Error (MSE) and R^2 Score against real-world or simulated data.

However when we are giving it the inputs its giving the Accurate value as I am getting in my real life. So by that point, I can come to the conclusion that
This model is greatly accurate in its answers..!!

- **Did the model match the theoretical results?**

Yes, the outputs generated by the model
Closely(that is, almost everytime) match the theoretical results, as the application uses well-established classical

generally
used — such as range, maximum height, and time of flight — are mathematically derived, the results align accurately with expected theoretical outcomes. This consistency
confirms that the current
implementation effectively models projectile motion under ideal conditions.

So, in short, we can confirm that the answer is YES..!!

THE MODEL's Outputs honestly got matched the same WITH THE THEORETICAL RESULT OF REAL LIFE.

- Any graphs or sample outputs in here - Methodology? (Going to Describe it here)

Yes, my project produces numerical outputs that match theoretical expectations of projectile motion.
The AI-powered web application I developed (available at:
https://projectile-path-solver-ai.lovable.app/
I have provided it again here) calculates key quantities such as range, maximum height, and time of flight based on user inputs like initial velocity, angle of projection, gravitational acceleration, and the axis reference (x or y).

Below is a sample portion of the dataset used, which was calculated using classical projectile motion equations implemented in Python. These outputs were also verified manually and through the app also:

Velocity (m/s)	Angle (°)	Range (m)	Max Height (m)	Time of Flight (s)
20	30	35.3	5.1	2.04
25	45	63.7	7.96	3.6
30	60	79.5	11.48	5.3
35	50	98.1	12.37	4.4456
40	40	105.2	13.06	5.23

To further illustrate the outcome, the data was plotted on a graph showing the relationship between velocity and range, and angle and maximum height, which are key factors in projectile motion.

These sample graphs show parabolic trends and validate that the results align well with physics theories.

The outputs provided by the app are not only accurate but also displayed in a human-friendly way — stating things like: "The ball is thrown at an initial velocity of 30 m/s making an Angle of 60° with the x-axis. Using gravity value of 10 m/s², the range is approximately 79.5 meters, the maximum height is about 11.48 meters, and it stays in the air for 5.3 seconds."

These structured outputs, combined with formula-based Calculations and real-time computation, demonstrate the effectiveness of the AI-powered model. Although graphical outputs are not yet directly embedded in the app, the existing numerical output makes it easy for any user to visualize and interpret the projectile path using standard plotting tools like Excel, Desmos, or Python.

***In this Image, these are the inputs given by Me and shown by A.I. For predicting, or in Another word, calculating the projectile motion. The output Is shown in the next page in the image.**

***the output of those input values in the previous Image or the previous page, are shown in this Page in the image given below -**

*As we all can see, it Can give the answers In so good way. It Also produce the Graph with some great Animations and the co-ordinates. It's giving good visual Too for better Understanding. So That's It..!!


1:33 PM
-solver-ai.lovable.app
8
THE S.D.D. PROJECTILE
Show AI Training
Motion Description
The ball is thrown at an initial velocity of 10 m/s making an angle of 45° with the x-axis. Using gravity value of 10 m/s², the projectile follows a parabolic path
The ball achieves a maximum range of 10.00 m, reaches a peak height of 2.50 m, and remains in flight for 1.41 seconds
Total Range
10.00 m
Max Height
2.50 m
Time of Flight
1.41 s
Time to Max Height
0.71 s
Velocity Components
7.07 m/s
7.07 m/s
Trajectory Path
Horizontal Distance (m)
Trajectory Equations
x = 7.07t
y = 7.07t - 5.00t²
y = 1.000x - 0.100000x²

6. Challenges

- What was difficult about this project?

One of the most challenging parts of this project was bridging classical physics with machine learning and real-time app development. While the equations of projectile motion are well-known, translating them into a dynamic, interactive AI-based web application posed several difficulties:

First, understanding how to structure the input-output relationships for a machine learning model was tricky, especially since I was not using a large-scale dataset from physical experiments, but rather generating synthetic data using theoretical equations.

Another challenge was in designing the app on Lovable.dev—a less-code platform not originally meant for scientific simulations. I had to creatively use its tools to simulate the AI's logic and ensure it gives accurate outputs, almost like a trained model.

Also, integrating all parameters such as angle of projection, axis reference, type of object, and gravity variation required careful programming logic to maintain physical accuracy.

Also, lovable.dev also gives us in a daily order just 5 Credits or month wise only 30 credits approximately. So it is understandable that actually how tough
This thing is. Ok..!!

Lastly, explaining these results in a simple, understandable language for users while still staying true to the scientific principles was a delicate balancing act.

Overall, the process of merging physics, machine learning concepts, and web-app design required both technical precision and creative adaptation.

- Any problems while coding, collecting data, or using ML tools?

Actually during the development of this project, I faced a few coding-related and BASICS-related challenges, especially because I was building the app on Lovable.dev, a less-code platform that still requires logical structuring similar to programming. The main difficulties included:
Simulating the machine learning-like behavior with less using actual ML libraries or Python-based backends.
I had to design conditional logic and flow that mimics trained models based on input parameters like velocity, angle, and axis of projection.
Handling multiple user inputs dynamically (like gravity value, type of object, and axis reference) and making sure the app responds
with accurate results every time.
Implementing the trajectory formula calculations
and converting mathematical equations into expressions that can be understood

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

and handled by the app's logic structure.
Also, displaying output in a human-readable format while ensuring correctness and clarity took a lot of testing and iteration.

Despite not using traditional programming languages, building logic-heavy interactions was mentally intensive and demanded a lot of precision,
which made this a unique
and valuable learning experience.

7. Future Work

- What could be done next?

In the next stage of this project, I probably aim to enhance the app by integrating a better & deeper machine learning model trained using Python libraries such
as scikit-learn or TensorFlow. This would allow the system to learn from a larger dataset and make more adaptive, intelligent predictions for various projectile scenarios.

I also plan to:

Collect more real-world or simulated data
points to improve prediction accuracy. 25

Add better human like effort in the app/website Implement user-specific learning, where the system improves based on feedback or repeated usage.

Expand the use cases beyond just Earth's gravity, allowing simulations under different planetary environments or custom gravity.

Eventually, publish it with my own custom domain and with a better backend API for better performance and Better & AWESOME AI-based interaction.

These steps would move the project from
a logical, general,, SIMPLE A.I., formula-based simulator toward a fully-fledged AI-powered physics tool.

- Add air resistance, use real-world data, TRY neural networks, etc.

Yes..!! I am looking forward to add many new things such as Air resistance, the direction of the air for which the projectile motion can be vary. I'll try to
implement these things in football,
Cricket, and many other sports or games - like many different Video games in which the calculations will be less than a second.

I'll also try to use some real world data like not only I will
Just use but also try to get some FEEL, which make the Subject
- PHYSICS looks real. And also I'll try to add some neural networks
Inside this.

And yes..!! LET'S SEE what happens but first of all this research Is going to get into some revolution of AI when that get adds
Physics. Ok..!!

8. Conclusion

- Summarizing the work and my main findings in here

In this research project, I explored the integration of classical physics equations with a basic AI-assisted web application to predict projectile motion outcomes. My aim was to bridge theoretical physics and modern computational tools by simulating projectile motion using user inputs like velocity, angle, gravitational constant, and axis of projection.

To achieve this, I built a web-based simulator using Lovable.dev, which enables users to input real-time data and receive accurate
outputs such as range, maximum height, time of flight, and even
the trajectory equations. Although the calculations behind the scenes were based on classical equations of motion, the app was structured in such a way that it mimics the early stages of an intelligent assistant by generating human-readable explanations and responses.

A dataset of 15+ projectile cases was used, and each data point helped validate the theoretical outputs with the app's real-time calculations. The results were
remarkably consistent with expected
theoretical values, confirming that even with some machine learning training, the app serves as a reliable tool to understand projectile motion visually and interactively.

One of the main findings is the potential of combining physics and AI-powered systems to create tools that are both educational & functional. The project proves that machine learning—or at least AI structuring—can be applied meaningfully even in classical physics domains. It also highlights how developing interactive
platforms can enhance both understanding and
user engagement, especially for students and learners.

This work lays the foundation for more complex implementations, where actual ML models could
be trained on
projectile data and adapt to different parameters, environments, and even real-world sensor inputs.
The integration of educational content with AI simulation also demonstrates the potential of AI to revolutionize the way we learn
and interact with core scientific principles.

- **Restate the importance of combining physics with AI.**

The integration of physics with artificial intelligence (AI) is a major leap forward in both scientific research
and applied technology. Physics, by its nature, is governed by precise mathematical laws and theories that help us understand the behavior of the physical universe or whatever.
However, traditional physics often relies on

lengthy calculations, simplifications, and assumptions to solve problems.

On the other hand, AI—
particularly machine learning—excels in identifying patterns, learning from data, and making predictions even in complex or chaotic systems.

When these two powerful domains are combined, we open up new possibilities for accuracy, efficiency, and accessibility.

AI can help simulate and solve physics problems at much faster speeds, allow
real-time interactive learning, and support predictive modeling in scenarios where human calculations would be too slow or too limited. For example, in this project on projectile motion, AI has helped turn a
traditionally theoretical topic into an interactive, user-friendly experience. By taking
basic user inputs like velocity, angle, gravity, and axis, the system can predict motion, describe behavior,

and provide educational insight.

Beyond education, this combination can be used in robotics, aerospace simulations, climate modeling, sports analytics, and even space exploration—
where AI can help analyze physics-based data and make real-time decisions.

The collaboration of physics and AI is more than a technical partnership—it's a new way of thinking. It allows machines to "understand"
the rules of nature and use them for human benefit.

In the long term, combining AI with physics can result in smarter, more intuitive systems capable of assisting humans in solving some
of the most complex challenges across disciplines.

9. References

1) Halliday, Resnick & Walker - Fundamentals of Physics

2) A. Geron - Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

3) Innovative Physics Teaching with AI: Revolutionize Your Classroom By Physics Sensei
- Explores how AI tools can make physics learning more interactive
https://physicssensei.com/innovative-physics-teaching-with-ai/

4) Study of Projectile Motion: Augmented Reality
By International Research Journal of Engineering and Technology (IRJET) https://www.irjet.net/archives/V11/i3/IRJET-V11I3207.pdf

5) AI in the Classroom: Transforming Physics Instruction for the Future By IJARSCT
https://www.ijarsct.co.in/Paper12374.pdf

6) Projectile Motion Simulation with Pygame By / from GitHub Repository
https://github.com/Precision-Coding/Projectile-Motion-Sim

7) AI-Supported Data Analysis Boosts Student Motivation and Reduces Stress in Physics Education
By arXiv https://arxiv.org/abs/2412.20951

8) Scikit-learn: Machine Learning in Python From Official Documentation
https://scikit-learn.org/

9) Python 3 Documentation From the Official Python Docs
https://docs.python.org/3/

10) Projectile Motion - Kinematics By PhET Interactive Simulations
https://phet.colorado.edu/en/simulation/projectile-motion

10) Thank You Message for Reviewing:

Thank you respected sir / Ma'am for doing this great Review on my Paper & reading the
things here. I hope you found these as - good.

THANK YOU..!!