Portfolio Report

Full Name: Abdelrahman Tageldin
University ID: AA2200780

Table of Contents

Table of Figures and Tables

Introduction

This portfolio covers a wide range of aspects of Artificial Intelligence, from theoretical perspectives to ethics and then to applications. I will try, through this report, to demonstrate my understanding of some key principles of AI, the analysis of real applications, and reflection on challenges and solutions surrounding AI ethics, knowledge representation, reasoning, and decision-making processes. Besides this, the portfolio tries to emphasize how those learning outcomes have contributed toward my higher-level understanding of how AI brings a revolution, both to society and technology.

TASK 1(Theory)

AI Ethics

Recently, AI usage has been implemented in day-to-day basic things, from using a personal assistant to self-driving cars. AI has an impact on our decision-making, communication, and working methods. However, as this technology grows in capability, it also brings up important moral issues. For example, artificial intelligence (AI) has the potential to transform banking by facilitating quicker and more accurate loan approval decisions, but it also runs the risk of promoting stereotypes or trespassing on patient privacy. These issues are essentially social and ethical in nature rather than merely technological. Privacy, bias, job displacement, and decision-making are the four main ethical issues concerning AI that are looked at in this article. Addressing these concerns is crucial to ensuring that the advantages of AI are shared fairly and do not jeopardize basic rights as these systems become more autonomous and widespread. This debate will demonstrate the critical need for ethical frameworks to direct AI development and application by looking at real-world instances and suggested solutions.

Data Privacy

Data is vital for artificial intelligence, but this reliance creates serious privacy issues. In order to function effectively, AI algorithms in smartphones, social media sites, and security cameras gather enormous volumes of personal data. This data exposes people to potential privacy breaches even while it can enhance user experiences—for example, by offering personalized suggestions. For example, the widespread use of facial recognition technology for security has drawn criticism due to its involvement in widespread surveillance. These tools can be used by private organizations and governments to monitor people without their permission, compromising their right to privacy. A case where Facebook users' data privacy was invaded in 2012 where users' private personal data was collected without their consent by the Cambridge Analytica consulting firm and used for political advertising. Facebook was penalized with a 5 billion dollar for violating consumers' privacy and almost 20 times greater than the largest privacy or data security penalty ever imposed worldwide[1] (FTC, 2019). Therefore, developers need to keep in mind that in order to guarantee that users maintain control over their data, developers must give top priority to building AI systems that respect privacy by design.

Bias in AI

Artificial intelligence bias is a serious ethical issue since it has practical repercussions that impact both people and communities. AI systems may produce discriminating results because they frequently inherent biases from the datasets they are trained on. An example of this happened in 2015 when Amazon was

looking to hire software engineers. They used an AI that checks resumes and returns the top 5 applicants and hire them, the system's design used an AI that taught itself that male applicants more preferred than women applicants and it penalized resumes that included the word women as in "women chess captain" and graduates of all-women universities[2] (Lloyd, 2018). This was caused by the AI observing previous submitted resumes that were accepted and mostly were male applicants. AI bias has also impacted important areas like criminal justice. AI systems used were racial profile people as it marked black offenders as higher risk than other ethnicities even if they have similar criminal records. The algorithm used is named COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) these biases raise moral concerns about justice and injustice and diminish public confidence in AI systems.

Developers need to keep that in mind when using AI, providing it with a varied dataset so that the AI has a clearer image of human complexity. A method to reduce and detect discriminatory patterns can be achieved with routine bias audits. Diverse perspectives, especially those from under-represented groups, can be implemented into the development process to make sure that the AI systems used are fair to all societal members.

Job Displacement

The global market is changing as a result of artificial intelligence, because a lot of jobs can be done without the use of humans as AI can complete tasks much faster and cheaper than humans therefore a lot of people fear the loss of their jobs. From manufacturing to customer service, systems that are implementing AI are quickly automating repetitive jobs and frequently outperforming human work in terms of cost and time. For instance, financial analysts and personal financial advisors are very vulnerable to AI replacing their jobs. These experts deal with a large quantity of numerical data, spotting patterns and suggesting investments. AI is more effective at this as it can forecast better investment, analyze data, and identify trending patterns[3] (Palmer, 2023). In a similar vein, cashiers are being replaced by automated checkout systems in supermarkets, which is causing a large loss of jobs in the industry.

There are serious issues raised by this as the economic disparity is increased as a lot of people can't compete with AI, and they can't afford to learn skills needed to use AI. For example, even though AI is creating high-skilled employment in data analysis and programming, those without academic degrees are usually unable to obtain these jobs. Because employment and social well-being are strongly related, job displacement has an impact on workers and causes a lot of harm as they can't find jobs with the skills they had before AI. To address this problem and avoid a rise in unemployment percentages, governments and businesses should assist employees and citizens in moving into jobs that require human creativity and problem-solving abilities, which AI cannot replace.

Conclusion

To conclude this essay,AI has increased my awareness of the opportunities and difficulties of ai tech.the most important lesson i learnt is that in order to benefit from ai i must find a balance and avoid breaking moral standards .AI has the ability to change the world for the better by enhancing productivity and solving some of the biggest challenges humankind has faced but AI has raised serious moral and ethical questions. Governments, businesses, and individuals should work together to make certain rules when using AI to avoid the consequences of AI. The ways we use AI will determine whether AI becomes a tool for fair improvement in work or a cause of problems in societies.

Reference

[1] Federal Trade Commission. (2019, July 24). FTC imposes $5 billion penalty and sweeping new privacy restrictions on Facebook. Federal Trade Commission Press Release. Retrieved from https://www.ftc.gov/news-events/news/press-releases/2019/07/ftc-imposes-5-billion-penalty-sweeping-new-privacy-restrictions-facebook

[2] Lloyd, J. (2018, October 10). Insight: Amazon scraps secret AI recruiting tool that showed bias against women. Reuters. Retrieved from https://www.reuters.com/article/world/insight-amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK0AG

[3] Palmer, E. (2023, June 20). The jobs most at risk from AI—and the employment tasks AI can't replace. Newsweek. Retrieved from https://www.newsweek.com/jobs-most-risk-ai-employment-tasks-artificial-intelligence-1909671

4.IBM. (2023, October 16). *Shedding light on AI bias with real world examples*. Retrieved December 7, 2024, from https://www.ibm.com/think/topics/shedding-light-on-ai-bias-with-real-world-examples

5.SEO.ai. (n.d.). *AI replacing jobs: Statistics and trends*. Retrieved December 7, 2024, from https://seo.ai/blog/ai-replacing-jobs-statistics

6.Velaro. (n.d.). *The privacy paradox of AI: Emerging challenges on personal data*. Retrieved December 7, 2024, from https://velaro.com/blog/the-privacy-paradox-of-ai-emerging-challenges-on-personal-data

Task 3(Practical)

Semantic web

The semantic web is an addition to the existing web that makes it possible for data to be shared and utilized by many communities, businesses, and apps. It's made so it can make data linked and machine-readable. This allows computers to share and interpret information in a way that is useful choices. The web is currently split into 3 types of web 2.0, web 3.0 and web 4.0.

Web 2.0: it emphasizes user usability and interoperability and mainly works with user generated content, for example a picture shared on social media, and it allows users to participate and interact with each other online like on social media and blogs.

Web 3.0: It's known as the semantic web. It prioritizes smart information processing like for example Ai driven tools like ChatGPT and blockchain and cryptocurrency like Bitcoin and Ethereum transactions. It utilizes machine readable data.

Table 1:Web generation comparison

Web generation	Technologies Used	Applications

Web 2.0	- AJAX (Asynchronous JavaScript and XML) - RSS Feeds - APIs for service integration - JavaScript frameworks (e.g., jQuery) - Eclipse IDE	- Social media platforms (Facebook, Twitter) - Blogging platforms (WordPress, Blogger) - Collaborative projects (Wikipedia) - Video sharing sites (YouTube)
Web 3.0	- Semantic Web technologies (RDF, OWL, SPARQL) - Artificial Intelligence (AI) and Machine Learning (ML) - Natural Language Processing (NLP) - Blockchain technology	- Intelligent virtual assistants (Siri, Alexa) - Personalized recommendation systems (Netflix, Spotify) - cryptocurrency platforms (Ethereum)
Web 4.0	- Advanced AI and ML algorithms - Internet of Things (IoT) - Virtual Reality (VR) and Augmented Reality (AR)	- Smart environments (smart homes, smart cities) - Autonomous vehicles (self-driving cars) - Immersive VR and AR experiences (education, gaming)

**Note.* Adapted from Rouse (2020).*

Is Ai considered a Semantic web?

No, AI is not a semantic web itself but it plays a crucial role in it as semantic webs uses AI technology to process and make a large amount of data machine readable and then AI use that data to interpret the context and the meaning of the information provided by the structure that was given semantic web.

In conclusion, the semantic web is a an evolved version of the web where it process data and structures it so machines can understand it.

Web2.0 vs Web3.0 vs Web4.0:

- Web 2.0 focuses on user activity between each other.
- Web 3.0 emphasizes smart data interpretation and AI assistance.
- Web 4.0 it’s a concept of the future where intelligent web interactions are made seamlessly with other users and devices.

Figure 1:Ontology

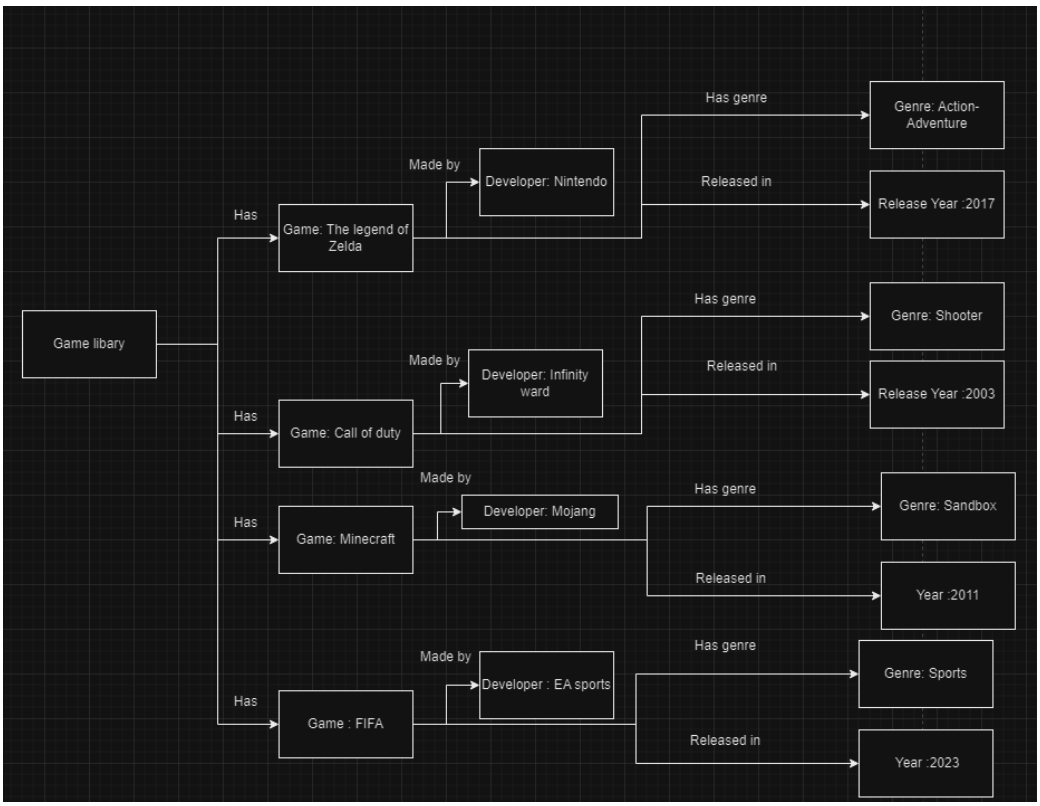

Figure 2:Semantic Web Code 1

```python
from rdflib import Graph, URIRef, Literal, Namespace, RDF

# Initialize the RDF graph
g = Graph()

# Define a namespace for our ontology
ex = Namespace("http://example.org/")

# Define game entities
game1 = URIRef("http://example.org/game/The_Legend_of_Zelda")
game2 = URIRef("http://example.org/game/Call_of_Duty")
game3 = URIRef("http://example.org/game/Minecraft")
game4 = URIRef("http://example.org/game/FIFA")

# Define developer entities
developer1 = URIRef("http://example.org/developer/Nintendo")
developer2 = URIRef("http://example.org/developer/Infinity_Ward")
developer3 = URIRef("http://example.org/developer/Mojang_Studios")
developer4 = URIRef("http://example.org/developer/EA_Sports")

# Define genre entities
genre_action_adventure = URIRef("http://example.org/genre/Action-Adventure")
genre_shooter = URIRef("http://example.org/genre/Shooter")
genre_sandbox = URIRef("http://example.org/genre/Sandbox")
genre_sports = URIRef("http://example.org/genre/Sports")

# Define year entities
year_2017 = URIRef("http://example.org/year/2017")
year_2003 = URIRef("http://example.org/year/2003")
year_2011 = URIRef("http://example.org/year/2011")
year_2023 = URIRef("http://example.org/year/2023")

# Adding triples for the first game
g.add((game1, RDF.type, ex.Game))  # The Legend of Zelda is a Game
g.add((game1, ex.title, Literal("The Legend of Zelda: Breath of the Wild")))  # Title of the game
g.add((game1, ex.developedBy, developer1))  # Developed by Nintendo
g.add((game1, ex.hasGenre, genre_action_adventure))  # Belongs to the Action-Adventure genre
g.add((game1, ex.releasedIn, year_2017))  # Released in 2017
```

Figure 3:Semantic Web Code 2

```python
# Serialize and print the graph in Turtle format
print(g.serialize(format="turtle"))

# Updated SPARQL query to retrieve games released before 2015
query = """
    PREFIX ex: <http://example.org/>  # Define the namespace prefix for our ontology

    SELECT ?gameTitle ?developerName ?genreName ?yearValue  # Select the variables to retrieve
    WHERE {
        ?game ex:title ?gameTitle .                    # Match the game and its title
        ?game ex:developedBy ?developer .              # Match the game and its developer
        ?developer ex:name ?developerName .            # Retrieve the developer's name
        ?game ex:hasGenre ?genre .                     # Match the game and its genre
        ?genre ex:name ?genreName .                    # Retrieve the genre's name
        ?game ex:releasedIn ?year .                    # Match the game and the year entity
        ?year ex:yearValue ?yearValue .                # Retrieve the numeric value of the year
        FILTER(?yearValue < 2015)                      # Filter to include only games released before 2015
    }
"""

# Execute the query and print results
for row in g.query(query):
    # Print each game's details: title, developer, genre, and release year
    print(f"Game: {row.gameTitle}, Developer: {row.developerName}, Genre: {row.genreName}, Year Released: {row.yearValue}")
```

Figure 4:Semantic Web Code Output

```
@prefix ns1: <http://example.org/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://example.org/game/Call_of_Duty> a ns1:Game ;
    ns1:developedBy <http://example.org/developer/Infinity_Ward> ;
    ns1:hasGenre <http://example.org/genre/Shooter> ;
    ns1:releasedIn <http://example.org/year/2003> ;
    ns1:title "Call of Duty" .

<http://example.org/game/FIFA> a ns1:Game ;
    ns1:developedBy <http://example.org/developer/EA_Sports> ;
    ns1:hasGenre <http://example.org/genre/Sports> ;
    ns1:releasedIn <http://example.org/year/2023> ;
    ns1:title "FIFA" .

<http://example.org/game/Minecraft> a ns1:Game ;
    ns1:developedBy <http://example.org/developer/Mojang_Studios> ;
    ns1:hasGenre <http://example.org/genre/Sandbox> ;
    ns1:releasedIn <http://example.org/year/2011> ;
    ns1:title "Minecraft" .

<http://example.org/game/The_Legend_of_Zelda> a ns1:Game ;
    ns1:developedBy <http://example.org/developer/Nintendo> ;
    ns1:hasGenre <http://example.org/genre/Action-Adventure> ;
    ns1:releasedIn <http://example.org/year/2017> ;
...


Game: Call of Duty, Developer: Infinity Ward, Genre: Shooter, Year Released: 2003
Game: Minecraft, Developer: Mojang Studios, Genre: Sandbox, Year Released: 2011
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Reference list:

Rouse, M. (2020). *Web 2.0 (or Web 2)*. TechTarget. https://www.techtarget.com/whatis/definition/Web-20-or-Web-2

TASK 4(Theory)

Comparison of Production Rules and Frames in Knowledge Representation

Knowledge Representation:

Knowledge representation is a big component of AI, allowing machines to reason intelligently, as would humans. It achieves this through the organization of data into formats that are machine-readable. The two most important ways in which AI systems organize data are through production rules and frames. Each has an area in which it is applied best; some are better in contexts than others.

This comparison will highlight their advantages and disadvantages and which scenarios each is best in.

Production Rules:

Production rules are a type of knowledge representation that utilize conditional statements to specify which actions will be taken when certain conditions are satisfied. They have the form of [IF {condition} Then {action}].

Advantages:

- Easy to understand and change.
- Good for making decisions based on changing data.
- Rules work independently of each other.

Weaknesses:

- Tends to run slow if there are a lot of rules
- Hard to decide which rule to apply when multiple rules fit.
- Doesn't show relationships between different pieces of data.

Example Use:

- Medical diagnosis systems that suggest possible illnesses based on symptoms.

Frames:

Frames are templates that store details about things or ideas. They have spaces for various values and qualities.

Advantages:

- Organizes the information in a way that has structure.
- Supports inheritance, where specific frames may build on general ones.
- Suitable for default values when information that is incomplete is mentioned.

Weaknesses:

- Not so flexible to change or procedural information.
- It might become complicated with many interconnected frames.
- Conflicting cases are possible using inherited information.

Example Use:

- Organizational chart modeling of departments and employee roles.

Table 2:Production Rules vs Frames

Aspect	Production Rules	Frames
Definition	Use "IF condition THEN action" statements to decide actions when conditions are met	Use data structures with slots for attributes and values to represent objects or concepts
Knowledge Type	Procedural knowledge (how to do things)	Declarative knowledge (facts and information)
Structure	Flat structure; each rule works independently	Hierarchical structure; supports inheritance and relationships
Strengths	- Easy to understand and modify- Good for dynamic decision-making- Flexible with changing conditions	- Organizes information well- Shows relationships between concepts- Handles default values
Weaknesses	- Can slow down with many rules- Hard to resolve conflicts when multiple rules apply- Lacks relational structure	- Less flexible for procedural knowledge- Can get complex with many frames- Inheritance conflicts can occur
Suitable For	Systems that need decisions based on changing conditions, like medical diagnosis	Modeling structured information with relationships, like organizational charts
Example Use	Medical system: "IF patient has fever THEN suspect infection"	Employee frame with slots: name, position, department
Scalability	May become inefficient with many rules	Can become complex with many frames and relationships
Modification	Easy to add or change rules	Changes can affect multiple frames due to inheritance
Inference	Uses rule chaining (forward and backward)	Uses slot filling and inheritance

References List:

1. GeeksforGeeks. (n.d.). *Knowledge Representation in AI*. Retrieved December 5, 2024, from https://www.geeksforgeeks.org/knowledge-representation-in-ai/

2. JavaTpoint. (n.d.). *AI - Techniques of Knowledge Representation*. Retrieved December 5, 2024, from https://www.javatpoint.com/ai-techniques-of-knowledge-representation

Task 6(Practical)

Markov decision processes

Markov decision processes (MDP) is a mathematical frame used to model decision-making scenarios where outcomes a random and under control of a decision maker. This project implements MDP in dynamic programming method to find a path for a robot from start state to end state in a grid with obstacles.

Formulation of MDP:

A grid size of 6x6 was used that included obstacles and allowed movements were up, down, left and right. A reward function was also used giving a +10 for reaching the goal stat and -1 for each other step. Along with a discount factor of Gamma it was used to encourage the robot by giving it small rewards, in this case 0.9.

Implementation details:

The visualization was made using Pygame. It helped in visualizing the states obstacles and start and end state and path taken. The optimal path taken was calculated based on maximizing the value function at each state.

Testing and results:

Different size grids were used with varying obstacles start and end states in each case the program correctly found the optimal path also tested that the visual path is correct.

Figure 5:MDP Code

```python
def reward(state):
    # Define the reward function.
    # The agent receives a positive reward if it reaches the goal state, and a negative reward otherwise.
    return 10 if state == goal_state else -1

Tabnine | Edit | Test | Explain | Document | Ask
def transition(state, action):
    # Defines the next state given the current state and an action.
    # The agent moves one cell in the specified direction if possible.
    # If the move is blocked by an obstacle or out of the grid bounds, the state does not change.
    if state in obstacles:
        return state  # No movement allowed if currently on an obstacle (though ideally agent shouldn't be there)
    x, y = state
    # Check the action and move accordingly if valid
    if action == "UP" and x > 0 and (x - 1, y) not in obstacles:
        return (x - 1, y)
    if action == "DOWN" and x < ROWS - 1 and (x + 1, y) not in obstacles:
        return (x + 1, y)
    if action == "LEFT" and y > 0 and (x, y - 1) not in obstacles:
        return (x, y - 1)
    if action == "RIGHT" and y < COLS - 1 and (x, y + 1) not in obstacles:
        return (x, y + 1)
    # If no valid move, remain in the same state
    return state

Tabnine | Edit | Test | Explain | Document | Ask
def value_iteration(states, actions, theta=1e-4, gamma=0.9):
    # Perform value iteration to compute the value function for each state.
    # The value function V(s) represents the maximum expected reward achievable from state s.
    # gamma is the discount factor, theta is the threshold for stopping.
    V = {state: 0 for state in states}  # Initialize all state values to 0
    while True:
        delta = 0  # Tracks the maximum difference between new and old values in this iteration
        for state in states:
            if state in obstacles:
                # Skip updating values for obstacles as the agent shouldn't move onto them
                continue
            v = V[state]  # Current value of the state
            # Compute the new value as the maximum over all actions of [reward(s) + gamma * V(s')]
            # where s' is the next state after taking that action
            V[state] = max(reward(state) + gamma * V[transition(state, action)] for action in actions)
            delta = max(delta, abs(v - V[state]))  # Update delta with the change in value
        # Stop iterating once values have converged (change is less than theta)
        if delta < theta:
            break
    return V

Tabnine | Edit | Test | Explain | Document | Ask
def find_optimal_path(start_state, values):
    # Given the computed value function, find the optimal path from start_state to goal_state.
    # The path is obtained by repeatedly choosing the action that leads to the state with the highest value.
    current_state = start_state
    path.clear()
    path.append(current_state)
    while current_state != goal_state:
        # Choose the action that leads to the highest value in the next state
        best_action = max(actions, key=lambda action: values[transition(current_state, action)])
        next_state = transition(current_state, best_action)
        if next_state == current_state:
            # No further movement possible; break out if stuck
            break
        path.append(next_state)
        current_state = next_state
```

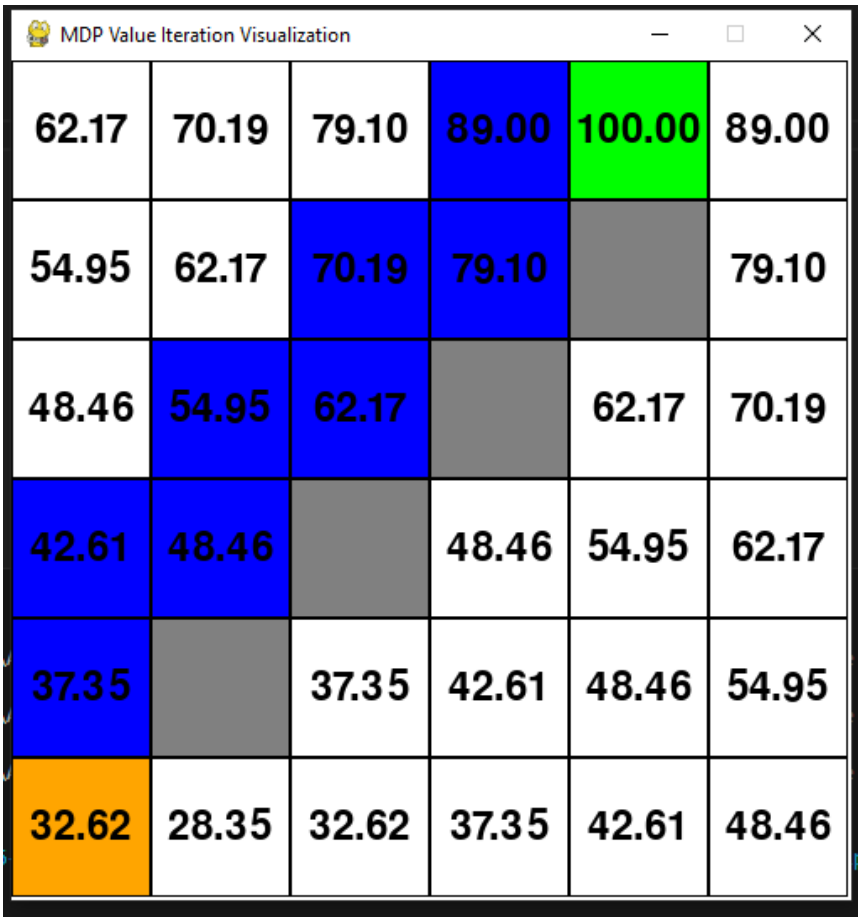

Figure 6:MDP Test Case 1

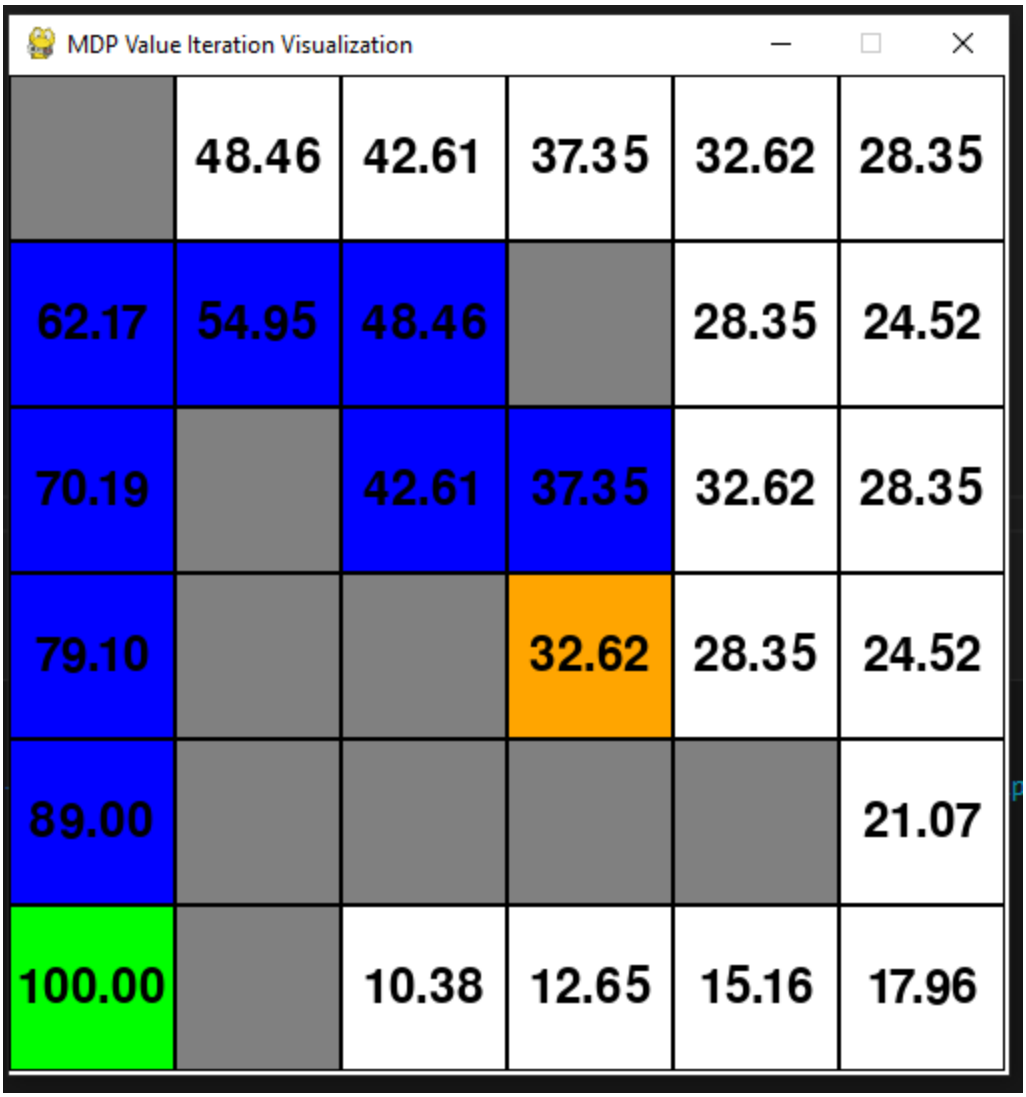

Figure 7:MDP Test Case 2

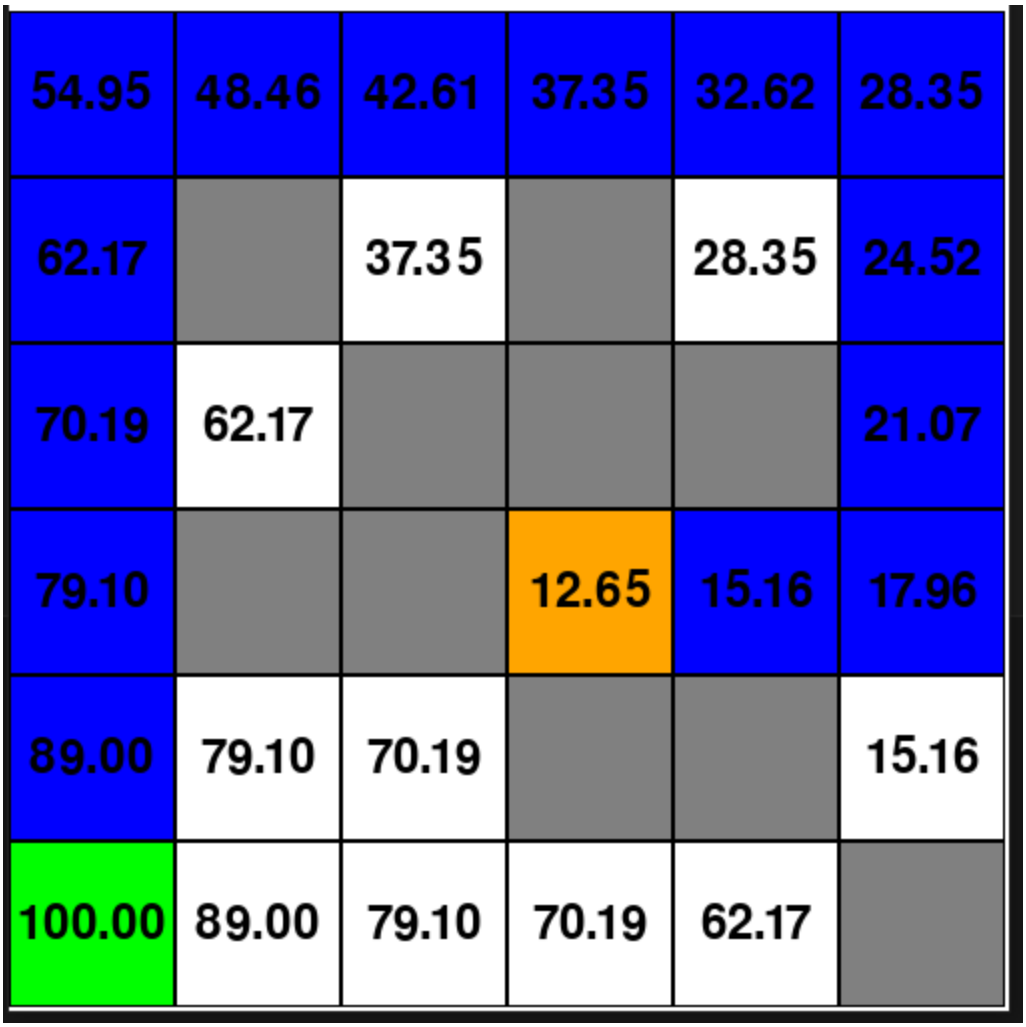

Figure 8:MDP Test Case 3

Task 11(practical)

Binary Classification: Spam Detection

Support vector machine (SVM) is an AI algorithm used for regression and classification tasks that works by spreading the data in unique categories then finds the optimal decision boundary called hyperplane. Using the support vector machines algorithm in this case to do binary classification for a spam detection AI.

```
# Evaluate how well the model performed using accuracy_score
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy * 100:.2f}%")
```
[41] ✓ 0.0s

```
Model Accuracy: 98.57%
```

```
# Test the model with a custom tweet
test_msg = ["Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's"]

# Transform the message using the trained CountVectorizer
test_msg_transformed = cv.transform(test_msg)

# Predict the sentiment of the custom message
print("Custom Message Prediction:")
print(svm_model.predict(test_msg_transformed))
```
[42] ✓ 0.0s

```
Custom Message Prediction:
['spam']
```

```
# Load another dataset named "test.csv" containing tweets to predict sentiment
test_dataset = pd.read_csv("test.csv")

# Extract features from the test dataset
test_x = test_dataset.iloc[:, 1].values  # Assuming the tweets are in the second column

# Transform the test data using the CountVectorizer
test_x_transformed = cv.transform(test_x)

# Make predictions using the trained Support vector machines
test_predictions = svm_model.predict(test_x_transformed)

# Replace numerical class labels with text labels
text_test_predictions = ["Non Spam" if prediction == "ham" else "Spam" for prediction in test_predictions]

# Create a DataFrame to display the results
results_df = pd.DataFrame({
    'Tweet': test_dataset.iloc[:, 1],  # Original tweets
    'Predicted_Label': text_test_predictions
})

# Display the results DataFrame
print("Predictions on New Test Dataset:")
print(results_df)
```
[43] ✓ 0.9s

```
Predictions on New Test Dataset:
                                               Tweet Predicted_Label
0      #studiolife #aislife #requires #passion #dedic...        Non Spam
1       @user #white #supremacists want everyone to s...        Non Spam
2      safe ways to heal your #acne!!    #altwaystohe...        Non Spam
3      is the hp and the cursed child book up for res...        Non Spam
4        3rd #bihday to my amazing, hilarious #nephew...        Non Spam
```

Figure 9:Binary Classification Code

Reference list:

Task 9 (Theory)

Medical Expert System Development

Expert system is a program that is designed to think like humans and make decision like a human that is expert in the field it will be used in. Expert systems are made from 3 important components firstly is Knowledge base which is rules that are structed as IF-Then statements. Then comes the inference engine which processes the rules found in the knowledge base then it compares the inputs to the rules and draws a conclusion based on the matches. finally comes the user interface which allows the user to input symptoms which are then send to the inference engine, the user interface also displays the final diagnosis. In this case, we are designing a system that diagnoses medical conditions based on syptomps.it uses Rule-Based approach to match symptoms to possible diseases.

Advantages of Medical expert system:

- Consistency: provides the same diagnosis every time the same symptoms are inputted.
- Accessibility: people with no medical knowledge can figure out their diagnosis.
- Efficiency: Much faster at finding a person diagnose than humans.

Limitations:

- Scope: its limited to the conditions and symptoms in the system.
- No personalization: cannot adapt or patient histories or previous diagnosis.
- Accuracy: it depends solely on the rules and may not account for complex conditions

Steps to implement medical expert system:

1. we need to define the scope of the programs by deciding on what medical conditions the system will cover in this case we will focus on FLU, Common cold, Migraine and measles.
2. we need to collect knowledge in this case we will use reliable source to understand symptoms for each condition. In this case we used 3 reliable sources from Mayo clinic, World Health Organization and centers for disease control and prevention (CDC).
3. Then we create rules in the form of If Then format.
    a. Example:
        i. Rule 1: If fever and Cough Then possible diagnosis is Flu.
        ii. Rule 2: If Headache, nausea and light sensitivity then Migraine.
4. Build the inference engine, we will use forward-chaining .forward-chaining work by starting at the symptoms and moving toward the possible outcome in this case diagnoses.
5. Finaly, we need to test to make sure the input of different combinations of symptoms lead to the accurate diagnoses.

Rules:

1. Flu
    a. Symptoms:
        i. Fever,Cough,Fatigue,Sore Throat.
    b. Rule:
        i. IF Fever and Cough,THEN diagnosis is Flu.

2. Migraine
    a. Symptoms:
        i. Headache,Light sensitvity,Nausea.
    b. Rule:
        i. IF Headache,Light sensitvity,Nausea ,THEN diagnosis is Migraine.
3. Common Cold
    a. Symptoms:
        i. Runny nose,Sore Throat,Sneezing,Mild Cough.
    b. Rule:
        i. IF Sore Throat and Sneezing ,THEN diagnosis is Common Cold.
4. Measles
    a. Symptoms:
        i. Fever,Rash,Red Eyes.
    b. Rule:
        i. IF Fever and Rash ,THEN diagnosis is Measles.

Reference List:

1. Centers for Disease Control and Prevention. (n.d.). Common cold: About. Retrieved December 6, 2024, from https://www.cdc.gov/common-cold/about/index.html

2. Mayo Clinic. (n.d.). Migraine: Symptoms and causes. Retrieved December 6, 2024, from https://www.mayoclinic.org/diseases-conditions/migraine-headache/symptoms-causes/syc-20360201

3. World Health Organization. (n.d.). Influenza (seasonal). Retrieved December 6, 2024, from https://www.who.int/news-room/fact-sheets/detail/influenza-(seasonal)

4. World Health Organization. (2024, November 14). Measles. Retrieved December 6, 2024, from https://www.who.int/news-room/fact-sheets/detail/measles

5. Javatpoint. (n.d.). Expert systems in artificial intelligence. Retrieved December 6, 2024, from https://www.javatpoint.com/expert-systems-in-artificial-intelligence

Task 10 (Theory)

Reasoning and Problem-Solving Techniques

Reasoning in AI is the process of solving problems by drawing conclusions based on info gained and logic. There are different reasoning approaches that are used to solve problems in different scenarios, the reasoning methods that will be compared in this essay will be heuristic reasoning and Bayesian reasoning. This essay compares the 2 different methods and explores their core principles and analyzes how they are used in problem-solving with AI.

Heuristic Reasoning

The fundamental idea of heuristic reasoning is the process of finding quick solutions by applying experience-based strategies and basic logic. It does not guarantee the best and most efficient solution but provides an answer in reasonable time. It works by breaking down complex problems into smaller parts; it also prioritizes speed over in-depth analysis. It solves problems by relying on previous experiences. Heuristic usually goes for a good enough rather than the best solution possible.

Uses in AI

In AI, heuristic reasoning is used in:

- Searching algorithms: Methods like A* algorithm use heuristics to estimate the best path to a target state.
- Optimization problems: Methods or algorithms to find satisfactory solutions in large search spaces.

Bayesian Reasoning

The fundamental of Bayesian reasoning is an approach based on probabilities; it involves a mathematical formula to update the probability estimate for a hypothesis as new evidence or new information becomes available. Bayesian usually requires a large amount of data as it calculates the probability; it also prioritizes accuracy.

Uses in AI

In AI, Bayesian reasoning is used in:

- Machine Learning Algorithms: Used in Naïve Bayes classifiers and Bayesian regression uncertainty in predictions.
- Probabilistic models: Used in Bayesian networks to represent variables and their dependencies for reasoning under uncertainty.

Table 3:Heuristic Vs Bayesian

Aspect	Heuristic Reasoning	Bayesian Reasoning
Definition	Experience-based shortcuts for quick decision-making.	Probabilistic approach updating beliefs with new evidence.

Core Principle	Simplification and efficiency over exhaustive analysis.	Bayes' Theorem for probabilistic inference and updating.
Approach	Uses rules of thumb and intuition.	Relies on mathematical probability and statistical data
Data Requirements	Can operate with limited or qualitative data.	Requires quantitative data for probability assessments.
Handling Uncertainty	Often overlooks uncertainties due to simplification.	Explicitly quantifies and incorporates uncertainty.
Accuracy vs. Efficiency	Prioritizes speed; may sacrifice accuracy.	Prioritizes accuracy; may require more computation.
Cognitive Biases	Susceptible to biases and errors due to oversimplification.	Aims to minimize biases through formal updating; priors may bias.
AI Applications	Search algorithms, optimization problems, expert systems.	Probabilistic models, machine learning, robotics.
Strengths	Fast, simple, and practical for immediate problem-solving.	Accurate, robust to uncertainty, and continuously learning.
Limitations	May not find optimal solutions; can overfit to known scenarios.	Computationally intensive; sensitive to prior assumptions.
Best Used When	Quick decisions are needed; limited data; low-stakes situations.	Precision is crucial; ample data; managing uncertainty is key.

Reference list:

1. Mukherjee, A., & Chang, H. H. (2024). *Heuristic reasoning in AI: Instrumental use and mimetic absorption*. arXiv. Retrieved from https://arxiv.org/abs/2403.09404

2. GeeksforGeeks. (2023). *Bayes' theorem in artificial intelligence*. Retrieved from https://www.geeksforgeeks.org/bayes-theorem-in-artificial-intelligence/

3. GeeksforGeeks. (2023). *Heuristic function in AI*. Retrieved from https://www.geeksforgeeks.org/heuristic-function-in-ai/

4. GeeksforGeeks. (2023). *Applications of Bayes' theorem in artificial intelligence*. Retrieved from https://www.geeksforgeeks.org/applications-of-bayes-theorem-in-artificial-intelligence/

Task 12 (Practical)

Figma link: https://www.figma.com/proto/A1Xsgp8Jfggd8KKV1MMuVq/Fincaial-advisor?node-id=1-30&starting-point-node-id=1%3A21&t=xiIIXQHltSYBkETh-1

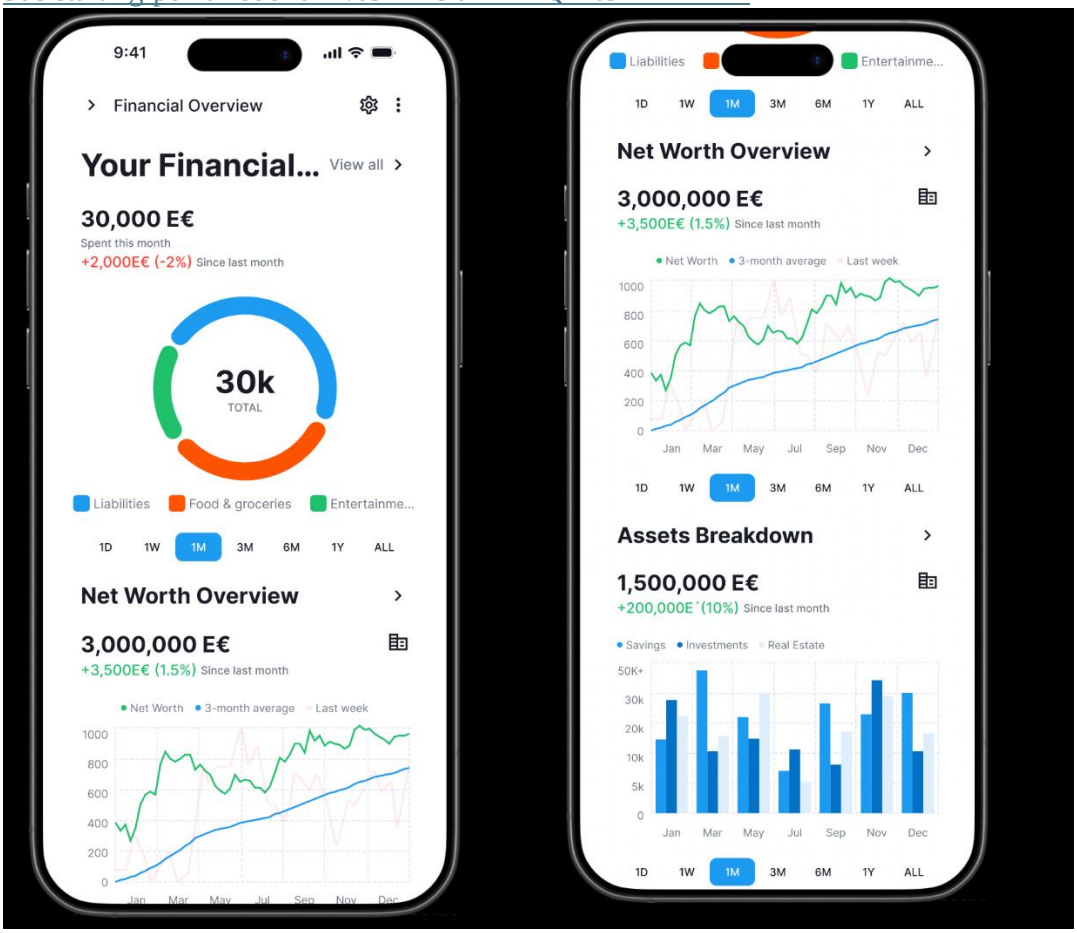

Figure 10:Figma 1

Figure 11:Figma 2

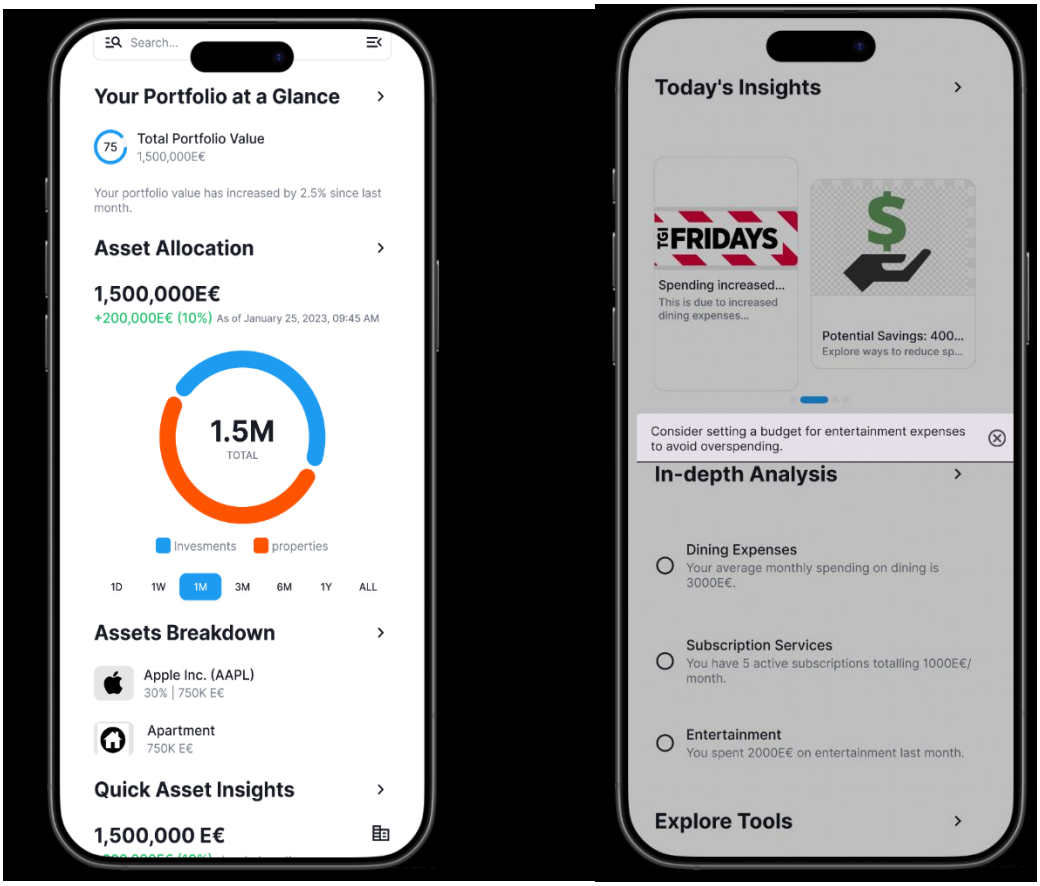

Figure 12:Figma 3

Figure 13:Figma 4

Conclusion

To conclude this essay, AI has increased my awareness of the opportunities and difficulties of AI tech.The most important lesson I learnt is that in order to benefit from ai I must find a balance and avoid breaking moral standards .AI has the ability to change the world for the better by enhancing productivity and solving some of the biggest challenges humankind has faced but AI has raised serious moral and ethical questions. Governments, businesses, and individuals should work together to make certain rules when using AI to avoid the consequences of AI. The ways we use AI will determine whether AI becomes a tool for fair improvement in work or a cause of problems in societies.