Include a Running head: OPTIMIZING PICKING ROUTE FOR IMPROVING WAREHOUSE EFFICIENCY

Optimizing picking route for improving warehouse efficiency

Kannan Nova

Professor/Manager: Dr. Michele Bennett

Master of Science in Data Science

College of Science, Engineering, and Technology

Grand Canyon University

08/02/2021

[Copyright Note – *Optional*]

[Legal Notice – *Optional*]

Table of Content

[**Optimizing picking route for improving warehouse efficiency** 6](#_Toc80203354)

[Project Overview and Project Objectives 6](#_Toc80203355)

[Project Scope 7](#_Toc80203356)

[Project Completion 7](#_Toc80203357)

[Project Controls 8](#_Toc80203358)

[Project Schedule 8](#_Toc80203359)

[Timeline and scheduled activities 8](#_Toc80203360)

[Project Cost Estimate (or Alternative Criteria) 10](#_Toc80203361)

[Project Issue Log 10](#_Toc80203362)

[Requirements Analysis: Use Cases 11](#_Toc80203363)

[Requirements Analysis: System Design 11](#_Toc80203364)

[Requirements Analysis: Technical Requirements 13](#_Toc80203365)

[Requirements Analysis: Data Science Model Design 13](#_Toc80203366)

[Requirements Analysis: Reports 14](#_Toc80203367)

[Requirements Analysis: Screen Definitions and Layouts 16](#_Toc80203368)

[Requirements Analysis: Security 16](#_Toc80203369)

[Detailed Model Pipeline Design 17](#_Toc80203370)

[System Entities 28](#_Toc80203371)

[Functional Requirements 29](#_Toc80203372)

[Source Code Listing 30](#_Toc80203373)

[Code Review 36](#_Toc80203374)

[Implementation 36](#_Toc80203375)

[Security 46](#_Toc80203376)

[Completion OR Application Functionality and Execution 48](#_Toc80203377)

[Performance Analysis 61](#_Toc80203378)

[User Guide 68](#_Toc80203379)

[System Administration Guide 71](#_Toc80203380)

[Digital Poster 75](#_Toc80203381)

[Project Presentation Phase 77](#_Toc80203382)

[Final Project Submission 77](#_Toc80203383)

[GitHub Repository 77](#_Toc80203384)

[Industry Terminology 77](#_Toc80203385)

[Conclusion 79](#_Toc80203386)

Optimizing picking route for improving warehouse efficiency

### Project Overview and Project Objectives

#### Overview

The warehouse is a part of supply chain logistics industry where we store the goods and bring back it from storage for shipment.

Today’s consumers are shopping online more than store purchase and 31% of consumers purchase online weekly as per “Global Consumer Insights 2019” report from PWC and this rate will keep increase (Davis, 2019). They expect the goods within a couple of days. This shift / expectation increase pressure on warehouse facilities to meet this new demand, so we need to optimize each and every warehouse operation to achieve the efficiency and meet consumer’s demand. Warehouse has two major core processes such as

Put away where we download the goods from truck and place it into final location inside warehouse (McCrea, 2016).

Pickingwhere we take out the goods from final location and stuff into truck for shipping (Lopienski, 2020). It is simply the process of pulling out the right products from a warehouse for an order.

These two warehouse operations are very complex and more labor-oriented tasks. The workers manually go there and pick the goods if it is a small and light weight otherwise, they operate forklifts for picking. Sometimes robots are used for picking. If the system is not providing the proper optimal route/shortest path details for picking a particular order, then workers spend more time on walking instead of performing the actual picking job.

Based on literature reviews, the picking process is optimized by linearly or deterministic or deep learning, but it is a dynamic and stochastic because goods can be placed anywhere in warehouse during put away/storing (Janse, 2019). Thus, these enable to use reinforcement learning (RL) and warehouse floor layout can easily be converted into RL’s state data because warehouse floor typically has rack/row and bin/column for storage location as matrix format and storage location can be identified by aisle, rack and bin (LeBlanc, 2020).

It also shows that the optimizing and improving picking process will improve / cause warehouse efficiency positively. This project is based on dynamic environment and if picking process failed, system/agent will learn it from failure for future improvement and it needs the data for environment, state, Q-table and locations (Violante, 2019).

#### Objectives

This capstone project demonstrates an approach to find the optimal shortest path for picking in warehouse using reinforcement q-learning algorithms and environment, states, q-values, actions and rewards are key components of this project (Bhatt 2019).

### Project Scope

This project addresses only picking operation even if warehouse has a lot of operations such as put away, inventory management, cross docking, zero inventory and just in time inventory.

#### Assumptions

It picks one order at a time from one location. It works for warehouse where its layout or locations can be derived as grid/matrix format. RL requires states or locations.

### Project Completion

The project has four major milestones such as milestone 1, milestone 2, milestone 3 and milestone 4. In milestone 1, the requirement analysis, project title selection, and proposals are done. In milestone 2, model design, data collection, and preprocessing are done. In milestone 3, the code execution, and deployment are done. In milestone 4, project submission and presentation are done.

### Project Controls

There are a few risks for this project and first major risk is that getting the warehouse layout data which is needed to create states in environment for reinforcement model. Normally warehouse keeps the transactional data such as locations, products/SKUs but not layout structure but it is very important data for RL. To mitigate this, the same data from more than one warehouse are collected.

The second risk is that if warehouse data is very large then there is a need the GPU / high end machine to train and deep reinforcement learning (DQN) may be used (Seno, 2017) instead of RL-Q learning which will take few days to train the model. For this, model training has to be started little earlier than schedule.

### Project Schedule

#### Timeline and scheduled activities

The Gantt chart will be used for tracking timelines, activities, tasks, and project risks (Meardon, 2016) and the project plan’s timelines and activities are as follows.

### Timeline and scheduled activities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Duration | Start Date | End Date | Days to Complete |
| Requirement Analysis | 8 Weeks |  |  |  |
| Project Title Selection |  | 1/7/2021 | 1/22/2021 | 15 |
| Requirement Analysis |  | 1/23/2021 | 2/14/2021 | 22 |
| Proposal |  | 2/15/2021 | 3/3/2021 | 16 |
| Final Requirements Analysis Submission |  |  | 7/7/2021 |  |
| Model Pipeline Design | 8 Weeks |  |  |  |
| Architecture / Design |  | 3/4/2021 | 3/15/2021 | 11 |
| Data Preparation / Collection |  | 3/16/2021 | 4/1/2021 | 16 |
| Data Preprocessing / Cleaning |  | 4/2/2021 | 4/28/2021 | 26 |
| Final Model Pipeline Design Submission |  |  | 7/14/2021 |  |
| Implementation | 8 Weeks |  |  |  |
| Training Model |  | 4/29/2021 | 5/9/2021 | 10 |
| Testing Model |  | 5/11/2021 | 5/20/2021 | 9 |
| Validating Model |  | 5/21/2021 | 6/1/2021 | 11 |
| Refining Model & Validate |  | 6/2/2021 | 6/12/2021 | 10 |
| Deploying Model |  | 6/13/2021 | 6/23/2021 | 10 |
| Final Implementation Submission |  |  | 8/4/2021 |  |
| Performance Analysis and Presentation | 8 Weeks |  |  |  |
| Monitoring |  | 6/24/2021 | 7/4/2021 | 10 |
| Collecting Metrics / Logs |  | 7/5/2021 | 7/17/2021 | 12 |
| Analyzing Performance |  | 7/18/2021 | 8/4/2021 | 17 |
| Presentation & Final Project Submission |  | 8/5/2021 | 8/18/2021 | 13 |
| Risk Management |  | 15% allocation |  |  |

#### Gantt chart

Gantt Chart

Chart, waterfall chart

Description automatically generated

It has four major milestones such as proposal, model pipeline design, implementation/build and conclude/performance analysis & presentation.

### Project Cost Estimate (or Alternative Criteria)

There is no cost involved for this project and the free cloud environment such as Heroku, google cloud provider, AWS and azure are used to deploy the model and present it from there.

### Project Issue Log

Any issue relevant to project will be tracked in the following chart.

Issue Log

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Name** | | Optimizing picking route for improving warehouse efficiency | | | |  |  |
| **Project Description** | | Optimizing picking route for improving warehouse efficiency using reinforcement q learning | | | |  |  |
| **Project Manager** | | Dr. Michele Bennett | | | |  |  |
| ID | Status | Priority | Issue Description | Assigned To | Expected Revolution Date | Escalate Required | Impact Summary |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

### Requirements Analysis: Use Cases

The use case helps to see the system functions from a user perspective and use case diagram emphasizes on what system does rather than how. It is used during the requirement analysis for capturing user requirements.

Use Case

Diagram

Description automatically generated with medium confidence

#### Use Case: Shortest optimal path

The robots / workers get the shortest optimal path use case where the system is trained with reinforcement learning – Q algorithms.

### Requirements Analysis: System Design

This project uses the top-down or water fall approach. All the tasks will be performed sequentially. First task starts from business problem or questions then finally it ends with finished workable or usable product. The project is using the overall system design through top-down / flow chart diagram.

Flow Chart

Diagram

Description automatically generated

### Requirements Analysis: Technical Requirements

To accomplish this project, we need the following instrumentations or technical components.

* Operating Systems: UNIX or Windows with RAM 32GB
* Environment: Cloud or On-Premises
* Language: Python (version 3:8)
* For visualizing the warehouse layout / structure: CAD floor software
* GitHub for storing codes and maintaining versions.
* Google cloud provider (GCP) is used for cloud environment.

### Requirements Analysis: Data Science Model Design

The following diagram explains the data model design.

Model Pipeline Design

Timeline

Description automatically generated

It has seven major modules such as data source, data analysis, training or implementation, testing, deployment, monitoring and reporting. First two modules will take more time than others especially more time will be spent on data analysis because data exploration, preprocessing, and PCA are especially important tasks in data science project. Once data is tradeoff properly with bias and variance, then training the model will be accurate.

### Requirements Analysis: Reports

This capstone project has few reports such as Q-values, and rewards matrix. The power BI / Tableau can be used for showing the results in graphically. The production model performance data / logs will be collected and represented it into proper format for the appropriate stakeholders.

#### Q-Values

It is initialized with zero at the beginning and once model started training, it keeps updating the q-values and final report is shown as below.

Q-Value ReportTable

Description automatically generated

#### Rewards Matrix

The walls and rack are called terminal states which has high negative rewards as -50 and aisle has -5 and packing area / goal which has positive reward 50. These details are shown below.

Rewards Matrix Report

Table

Description automatically generated with medium confidence

Storage or packing area has reward 50. The wall / rack has reward -50. The aisle or walkway has reward -5 which is aisle or walkway where workers or robots walks to pick the order. If the agent hits wall or rack, then it gets high punishment as -50 reward. The agent is always trying to maximize the cumulative rewards to reach the goal.

### Requirements Analysis: Screen Definitions and Layouts

This project is based on data science machine learning model, so it will not have much user interaction pages but it has one screen where user enters the picking location and submit the request. The request will be processed through trained model and return the shortest path. Screen will pass the data into API and API will call model to predict the result and the prediction output will be redirected back to the same screen (T, 2021).

User Interface

Graphical user interface, application

Description automatically generated

### Requirements Analysis: Security

There is no user interaction as well as it is not going to connect internet, so it won’t be any security issues, but it follows all the data security policies such as encryption, decryption, data ethics, standards, compliance and regulations. The cloud UI / URL will be SSL enabled for secure in transit.

### Detailed Model Pipeline Design

The pipeline for data as it flows through machine learning solution. Each step of the pipeline is fed data processed from its previous step. ML pipelines are cyclical and iterative as every step is repeated to until getting a successful algorithm (Koen, 2020).

The overview of the modules of detailed model pipeline design is shown in the following diagram.

Pipeline Design Diagram

Timeline

Description automatically generated

It has seven major modules/stages such as data source, data analysis, development/model build, testing, deployment, monitoring and reporting. First two modules will take more time than others especially data analysis may take more time than others because data exploration, preprocessing, and PCA are especially important tasks in data science project (Chaudhary, 2020). If data exploration is done properly, then rest of the modules will be smooth and model will be accurate.

Once data is tradeoff properly with bias and variance, then training the model will be accurate.

#### Overview

This capstone project has first stage as data source from where it gets the data for model build. The csv file used as input which has the RL’s state information. Once data is loaded from source stage then data analysis does data cleaning, PCA, and preprocessing then the model will be trained using RL algorithm. Once trained, the validated and promoted into another environment using pickle python package. Finally, the model is monitored in production and reports will be generated.

The model pipeline design should be an end-to-end scalable machine learning pipeline and the good ML design consists of data driven approach, less latency, loosely coupled, scalable, fault-tolerant, maintainability, support of batch and real-time processing (Keon, 2019).

#### Data Sources

This is a data ingestion service where the raw data is collected in offline as well as online. The data will be download from client’s environment/warehouse via secure ftp (SFTP) or internet (HTTP). If they are in cloud, then it will be downloaded from their storage buckets (Azure/GCP) and S3(AWS). The data will be in raw, and it is immutable/not changeable. The project uses a few collection tools or methods.

##### Data Collection Methods

The data science capstone thesis or project is driven by data and it is a core component of the research. The project needs the existing authentic data so interview, questioning, observations and secondary or constructed approaches are used for data collection (Creswell,2017).

##### Data Quality

The data quality is especially important one in data science otherwise model may be bias and variance. If the quality of data needs the improvement, then data must be relevancy, value-added, believability, accessibility, reputation of the data, accuracy, and timeliness (Hazen, 2014). For maintaining the quality of data, it needs to be collected the data consistently, and completeness(Vaughan, 2019).

#### Dataset Types and Formatting

Data may be unstructured, semi-structed data, and structured and it may not have the proper data for features. RL’s state data is integer data type and actions are also integer data type. During the data analysis phase, the raw data will be transformed or converted into useful information that will later be used for decision-making and model training. Data sources are merged and filtered. Finally, it has been into the proper layout and data type. This project uses the warehouse data, which is mostly transactional data, so it is structured data. This project uses reinforcement Q-learning (RL) algorithm so warehouse location layout data will be used for RL states and source is already providing the RL state data in csv file with integer data type values so there is no need of data type conversion required.

#### Data Cleaning Procedures

The data is the most valuable thing for machine learning. The data cleaning is a process of identifying the incomplete, incorrect, inaccurate, irrelevant, or missing part of the data and then modifying, replacing, or deleting and discard them if they are unnecessary or aberrant data that is not useful for later pipeline steps / stages (Bonner, 2019).

The project uses python language and reinforcement Q-learning algorithm. Python has numpy and pandas which will be used for data cleaning. The inconsistent columns can be dropped by dataframe.drop method. For missing data, we can leave it as or fill the values, or drop them. Python has dropna, isnull, fillna methods for handling missing values. This project uses these python methods for data cleaning process.

#### Data Exploration

Project uses some tools such as graphs, univariant analysis, bivariate, multivariate, and principal component analysis (PCA) for correlation to understand the raw. It also checks the unwanted data such as noise, anomaly, trends, incorrect, missing, skewed data null and empty values in data and it removes them if not needed (Kibish, 2018). The feature extraction task is also performed as well as data separation such as training dataset, testing dataset and validation dataset. The boxplot is used to find the any outliers / anomaly (Aissa,2016). This is reinforcement learning so it makes sure that the proper valid data is available for agent, states, environment, and rewards.

#### Data Model

This capstone project uses reinforcement Q-learning (RL) algorithm. Q-learning is a model free algorithm which estimates the optimal policy without using transition and reward functions of the environment and it is an off-policy learner (Violante, 2019).

For building the model, first, an environment, agent, states, q-values, q-function and rewards are defined then build the model.

The model’s algorithms use Q-Function for learning which uses Bellman equation and its details are as follows

Diagram

Description automatically generated

Q-function takes two inputs such as state and action and it has a few parameters such as learning rate, gamma / discount factor and epsilon.

*Gamma*

It is used to balance the immediate and future reward and typically this value will be a range between 0.1 and 0.9.

*Learning Rate*

It also referred as alpha, and it defines how much system can accept the new value vs the old value. Value between new and old and then multiplying that value by learning rate. This value will be added to previous q-value which moves it in the direction of our latest update.

The Q-learning algorithm process flow is shown below.

Q-learning algorithm process

Diagram

Description automatically generated

##### Initialize Q-Table

It has n number of columns and m number of rows.

Where n = number of actions. It has up, down, right and left

m=number of states and here warehouse layout location is used

Initially Q-table has zeroes for q-values and there is an iterative process for updating q-values.

*Choose action and perform action*

Once start exploring the environment, q-function provides the better approximations continuously updating q-value in the table (Shyalika, 2019). Then, update the Q-values for being at the start and moving right using the Bellman equation which is stated above. At beginning epsilon greedy rate is higher, agent explores environment and choose more random actions because agent does not know anything about environment. Once agent explores the environment, epsilon rate decreases, and an agent exploits the environment as well as it gets more confident in estimating the Q-values. Here environment is warehouse and agent is forklift or robot or system.

*Measure Rewards*

Now an action is taken and observed an outcome and reward and the rewards matrix with multiple episodes are run /trained until reaching the accuracy (Bhatt, 2019).

*Update Q-table*

Here function Q(s,a) is updated and this process is repeated until learning is stopped. In this way, Q-table is updated, and function Q is maximized.

##### Testing and Validation

For performance indicators, the statistical and domain knowledge’s are considered and the key performance indicators (KPI) details are collected from domain experts and test the models and compare the results against KPIs and verify whether prediction is acceptable or not

This project is based on reinforcement learning algorithms, so there several evaluation / validation measures are done, and they are such as agent performance, cumulative rewards, steps and total time taken for shortest optimal path. Due to reinforcement learning algorithm, there is a need for testing the agent’s quality whether it is trained well or not.

Once trained the model, the model must be tested with testing sample. Then the result with actual is compared or theoretically best result as well as it should be proper tradeoff between bias and variance. The error rate and accuracy of the model should be within acceptable criteria.

Agent’s decision quality must be tested because agent plays a major role in this project / algorithm.

The performance of the reinforcement learning algorithms can be plotted for cumulative rewards vs number of the steps.

The performance graph between rewards vs steps

Chart, line chart

Description automatically generated

It runs the actual reinforcement learning algorithms against training data. First it builds model then it tests against testing dataset as well as validation dataset to get the appropriated accuracy.

The following details have to be tested / evaluated in reinforcement learning.

*The quality of the learned policy*

*How much rewards are obtained if the agent follows the policy*

*How much resources are used such as memory, CPU / GPU*

*Data / sample / experience needed to converge the certain level of performance*

*Hyper parameter tuning such as learning rate and discount factors*

#### Methodology

The methodology in model design pipeline is the best way to organize the pipeline components in a proper sequence. The detail model pipeline design methodology has few steps such as

#### Understand the use case / business problem

Figuring out what needs to be solved using data and machine learning.

#### Pose the problem

Objective that model wants to optimize.

#### Select the right dataset

How to deal with bias, skewed, labels

#### Feature engineering

Transform data into features.

#### Model selection

what model is best suited for business

#### Training

Model will be trained and validated.

#### Productionizing

#### The model will be deployed, monitored, and get feedback.

It is the routine for finding solutions to a specific business problem and this is a cyclic process for business understanding or strategy, analytic understanding, data requirements, data collection, data understanding, data preparation, modeling, evaluation, deployment, and feedback (Gajare, 2019). The methodology should convert the business problem to data science solution.

For project management, the agile methodology will be used because it is an interactive process as well as the trained model results will be validated with business immediately.

##### Deployment

Once model is tested thoroughly and the desired output is achieved then the trained model will be deployed into production. For deploying model into different environments, first it has to be converted it into binary format (pickle file). The binary file will be deployed and loaded back to trained model stage for future prediction.

##### Performance and monitoring

Once mode is in production, the performance data and logs will be collected and analyzed for further improvements as well as model will be monitored.

*Reports*

The reports are generated from production logs and performance data and reported to the corresponding stakeholders.

#### Configuration Changes OR Alternative Criteria

The configuration changes / management is a system engineering process for establishing consistency of a product’s attributes throughout its life and it is an IT management process that tracks individual configuration or change items of an IT system. It consists of version control, auditing, identification, continuous integration / continuous deployment (CI/CD), DevOps and baseline (Buchanan, 2018).

The project uses templates for system configuration and deployment, and they are a combination of configuration and execution scripts that define scope, type, and configuration changes.

The codes are stored in GitHub and all the changes and versions are maintained in github. The github url is https://github.com/kannannova. The cloud providers have their own versioning. This project is also deployed into Google cloud provider (GCP) – App Engine. Whenever any changes to code, it automatically creates a newer version. The single template is used for all the environments and DevOps is used for automating the deployment. CI/CD is used for automated deployment from version control to environment. The all the changes and logs are captured, and verification or peer review process are used (Broda,2019).

#### Security

This capstone project uses data heavily so data and network must be secured. The entire data pipeline processes must be secured (Zimmerman, 2019). The project has one user interface (UI) html page where user provides input for model to predict the output and secure socket layer (SSL) is enabled for URL so that url will be HTTPS instead of HTTP. For data, the encryption, decryption, digital signature, and lease privileged access control are used. If the model is deployed into cloud, so data is secure in transit as well in rest by default.

It also follows all the data security policies, data governance, data ethics, standards, compliance, regulations, proper security protocols and data security such as encryption, decryption, digital signature, authentication, and authorization.

#### Hardware and Software Technologies

To accomplish this project, the following instrumentations, or technical components.

* Operating Systems: UNIX or Windows with RAM 32GB
* Environment: Cloud or On-Premises
* For visualizing the warehouse layout / structure: CAD floor software
* GitHub for storing codes and maintaining versions.
* Cloud: Google Cloud Provider (GCP) – App Engine.
* Language: Python (version 3:8)

Python libraries and their details as follows.

dataframe-image==0.1.1

Flask==1.1.2

gunicorn==19.9.0

numpy>=1.9.2

imagesize==1.2.0

imgkit==1.2.2

Jinja2==2.11.2

joblib==1.0.1

json5==0.9.5

matplotlib>=1.4.3

pandas>=0.19

Werkzeug>=0.15

requests==2.2.1

cloudstorage==0.11.0

gcloud==0.18.3

### System Entities

The capstone project has the system entities such as logistics warehouse operations such as put away, picking, and a reinforcement Q-learning (RL) component. RL has environment, agent, states, actions, and rewards as system entities. The actions have four entities such as up, down, right, and left and these are converted into integer value. Rewards have the numeric value such as -50, -5 and 50. The algorithm uses bellmen equation and it has visual diagrams such as performance diagram, boxplot, and reports.

The source of the data is from Ryder Inc’s warehouse system and warehouse layout data is used for RL’s state information. It is a CSV file, and all the features or columns are integer data type.

### Functional Requirements

This section describes the functional requirements about this capstone project. The business requirement is that the model should improve the overall warehouse efficiency through optimizing the picking operation. The business requirement should be converted into functional requirements and mapped into technical requirements such as data source, data analysis, model build, testing, production, monitoring and reporting.

#### Functionality Mapping

There are few functional requirements for this capstone project such as data collection, data exploration, build model which provides the shortest optimal path for picking, user acceptance, production deployment, support, and reports. The functional requirements into technical requirements / data pipeline mapping is an important process in the information technology. The following table has the mapping details between functional, data pipeline stages and OSEMN framework stages (Lau, 2019).

Mapping Table

|  |  |  |
| --- | --- | --- |
| **Functional requirement** | **Data pipeline stages** | **OSEMN stages** |
| Data | stages 1 - Data source | Obtaining data |
| Data exploration | stages 2 - Data Analysis | Scrubbing / Cleaning data & exploring / Visualizing data will allow to find patterns and trends |
| Build shortest optimal model | stages 3 -Model Build | Modeling data will give and predictive power as a wizard |
| User Acceptance Test | stages 4 - Testing | Interpreting data. |
| Go-live | stages 5 – Production deployment and enabling for user | Interpreting data. |
| Production Support | stages 6 -Monitoring | Interpreting data. |
| Reports | stages 6- Reporting | Interpreting data. |

The following table has the mapping between functional requirements and code.

Mapping the functional requirements

|  |  |
| --- | --- |
| Requirement 1 - FR1 Creating and Loading Data | qValues[], rewards[], actions[] |
| Requirement 2 - FR2 Training the model | isTerminalOrWall() findStartLocation() findNextAction() getNextLocation() findShortestPath() episode = 5000 |
| Requirement 3 - FR3 Deployment | pickle.dump(wms, picklefile) |
| Requirement 4 - FR4 Prediction | wms.findShortestPath(6,2) |

### Source Code Listing

Source codes, their brief description of all the classes or pages and files are explained below.

#### Index.html

This serves as a user interface and it has html controls such as textbox, labels, and button.

the UI is shown below.

Index.html Page

Graphical user interface, text, application, chat or text message, email

Description automatically generated

Google cloud provider (GCP) – index.html

Graphical user interface, text, application, chat or text message

Description automatically generated

The GCP cloud public url is [ML API (numeric-span-317003.ue.r.appspot.com)](https://numeric-span-317003.ue.r.appspot.com/) or paste this <https://numeric-span-317003.ue.r.appspot.com/> url in web browser

Html code

Text

Description automatically generated

#### app.py

This is a main entry for program which has python flask library for providing the webframework and webserver for running index page and handling request and render\_template pages. It creates flask app and loads rlwms.pkl file which was trained in wms.py program. Finally, it renders index.html at home function.

Once user enters the data and clicks ‘Show Shortest Path’ button, it shows path details in textbox as well as in graph. The pseudo code is given below.

#### app.py code

Text

Description automatically generated

#### wms.py

Here the actual reinforcement q-learning is trained, tested, and converted it into pickle file for production deployment.

wms.py

Text

Description automatically generated

Text

Description automatically generated

#### requirements.txt

This file has all the python packages which are needed to install before deploying the model. This capstone project needs the following packages details in requirements.txt file.

dataframe-image==0.1.1

Flask==1.1.2

gunicorn==19.9.0

numpy>=1.9.2

imagesize==1.2.0

imgkit==1.2.2

Jinja2==2.11.2

joblib==1.0.1

json5==0.9.5

matplotlib>=1.4.3

pandas>=0.19

#### app.yaml

This model will be hosted into google cloud provider (GCP). For this, app.py will be renamed into main.py and new file app.yaml file is also created. App.yaml has the following content.

runtime: python38

#### Python Libraries

##### *NumPy*

It is an open-source python package, and it is fast and versatile, indexing and broadcasting are standards of array computing. It supports multidimensional array.

##### *Pandas*

It is an open-source python package, and it allows importing data from various file formats. It is also used for data manipulation operations.

##### *Flask*

It is a python API which allows to build the web applications.

##### *Request*

It is a http requests in Python.

##### Jsonify

It is a function in Flash’s flash json module.

##### render\_template

It is used to generate output from a template file based on the Jinja2 engine that is found in the application's templates folder.

##### pickle

It is to serialize and deserialize objects in Python.

### Code Review

The goal of code reviews is to ensure that code works well without any bugs or error, and it also helps to minimize the future problems with codebase. The code review can be conducted during development as well as after development (Dhanani, 2018). This project has the python code so better to check whether code is written with best practices and patterns as well as whether it handles error handling properly or not. Each function or code is gone through the unit test prior to QA / UAT testing.

Once deployed the model into production, the live model performance will be monitored and collected the logs. The logs will be reviewed which is also one kind of code review which helps to find out how code works in production. If any bugs or performance issues will be fixed in the next deployment.

### Implementation

The implementation plan has the detail plan for implementing or executing the project which includes the description of the procedures, practices, software libraries, databases, programs, hardware components, and activities which are needed to carry out the project into production successfully.

Once model is trained then next step is to test the it whether it performances well against test and validation dataset. Once all the testing are over and met the required accuracy then trained model will be promoted into production. The following section describes the implementation plan and its components such as data pipeline architect, software, software, hardware, and deployment procedure.

#### Data Pipeline Architect

The pipeline or pipe has in and out and the data enters pipeline, and the insights or predictions exit from pipeline. The data pipeline is used to automate the machine learning / data science workflow from beginning to end.

The machine learning workflows can be split into independent, modules/micro services, and reusable then they can be pipelined together to create the models and orchestrate them in a controlled way. Thus, more efficient, simplified models and cutting out the redundant work can be achieved as well as easy to debug because it is modularized. This capstone project is used to find the shortest optimal path for picking process in warehouse using reinforcement Q-learning algorithms and its pipeline design is shown below.

Pipeline Design

Timeline

Description automatically generated

#### Source of Data

The data will be download from client’s environment/warehouse via secure ftp (SFTP) or internet (HTTPS) or database. If they are in cloud, then it will be downloaded from their storage buckets (Azure/GCP) and S3(AWS). The data will be in raw, and it is immutable/not changeable.

#### Data Exploration and Visualization

Data exploration is the first step in data analysis stage used to explore and visualize data to open the hidden insights from data. It is used to understand the bigger picture and get to insights faster. Here checking the data distribution, finding correlation, finding the outliers, and missing values tasks are performed.

##### Data Model

The capstone is that will improve the warehouse operations using machine learning through reinforcement learning. The model’s algorithms is Q-Function for learning which uses Bellman equation and its details are as followsDiagram

Description automatically generated

For building the mode, first environment, agents, states, and rewards are defined then build the mode. Initially Q table has zeroes and there is an iterative process for updating the values. Once start exploring the environment, q-function provides the better approximations continuously updating q-value in the table (Shyalika, 2019).

#### Development / Model Build

The model uses reinforcement Q-learning algorithms for finding the shortest optimal path, so model creates environment, agent, states, rewards as well as updates the q-table and rewards matrix with multiple episodes until reaching the accuracy (Bhatt, 2019).

#### Testing

This project is based on reinforcement learning algorithms, so there several evaluation / validation measures are done, and they are such as agent performance, cumulative rewards, steps and total time taken for shortest optimal path. Due to reinforcement learning algorithm, there is a need for testing the agent’s quality whether it is trained well or not.

Once trained the model, the model must be tested with testing sample. Then the result with actual is compared or theoretically best result as well as it should be proper tradeoff between bias and variance. The error rate and accuracy of the model should be within acceptable criteria.

Agent’s decision quality must be tested because agent plays a major role in this project / algorithm.

The performance of the reinforcement learning algorithms can be plotted for cumulative rewards vs number of the steps.

The performance graph between rewards vs steps

Chart, line chart

Description automatically generated

The following details must be tested / evaluated in reinforcement learning.

*The quality of the learned policy*

*How much rewards are obtained if the agent follows the policy*

*How much resources are used such as memory, CPU / GPU*

*Data / sample / experience needed to converge the certain level of performance*

*Hyper parameter tuning such as learning rate and discount factors*

##### Complete Test Plan

*Test Plan*

It involves explicit checks for behaviors that model to follow. In machine learning, there are two types of tests such as pre-train tests and post-train tests. It will have a plan for testing the overall business functionalities and it may have one or more test cases.

*Pre-train tests*

This is performed early and catches the bugs before running the model and it does not train model. It checks whether any labels missing or not.

*Post-train tests*

It is performed on a trained model and checks whether it performs correctly. Here the invariance tests, directional expectation tests and minimum functionality tests are performed.

##### Test Cases

*Model / User Interface Validation – Invalid Picking Location*

If user enters invalid picking location, terminal / wall then model returns or says ‘Picking Location ['27', '17'] is invalid’. here 27,17 is invalid pick location.

Graphical user interface, text, application, chat or text message

Description automatically generated

*Model / User Interface Validation – Valid Picking Location*

If user enters invalid picking location such as example 25.6 then model returns the result, and it shows it in text box as well as in picture / graph.

Table

Description automatically generated

If user clicks ‘Show Shortest Path’ button without input values, then it shows ‘please fill out this field’

Graphical user interface, text, application

Description automatically generated

#### Deployment and Performance Monitoring

For deployment environment, Heroku or google could be used and the api interface will be created and it will be acting as user interface where users can enter the input for prediction.

Here the trained model will be converted into binary (pickle) and deployed into another environment. The pickle file will be loaded back to trained model and prediction will happen from trained model. The model will be hosted in google cloud provider (GCP).

#### Reports

This capstone project has few reports such as Q-values, and rewards matrix. The power BI / Tableau can be used for showing the results in graphically.

##### Q-Values

It is initialized with zero at the beginning and once model started training, it keeps updating the q-values and final report is shown as below.

Table

Description automatically generated

##### Rewards Matrix

The walls and rack are called terminal states which has high negative rewards as -50 and aisle has -5 and packing area / goal which has positive reward 50. These details are shown below.

Table

Description automatically generated with medium confidence

Calendar

Description automatically generated

The green square is storage or packing area where from the items are shipped, and it has reward 50. The light black is wall, and it has reward as -50. The light blue is rack which also has reward -50. The white square is -5 which is aisle or walkway where workers or robots walks to pick the order. If the agent hits wall or rack, then it gets high punishment as -50 reward. The agent is always trying to maximize the cumulative rewards to reach the goal.

##### Performance Plot

The performance graph / plot between cumulative rewards vs steps.

Graphical user interface

Description automatically generated with medium confidence

Algorithm Performance Plot

Chart, line chart

Description automatically generated

Once episodes are increased, the rewards are also increased which means that agent is learned well. At the beginning, agent exploring then later, it does exploitation.

### Security

The security consideration is taken care across all the stages of data pipeline. The network and data are secured on transit as well in rest. The user interface will have SSL for https URL / network and code and data are in cloud platform which means there are secure in transit as well as at rest. For https, certificates will be used which encrypts network so no one can decrypt without private key / certificates. For data, encryption and digital signatures will be used. If there is regulations or compliance, then it will be followed and only the authorized person will be allowed to access the data.

#### Governance

The model is not dealing with any compliance data, so it won’t be any compliance or regulation or policy or rules as well as no data governance.

#### Risk

There are a few risks for this project and first major risk is that getting the warehouse layout data which is needed to create states in environment for reinforcement model. Normally warehouse keeps the transactional data such as locations, products/SKUs but not layout structure but it is especially important data for RL. To mitigate this, the same data from more than one warehouse but if there is no data then data is created or simulated based on CAD floor layout drawing based on warehouse structure.

The second risk is that if warehouse data is exceptionally large then the GPU / high end machine is needed to train, and the model will be Deep RL (DQN) instead of RL Q-learning which will take few days to train the model.

#### Quality Control

The data quality will be tested at preprocessing stage as well as it will be tested after trained the model.

##### Data Quality

The data quality is especially important one in data science otherwise model may be bias and variance. If the quality of data needs the improvement, then data must be relevancy, value-added, believability, accessibility, reputation of the data, accuracy, and timeliness (Hazen, 2014). For maintaining the quality of data, it needs to be collected the data consistently, and completeness.

### Completion OR Application Functionality and Execution

#### Application Functionality

This capstone project has an UI which is created by index.html page. It has textbox control and submit button. Once user enters the pick location in textbox and click the submit then the values are passed into trained model and gets the back the shortest optimal path details. The return value is shown in two format such as text and graph.

The following section describes the application execution details. The main project functionality is written in wms.py python file and its details are as follows

#### Code - wms.py

##### Libraries

The below code block has the all the necessary libraries for this project

Graphical user interface, text, application

Description automatically generated

##### Read Data From File

A picture containing calendar

Description automatically generated

Data is loaded from csv file, and it is used for creating reinforcement Q-learning states.

##### Data size

RL state has 30 rows and 30 columns.

Graphical user interface, text, application

Description automatically generated

##### Data Cleaning

Graphical user interface

Description automatically generated

All the missing and null values are checked and corrected.

##### Data Exploration

Graphical user interface, application, table

Description automatically generated

*Correlation*

It is not supervised or unsupervised learning algorithm so there is no need to check the correlation covariance between variables which means that there is no dependent variables so there is no need of checking correlation.

*Outliers or anomaly detection*

The RL states are captured from warehouse floor layout, so model has to check whether the any anomaly or outliers are present or not. It can be done it by box plot.

Box Plot

Chart, bar chart

Description automatically generated

There is no outlier. Column / variable 14 is packing, or final location and it has positive rewards as +50 that is why it shows as an outlier, but it is not really an outlier. There are no outliers or anomaly in the data.

##### Data Preprocessing

Graphical user interface, text, application, email

Description automatically generated

RL needs the environment, states, actions, and rewards and these data are created in preprocessing stage.

##### Build Model

**Text

Description automatically generated**

The model has few helper functions such as findNextAction, findNextLocation, findStartLocation, isTerminalorWall and predict function as ‘findShortestPath’

##### Training the model

The model is trained 5000 times / episodes with discount factor = 0.9, learning rate =0.9 and epsilon = 0.8

***Text, application

Description automatically generated***

##### Evaluation and Validation

Once model is trained then the validation will be done. It is not supervised / unsupervised learning so there is no need for data split for model and test. The few picking locations will be fed into trained model and output will be returned which will be shown in text control and graph as below.

Graphical user interface

Description automatically generated

#### Deployment Packages, Hardware and Software Technologies

The packages will have the following files.

#### UI - Index.html which is user interface UI where user enters the input for model to predict.

App.py and main.py for google cloud and it is an entry point / method.

App.yaml for google cloud deployment. It has entry as ‘runtime: python38’ which says that google cloud has to use python 3.8 environment for this project.

Rlwms.pkl which will have the trained model in binary format

#### Requirements.txt

It can be created using pip freeze python command and these packages must be installed prior to deploy the model.

python command for creating requirements.txt

Text

Description automatically generated

Requirements.txt file’s content is shown below

dataframe-image==0.1.1

Flask==1.1.2

gunicorn==19.9.0

numpy>=1.9.2

imagesize==1.2.0

imgkit==1.2.2

Jinja2==2.11.2

joblib==1.0.1

json5==0.9.5

matplotlib>=1.4.3

pandas>=0.19

Werkzeug==0.15.1

requests==2.24.0

#### Packing for deployment

#### Creating pickle file

Graphical user interface, text, application

Description automatically generated

The pickle library is used for creating binary object from trained model

##### Validating the deployment package

It loads rlwms.pkl pickle file into memory and calls findShortestPath method with real time inputs. For example, the following output is returned from model for input picking location (25, 17)

Graphical user interface, text, application

Description automatically generated

A picture containing table

Description automatically generated

#### Validating Index.html

##### Input

Graphical user interface, text, application

Description automatically generated

##### Output

The following output is for picking location (19,5)

Table

Description automatically generated

#### Hardware and Software Technologies

To accomplish this project, the following instrumentations, or technical components.

* Operating Systems: UNIX or Windows with RAM 32GB
* Environment: Cloud or On-Premises
* Language: Python (version 3:8)
* For visualizing the warehouse layout / structure: CAD floor software
* GitHub for storing codes and maintaining versions.
* Cloud: Google Cloud Provider (GCP)

#### Python Code File Details

The following files are attached for python code for this assignment

1. index.html.txt
2. wmsdata.csv.txt for data
3. app.py.txt
4. wms.py.txt
5. wms.ipynb.txt
6. wms.pdf
7. rlwms.pkl.txt
8. requirements.txt

please remove the .txt from files prior to run the code

### Performance Analysis

Once model is built or trained, the model must be tested to check whether it meets the requirements or not.

#### Test Plan

It involves explicit checks for behaviors that we expect our model to follow. In machine learning, we commonly we do the two types of tests such as pre-train tests and post-train tests. It will have a plan for testing the overall business functionalities and it may have one or more test cases.

##### Pre-train tests

This is performed early and catches the bugs before running the model and it does not train model. It checks whether any labels missing or not (Jordan, 2020).

##### Post-train tests

It is performed on a trained model and checks whether it performs correctly. Here we do the invariance tests, directional expectation tests and minimum functionality tests (Jordan, 2020).

#### Component Testing (Module Test Cases)

Here the individual module or component is tested. The following table format is used for testing and tracking the test results. There are three module test cases such as front-end user interface (UI) invalid location, front-end UI valid location, and model output / result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Name: Model results / output | | | | |
| Priority: High | | | | |
| Module: Reinforcement Q-Learning Algorithm | | | | |
| Test Objective: To make sure that model produces the desired results –shortest optimal path | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
| Step |  | Test Details | Expected Results | Problem/Issue |
| 1 |  | Pass picking location such as 13,27 | System allows to enter values | No issues are reported |
| 2 |  | Call the predict method with step 1 inputs | System allows to call API | No issues are reported |
| 3 |  | Display Results | The module provides the shortest optimal path in text and graph | No issues are reported |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Name: User Interface Validation – Invalid Picking Location | | | | |
| Priority: Medium | | | | |
| Module: index.html | | | | |
| Test Objective: The module accepts only the valid input. If input is not correct, it shows error message. | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
| Step |  | Test Details | Expected Results | Problem/Issue |
| 1 |  | No input and click the submit button | Shows error message ‘Please enter the picking location’ | No issues are reported |
| 2 |  | Enter invalid picking location 27,27 | Shows error message ‘Invalid Picking location’ | No issues are reported |

If user enters invalid picking location, terminal / wall then model returns or says ‘Picking Location ['27', '17'] is invalid’. here 27,17 is invalid pick location.

Invalid Location

Graphical user interface, text, application, chat or text message

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Name: User Interface Validation – Valid Picking Location | | | | |
| Priority: Medium | | | | |
| Module: index.html | | | | |
| Test Objective: The module accepts only the valid input and it shows results | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
| Step |  | Test Details | Expected Results | Problem/Issue |
| 1 |  | Enter valid location 26, 6 | The module provides the shortest optimal path in text and graph | No issues are reported |

If user enters the valid picking location such as example 25,6 then model returns the result, and it shows it in text box as well as in picture / graph.

Valid Location

Table

Description automatically generated

***Requirements Testing***

Once module tests are over or passed, the requirements testing is performed against requirements to make sure that it meets the requirements gathered in the requirement phase of development. All the requirements are mapped into individual test cases, tests are performed against all the test cases for both functional and non-functional test cases including performance, reliability and usability. The verification of expected and actual result is done if the result is not as actual then bug ticket is created with the results. The following test checks the requirement of performance and scaling for a large data.

|  |
| --- |
| Component: Ability to find the shortest optimal path for picking operation in warehouse |
| Name of Developer: Kannan Nova |
| Name of Reviewer: Kannan Nova |
| |  |  |  | | --- | --- | --- | | Checklist | | | | Type | Pass | Comments | | functionality | Passed | Met all the requirements. | | Performance | Passed | It handles a large amount of data and there is no scaling issue | |

The requirement testing is validated against use cases and this project has the following use case

The use case helps to see the system functions from a user perspective and use case diagram emphasizes on what system does rather than how. It is used during the requirement analysis for capturing user requirements.

Use Case

Diagram

Description automatically generated with medium confidence

#### Use Case: Shortest optimal path

The robots / workers get the shortest optimal path use case where the system is trained with reinforcement learning – Q algorithms (Violante, 2019).

The following test checks the requirement of performance and scaling for a large data.

|  |
| --- |
| Component: Ability to find the shortest optimal path for picking operation in warehouse |
| Name of Developer: Kannan Nova |
| Name of Reviewer: Kannan Nova |
| |  |  |  | | --- | --- | --- | | Checklist | | | | Type | Pass | Comments | | functionality | Passed | Met all the requirements. | | Requirement: show the shortest optimal path | Passed | It shows the shortest optimal path for picking operation | |

#### System Testing

The system testing will be performed to verify that all the components of functional business requirements, business processes, data flows, integration, and other system criteria are met. It is a testing level to verify if a complete build aligns with functional and technical requirements made for it including sanity, usability, performance, stress and load testing (Dawood, 2020).

In system testing the following testing are conducted

##### Functional test

Once model is trained and deployed into QA environment, QA team tests whether system is proving the optimal shortest path details for picking operation.

If user enters picking location such as example 25,6 then model returns the result, and it shows it in text box as well as in picture / graph.

Functional test result

Table

Description automatically generated

The integration test is to make sure that all the components are working together as well as regression test helps to find whether the new module or functionality is not breaking the any existing functionalities (Dawood, 2020).

Once all the components including index page and trained model / pickle file deployed together into webserver, QA tests whether system works together and provides the results or not.

##### Operation and Maintenance

Once model is in production, the performance data, feedback, and logs will be collected. They are verified and analyzed for further improvements. The reports are generated and presented. The operation and support take care of day-to-day operations and the production support team or maintenance team does maintenance activities such as

* Monitoring the systems, and servers
* Checking the systems health
* Watching the security issues
* Backing up the data and servers regularly

### User Guide

The proper user guide will be created, and which will have general information, system summary, getting started and troubleshooting, help, contact, production support details, and FAQ.

The general information and system summary of this project is that it is used to find the shortest optimal path through machine learning – reinforcement q-learning then the result will be given to warehouse workers or robots for picking the goods from warehouse storage location. For system troubleshooting, all the logs will be used. User guide will also have glossary of common machine learning, supply chain logistics, warehouse, statistics, and data science terms.

The user guide or troubleshoot instruction is created.

*Step 1*

Browse index.html. This project is hosted in location and url <http://127.0.0.1:5000> if it is a cloud environment,

UI / Index.html

Graphical user interface, text, application

Description automatically generated

For cloud,

Graphical user interface, text, application, email

Description automatically generated

*Step 2*

Enter the valid picking location and click submit / ‘show shortest path’ button

Prediction Result

*Table

Description automatically generated*

The daily backup, weekly, and monthly backup are taken properly and if the system is corrupted, the backup will be stored.

#### Troubleshooting

If the user interface is not displayed in browser, then server might be down so check whether server is running or not.

If the user gets the error in browser, then model is corrupted, or the python packages / libraries are missing and reach out to support team to fix the issue.

#### Frequently Asked Questions (FAQ)

Is any client software installed at the user end?

No, this project uses web-based UI and web API so user can use web browser and get the predictions results on browser.

Is data secure?

Yes, the data is secure in transit and at rest.

#### Help and Contact Details

If there is any issues or model is not working, then please reach out to help desk for support as well create a ticket or incident so that IT or business support team will be notified.

#### Glossary

**A**

Agent

RL agent who learns the environment with actions and locations and provide the shortest optimal path

Action

It is a component in reinforcement learning.

**L**

Location

It is a warehouse layout details which has bin with rack and aisle combination

**P**

Picking

Where take out the goods from final location and stuff into truck for shipping (Lopienski, 2020).

Put away

Where download the goods from truck and place it into final location inside warehouse (McCrea, 2016).

**R**

Reinforcement Learning

The reinforcement learning is one of the machine learning algorithms. When environment is stochastic or dynamic or there is no training data or specific enough expertise about the problem then RL is useful

Reward

When Agent learns the environment, it gets rewards and tries to accumulate the maximum rewards. Rewards can be positive or negative.

**S**

**States**

It is a RL component. Agent takes an appropriate action and gets the rewards for that.

**W**

Warehouse

The warehouse is a part of supply chain logistics industry where store the goods and bring back it from storage for shipment.

### System Administration Guide

It is a like user guide, but this guide will help to troubleshoot the system with admin rights.

*Step 1*

First check whether the server is running or not

Server Status

*Text

Description automatically generated*

*Step 2*

Ping or browse <http://127.0.0.1:5000/> or [WMS RL API](http://127.0.0.1:5000/) the server whether it is reachable or not. If not, then restart the server

Index Page

*Graphical user interface, text, application

Description automatically generated*

#### System Overview

The system overview is that it is developed by reinforcement Q-learning algorithm which uses Bellmen equations, and it has few components such as environment, agent, states, actions and rewards (Shyalika, 2019). The warehouse location or layout data is used for RL’s states and once model is trained with data; it is used for finding the shortest optimal path for picking process in warehouse. The system is accessed from browser and model is exposed as web API and index or home accepts input and passes it into API. when API returns output which will be shown in index.html page and the model is trained with multiple episodes.

#### System Configuration

The capstone project has the following system configurations

* Operating Systems: UNIX or Windows with RAM 32GB
* Environment: Cloud or On-Premises
* Language: Python (version 3:8)
* For visualizing the warehouse layout / structure: CAD floor software
* GitHub for storing codes and maintaining versions.
* Google cloud provider (GCP) is used for cloud environment.

#### System Maintenance

The system is monitored and maintained by operational support team. Every Sunday morning between 12am and 3am, system will not be available to users and public due to maintenance. During that time, logs and temp files are moved into archived, storage volume is verified, and size of the space may be increased if needed. The full back is taken after bringing all the systems including server, database, and storage. This project needs windows 64-bit operating system, file server or cloud storage bucket for storing the data and images and these resources will be backed up.

#### Security Related Processes

The hosted URL is always secured by secure socket layer (SSL) with https. The archived and back up data is always encrypted, and only authorized person can decrypt the data (Zimmerman, 2019). This project follows all the rules, regulations, compliances, and data ethics (McRaw, 2020). During the maintenance, all the security certifications are verified and if expired then it will be renewed manually by operational support team.

#### CopyRight

Copyright © 2021 Kannan Nova.

All rights reserved. This project or any portion thereof may not be reproduced or used in any manner whatsoever without the express written permission except for the use of brief quotations in a review.

#### Appendices

The appendices can help readers better understand the materials from the supplementary documents or sections.

##### Appendix A

Warehouse – logistics and Supply Chain Management

It is an integral part of logistics and supply chain management system, and it has three core operations such as inbound /put away, outbound / picking and inventory management. Inbound is receiving goods from container or truck and stores them into storage. Outbound brings the goods back from storage and ships them for consumer or customer (Ullrich, 2019).

##### Appendix B

Reinforcement Q-Learning

The reinforcement learning is one of the machine learning algorithms. When environment is stochastic or dynamic or there is no training data or specific enough expertise about the problem then RL is useful (Kiili, 2019). The Q-learning is at the heart of all reinforcement learning and it is an off policy that seeks to find the best action to take given the current state. It has few components such as environment, agent, actions, locations, and rewards (Violante, 2019).

#### Table of Figures

Table of Figures

|  |  |
| --- | --- |
| Name | Description |
| UI / Index.html | This is home page figure or image where user enters the input values for prediction and submit |
| Prediction Result | The picture shows the prediction results in image / visual which will help to understand the whole picking path / route. |
| Invalid Location | If user enters invalid location, UI shows the invalid location |
| Valid Location | The figure / image provides the shortest optimal path in text and graph |
| Server Status | This image shows whether server is running or not. |

### Digital Poster

The original poster file is available (https://github.com/kannannova/590Project/ Kannan Nova-DigitalPoster.pptx at GitHub repository.

This project digit poster is as follows

Graphical user interface, text

Description automatically generated

The poster provides the high-level overview of the project and its important artifacts and components. It has Context and Background, components of project – model, implications, hardware/software, UI and Output-Preview and References.

### Project Presentation Phase

The project presentation video content is available at loom https://www.loom.com/share/3af978249dfc440b83671414cf3c1471 and GitHub (https://github.com/kannannova/590Project)

### Final Project Submission

All the project files including python code, documents, video (https://www.loom.com/share/3af978249dfc440b83671414cf3c1471 and poster are available at https://github.com/kannannova/590Project

### GitHub Repository

* The following files are uploaded into GitHub
* KannanNova\_590\_Capstone-Project.docx
* KannanNova\_590\_Project\_Video\_Presentation.mp4
* Kannan Nova\_590\_DigitalPoster.pptx
* KannanNova\_Review of Literature.docx

#### Code

* Wms.py
* wms.ipynb
* app.py
* request.py
* wmsdata.csv
* index.html

### Industry Terminology

The commonly used few warehouse – logistics industry terminologies are given below.

Industry Terminology

|  |  |
| --- | --- |
| **3PL** | Third Party Logistics – a warehouse or group of warehouses managed on behalf of the owner of the stock. |
| **Active Stock** | Stock in active pick locations and ready for order fulfilment. |
| **Aisle** | Any passageway within a storage area. |
| **Backorder** | A piece of stock ordered but out of stock and promised to be shipped when the item of stock becomes available. |
| **Bar Coding** | A way of encoding data for fast and accurate readability. Bar codes are a series of alternating bars and spaces representing encoded information which is read by scanners. |
| **Cargo** | Merchandise to be carried by some form of transportation. |
| **Compliance** | All products, services, processes and documentation comply with specific requirements. |
| **Consolidation** | Combining two or more shipments. |
| **Cross Aisle** | A passageway at right angles to main aisles, used for the movement of supplies, equipment and staff. |
| **Cross Docking** | Merchandise that is received at the warehouse or DC is not put away but prepared for shipment to retail stores. |
| **DC** | Distribution Centre |
| **FIFO** | First In First Out – using stock based on when it was received. |
| **Fixed Slot** | A slot that is reserved for a specific SKU (stock taking unit). |
| **Fulfillment** | Fulfilling a customer order. |
| **Order Picker** | A staff member that is assigned to make withdrawals of warehousing units. |
| **Pallet** | Is a flat transport structure that supports goods in a stable fashion while being lifted by a forklift, pallet jack, front loader, work saver, or other jacking device, or a crane? |
| **Pallet Picking** | Retrieval of full pallets. |
| **Picking** | A staff member pulls the relevant stock items from storage areas to complete a customer order. |
| **Put Away** | Unload the products from cargo and place it into final location or storage area in warehouse |
| **Pick List** | A list of stock items to be picked to fill an order. |
| **POD** | Proof Of Delivery |
| **RFID** | Radio Frequency Identification – electromagnetic field to identify and track tags on objects. |
| **WMS** | Warehouse Management System – systems used to effectively manage processes, activities within the warehouse. |
| **Zone Picking** | Subdividing a picking list by areas for more efficient picking. |

### Conclusion

The model provides the shortest optimal path to picking process workers and robots. This capstone project will contribute to the warehouse or logistics industry which will benefit picking and put a way process and will also be able to improve warehouse’s performance and efficiency. Due this efficiency and performances of the whole warehouse, the business will see the good return of investment (ROI) soon. The project design is generic, and it will work for most of the warehouses if their warehouse has the grid or matrix kind of floor layout. In future, the same project can be used for put away operations too by just applying reverse function of python in the results because put away and picking are using the same warehouse floor layout and put away process is exactly reverse operation of picking process. Thanks, Grand Canyon University for helping to complete this project.

References

Aissa, N., & Guerroumia, M. (2016). *Semi-Supervised Statistical Approach for Network Anomaly Detection*. Retrieved from <https://core.ac.uk/download/pdf/82602135.pdf>

Bhatt, S. (2019, April 19). *Reinforcement Learning 101*. Medium. https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292.

Bonner, A. (2019, November 17). *The complete beginner's guide to data cleaning and preprocessing.* Medium. https://towardsdatascience.com/the-complete-beginners-guide-to-data-cleaning-and-preprocessing-2070b7d4c6d.

Broda, E. (2019, September 19). *Why cloud config mgmt is mandatory for ai/machine learning.* Retrieved April 21, 2021, from <https://medium.com/@ericbroda/why-cloud-config-mgmt-is-mandatory-for-ai-machine-learning-8d376b2b7459>

Brownlee, J. (2020, August 27). *Save and Load Machine Learning Models in Python with scikit-learn. Machine Learning Mastery*. https://machinelearningmastery.com/save-load-machine-learning-models-python-scikit-learn/.

Buchanan, I. A. N. (2018). *Configuration management*: definition and benefits. Atlassian. <https://www.atlassian.com/continuous-delivery/principles/configuration-management>.

Chaudhary, M. (2020, August 14). *Importance of principal component analysis.* Medium. https://medium.com/@cmukesh8688/importance-of-principal-component-analysis-e9184a47ffa8.

Creswell, J, & Creswell, D. (2017). *Research Design:* Qualitative, Quantitative, and Mixed Methods Approaches (5th Edition). Retrieved from <https://www.gcumedia.com/digital-resources/sage/2018/research-design_qualitative-quantitative-and-mixed-methods-approaches_5e.php>

Davis, F. (2019, June 11). *Global consumer Insights Survey 2019: PwC*. Digital Marketing Community. https://www.digitalmarketingcommunity.com/researches/global-consumer-insights-survey-2019/.

Dawood, A., Kakade, P., & West, H. (2020, November 18*). Difference between system testing vs integration testing?* Software Testing Class. https://www.softwaretestingclass.com/difference-between-system-testing-vs-integration-testing/

Dhanani, S. (2018, January 8). *Code Reviewing Data Science Work.* Medium. https://medium.com/@shanif/code-reviewing-data-science-work-774747248e33.

Gajare, S. (2019, October 03). *Data science methodology and approach*. Retrieved from <https://www.geeksforgeeks.org/data-science-methodology-and-approach>

Gillman, D. (2016, May 16). *5 Ways Machine Learning Improves Warehouse Operations.* Retrieved from <https://www.toolbox.com/tech/erp/blogs/5-ways-machine-learning-improves-warehouse-operations-051616>

Hanks, C. (2018, October 31). *Your Guide to Warehouse Terms.* Wonolo. <https://www.wonolo.com/blog/your-guide-to-warehouse-terms/>

Hazen, B., Boone, C., Ezell, J., & Jones-Farmer, L. (2014, April 26). *Data quality for data science, predictive analytics, and big data in supply chain management:* An introduction to the problem and suggestions for research and applications. Retrieved February 11, 2021, from <https://www.sciencedirect.com/science/article/pii/S0925527314001339>

Jain, A. n. (2020, May 30). *Why turn into a Pickle?* Medium. https://towardsdatascience.com/why-turn-into-a-pickle-b45163007dac.

Janse, L., & Rensburg, V. (2019). *Artificial Intelligence for warehouse picking optimization.* – An NP-Hard Problem. Retrieved from <http://uu.diva-portal.org/smash/record.jsf?pid=diva2%3A1413926&dswid=4306>

Jordan, J. (2020, November 12). *Effective testing for machine learning systems.* Jeremy Jordan. https://www.jeremyjordan.me/testing-ml/

Jordan, J. (2021, January 3). *Organizing machine learning projects:* Project management guidelines. Jeremy Jordan. https://www.jeremyjordan.me/ml-projects-guide/

Kibish, S. (2018, April 10). *A note about finding anomalies*. Retrieved March 29, 2021, from <https://towardsdatascience.com/a-note-about-finding-anomalies-f9cedee38f0b>

Kiili, J. (n.d.). *Reinforcement learning Tutorial Part 1: Q-Learning*. MLOps Platform - Train, Evaluate, Deploy, Repeat. https://valohai.com/blog/reinforcement-learning-tutorial-part-1-q-learning/.

Koen, S. (2019, August 09). *Architecting a machine learning pipeline.* Retrieved April 21, 2021, from <https://towardsdatascience.com/architecting-a-machine-learning-pipeline-a847f094d1c7>

Lau, D. C. H. (2019, January 10). *5 Steps of a Data Science Project Lifecycle. Medium.* https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492.

LeBlanc, K. (2020, October 8). *Tips for effective warehouse numbering schemes.* ID Label Inc. https://idlabelinc.com/tips-for-effective-warehouse-numbering-schemes/.

Lopienski, K. (2020, November 31). *Warehouse order Picking 101: Systems, Methods, & strategies.* Retrieved from https://www.shipbob.com/blog/warehouse-picking/

McCrea, B. (2016, August 19). *8 new ways to think About putaway and storage in your warehouse.* Retrieved April 20, 2021, from <https://www.mmh.com/article/8_new_ways_to_think_about_putaway_and_storage> in\_your\_warehouse

Meardon, E. (2016). *What is a Gantt chart?* Atlassian. https://www.atlassian.com/agile/project-management/gantt-chart.

Phan, B. (2020, June 29). *10 Minutes to Deploying a Deep Learning Model on Google Cloud Platform*. Medium. <https://towardsdatascience.com/10-minutes-to-deploying-a-deep-learning-model-on-google-cloud-platform-13fa56a266ee>.

Seno, T. (2017, October 21). *Welcome to Deep Reinforcement Learning Part 1 : DQN*. Medium. https://towardsdatascience.com/welcome-to-deep-reinforcement-learning-part-1-dqn-c3cab4d41b6b.

Shashanka, M. (2019, December 13). *WHAT is a pipeline in Machine learning?* how to create one? Medium. https://medium.com/analytics-vidhya/what-is-a-pipeline-in-machine-learning-how-to-create-one-bda91d0ceaca.

Shyalika, C. (2019, November 16). *A Beginners Guide to Q-Learning.* Retrieved from https://towardsdatascience.com/a-beginners-guide-to-q-learning-c3e2a30a653c

Singh Gill, N. (2020, December 06). *Machine learning pipeline deployment, architecture and tools.* Retrieved from <https://www.xenonstack.com/blog/machine-learning-pipeline/>

T. M. D. (2021, July 1). *What is UI vs. UX design? What's the difference?* Medium. https://uxplanet.org/what-is-ui-vs-ux-design-and-the-difference-d9113f6612de.

Vaughan, J. (2019, November 1). *What is data quality and Why is it important?* SearchDataManagement. https://searchdatamanagement.techtarget.com/definition/data-quality.

Violante, A. (2019, July 1). *Simple reinforcement learning: Q-learning.* Medium. https://towardsdatascience.com/simple-reinforcement-learning-q-learning-fcddc4b6fe56.

Zimmerman, M., & Sullivan, D. (2019, October 04). *Secure your machine learning pipeline.* Retrieved from https://blogs.vmware.com/ml-ai/2019/10/03/secure-your-machine-learning-pipeline