

Smartphone Addiction Prediction Using Machine Learning

Ritu Bhamrah

Department of Applied Data Science, San Jose State University

DATA 270: Data Analytic Process

Dr. Eduardo Chan

December 10, 2021

Abstract

The exponential growth of smartphone addiction causes issues like health, financial, psychological, and social problems. Most of the studies that are done to understand the smartphone addiction problem have been focused on survey-based data. However, it is difficult to understand the problem merely considering the psychological factors based on survey-based data. This project focus on analyzing the smartphone addiction problem using log data and the log data for 500 users is provided by Apple. The data provided is the labelled data with six descriptive features and a target feature addictive(zero)/not addictive(one). The log data is being analyzed using Logistic regression and K nearest neighbor machine learning techniques. For logistic regression the combination of hyper tuned parameters chosen are penalty = L2 and c = one and for k nearest neighbor the combination of hyper tuned parameters chosen are n_neighbors = 17 and metric = Euclidean. The K fold cross validation has been applied on both the models using tuned parameters. The logistic regression model provided us the best results with accuracy score of 93 percent and f1 score values as 0.95 and 0.89 for zero and one respectively whereas k nearest neighbor model provided the accuracy score of 91 percent and f1 score values as 0.93 and 0.86 for zero and one respectively. The best model predicted in this study can be leveraged to analyze the smartphone addiction problem using stream data in the future.

Keywords: smartphone addiction, k nearest neighbor, logistic regression

Chapter 1 - Introduction

1.1 Project Background and Executive Summary A

The world has changed so much after the digital revolution. We have evolved from telegrams, postcards that were used to send the letters, to now smartphones that can provide text messages, audio-video calls, social media apps, and gaming apps in just one click. It is found that “the number of users worldwide was more than 1.08 billion in early 2012 and continues to increase exponentially” (Mok et al., 2014, p. 10).

Smartphones have improved our quality of life significantly, but have we ever thought that smartphones can be addictive? According to Lee & Kim (2021), smartphones are more dangerous for the young people as they found various ways to abuse the technology. The usage is increased for many age groups including infants. It had been also found that the highest usage can be seen in adults in past two years where it reached to 17.4 percent from around four percent.

The smartphones have become very addictive that people couldn't control looking at their phone even when they are driving, which could be very dangerous. The smartphones have changed the dynamics of the social gatherings, where people are more interested in digital world than interacting with people around. This could create a lot of personality disorders. The more damaging trend is happening in children due to smartphone addiction because children are not interested in participating in outdoor activities. On broader level smartphone addiction causes issues like health, financial, psychological and social problems. (De-Sola et al., 2016)

There are different studies that have been done to understand the problem of screen addiction. But the problem with most of these studies is that merely considering the psychological factors of the smartphone users using survey-based questions. One of the research

is done where data is collected by KISA every year in the form of survey. The machine learning techniques utilized on this data were random forest, decision tree, and Xgboost (Lee & Kim, 2021).

Analyzing smartphone addiction based on the survey data is not a good way since the data collected is not the actual usage data. Survey based data assumes that all the participants have answered the questions honestly and accurately. A smartphone user might think that his or her overall smartphone usage is two hours, but the user might use the device from four to six hours. Such kind of data results in inaccurate modelling results.

Keeping this in mind, to overcome this problem, this project aims on analyzing the problem using real time log data of different smartphone users. The log data will be analyzed using supervised machine learning techniques such as Logistic regression, , Naïve Bayes, k nearest neighbor, support vector machines, Decision Tree Classification, and Random Forest Classification. The best suitable model will be selected for this project based on the best evaluation metrics. The main goal of this project is to predict whether the user is addictive or not with high accuracy.

1.2 Project Requirements

The data requirement of this project is to collect real time log data for 500 different users which will be historical data. This batch data will be backed up for disaster recovery purposes. The selection of these 500 users should be in such a way that it covers distinct features such as app id, device id, user id, first name, last name, gender, age, from date, and to date. We will need the following log data parameters for these distinct users - social networking, entertainment, gaming, productivity, total screen time, addictive/not-addictive. These parameters will be used as inputs for our algorithm. The feature extraction process of machine learning will use these

inputs and extracts the descriptive features as gender, age, social networking, entertainment, gaming, productivity, total screen time.

The primary goal of this project is to use the above descriptive features from the dataset and attempt to predict whether the user is addictive or not using machine learning algorithms. The target feature of this analysis is binary value zero or one to indicate whether the user is addictive or not addictive. Zero indicates not addictive and one indicates addictive. The name for the target feature is Addictive/Not-addictive.

The type of machine learning technique which will be used to test this project is classification model which comes under supervised learning. Some of the examples of algorithms that comes under classification model are logistic regression, k nearest neighbor, etc. The results will be tested under various classification models and the one with the best accuracy will be the final model. The functional requirements for this project are as follows:

- Data cleaning and validation of historical data, this will be achieved using Google dataflow where noisy data will be handled. We will also ensure that data is valid, accurate, consistent, and uniform. We will also create data quality reports before and after cleaning of data which will help us to understand the factors like cardinality, missing value percentage, etc.
- Integration of big query, this is used to store model results, once we do the model validation and evaluation, results will be stored on bigquery.
- Classify addicted or not addicted user, this will be achieved using machine learning model. Zero indicates if the person is not addictive and one indicates if the person is addictive.

- Google cloud storage, this will be used to store the batch data and to store data after transformation.
- Data studio, if any kind of visualizations are required, that will be done using Google data studio.
- Model evaluation metrics, accuracy and f1 score will be used to assess the models where model accuracy should be above 90 percent and the f1 should also be more than 0.80.

1.3 Project Deliverables

Project deliverables usually are the completion of tasks, milestones or specific deliverables like a report, documents, or a product. Below are the deliverables of this project:

- Project plan, it is detailed document about the project execution during the initial phase of the project and will be updated in subsequent phases accordingly.
- User log data report, this will be delivered during initial phase of the project. This document has all the input data collected for the analysis.
- Risk assessment report, this will be delivered during the initial phase of the project. This document will have the risks and pitfalls in the project.
- Data visualization report, it is delivered during the final phase of the project. This document will have the visualization of all the machine algorithms that have been applied on the data.
- Final report, will be delivered at the end of the project. This is one of the main deliverables that has all the findings and future direction of the analysis. This report explains all the models used and how they are used and interpreted using the user data.

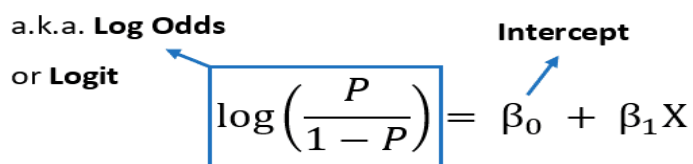
1.4 Technology and Solution Survey

In my observation the studies done in the past on smartphone addiction have used different machine learning algorithms that fall under classification models. To enhance the prediction accuracy score, I think the analysis should be done on the logs collected from devices instead of survey-based data. The problem with the survey data is that people can mention what according to them is correct. Therefore, in this project I would like to collect the historical smartphone logs from Apple on which I will apply machine learning classification models. This would help us to choose the outperforming model for prediction of smartphones addiction of a user. On the basis of technologies used in past, below are the classification model algorithms approaches that can be used to predict smartphone addiction.

Logistic regression, which is used to predict if an event is occurring or not. In this project, I could use this to predict if the user is addictive or not addictive to smartphone. This algorithm takes use of the sigmoid function to distribute the target variable. The model tries to find the best fitting relationship between target feature (addictive/not addictive) and a set of descriptive features.

Figure 1

Logistic Regression Formula



$$\log\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X$$

a.k.a. **Log Odds**
or **Logit**

Intercept

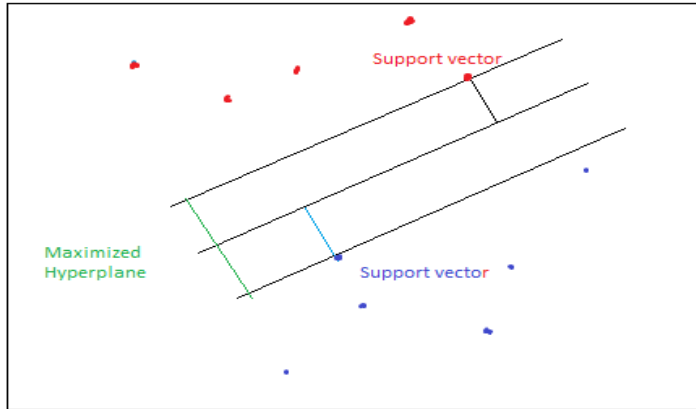
Note. The Figure 1 shows logic regression calculation formula. Reprinted from *Interpret the Logistic Regression Intercept* by Choueiry, G (2021). Copyright 2021 by Choueiry, G.

Figure 1 shows the y is value being replaced which shows the probabilistic approach which is one the advantage of this model. However, some assumptions of logistic regression can be considered as the disadvantages.

K-Nearest Neighbor (KNN), which is helpful in the scenario where there are two categories already present in our dataset. In our scenario the two categories are addictive and not addictive. To find the unknown data received, KNN applies to find the closest k number of nearest neighbors based on various distance calculation metrics like Euclidean distance, Manhattan distance, etc. The KNN algorithm will sort the data in ascending order and then assign the new point to the class which has the highest weight. The most common distance calculation metric used for KNN is Euclidean distance with its formula as $d = \sqrt{[(a_2 - a_1)^2 + (z_2 - z_1)^2]}$. Here, a_2 and z_2 are the query points and a_1 and z_1 are dataset points. Similarly, distance calculation from query and all the dataset points will be done.

The major disadvantage of KNN would be choosing the k value. This is very important as it can affect the overall results. However, the model is very fast and efficient in solving the problems which is one of its major advantages.

Support vector machine (SVM), which is about finding the best line between two distinctly classified datapoints in N-dimensional space. Here, N is the total fields.. A hyperplane needs to be identified that classifies the datapoints into addictive or not addictive. Figure 2 explains this by showing two support vectors with two different class of datapoints. The Hyperplane shown in Figure 2 should be chosen in such a way that it should have maximum value.

Figure 2*Support Vector Machine Illustration*

Note. The Figure 2 shows the two support vectors and the margin which should be maximum distance.

The major advantage of SVM is that it not sensitive to issues like overfitting. However, the disadvantage of model is that it can't be used on linear data. So, the model should be chosen wisely considering the facts of dataset.

Naïve Bayes, which uses bayes theorem that presume that one field is independent of others. Even with this assumption, this model surprisingly performs well in many cases, and it is one of the best models in machine learning.

The assumptions of naïve bayes are considered as drawback. However, the probabilistic approach of model is one of the major advantages for which it is widely used.

Figure 3 shows the formula for naïve bayes theorem and how this conditional classification can be solved using these formulas.

Figure 3*Naïve Bayes Formula*

$$P(A|B) = [P(B|A) * P(A)] / P(B)$$

$$P(A|B) = P(B_1|A) * P(B_2|A) * \dots * P(B_n|A) * P(A)$$

Note. The Figure 3 shows the naive bayes theorem and formula for calculating conditional classification using naïve bayes theorem.

Decision tree classification where the model has a tree and leaves like structure. It utilizes the if-then rules and the process continues until the final point is reached. Each descriptive feature in the dataset will be used as a node. Each descriptive feature will be used as if it is a question, and the results will be derived based on the possible outcomes. This algorithm involves complex mathematical calculations like entropy, entropy remainder, information gain, etc.

The major advantage of decision tree is that it does not require the feature scaling. However, the model cannot perform well on small datasets which is its drawback.

Random Forest Classification, which is also sometimes called as Random decision trees because it operates by creating multitude of decision trees. This model tries to fit multiple trees on subset of datasets and uses the mean of these subsets to enhance the prediction. Random forest classification has better accuracy than the decision tree classification because of its structure that reduces the over fitting. The dataset in this project can be divided into subsets using descriptive features like age, gender, and job classification. The sub classification can be

done either on one feature or multiple features. The better accuracy can be achieved by dividing subsets using multiple features.

One of the major pros of random forest classification is its outstanding performance due to which it is widely used. However, we need to choose the number of trees which can be drawback of this model as this can affect the overall results.

1.5 Literature Survey of Existing Research

Various supervised machine learning models have been used by different researchers to predict the addictive behavior towards the smartphone. The research by Chaudhury & Kumar (2018) used different machine learning algorithms to understand the impact of smartphone addiction in relation to academic performance. They have used Naïve Bayes, Support vector machine (SVM), and RBF classifier models on the data that was collected in the form of questionnaire from engineering college students. Their study concluded that SVM model gave the highest accuracy around 81 percent among the three models used. Their research suggests that smartphone addiction negatively affects the students' academic performance and thus smartphone usage should be discouraged among students.

In another research by Lee & Kim (2021) suggest that smartphone addiction causes harmful effects in not only students but also in people belong to various age groups from infants to senior citizens. This research is conducted on data collected from 29,712 people by KISA where researchers have used decision tree, random forest, and Xgboost machine learning algorithms. The data collected for this research was various age groups and was focused on the psychological factors of the participants. Their research concluded that random forest model achieved the highest level of accuracy among the three with an accuracy around 83 percent.

In another research done by Lee et al., (2018), the authors tried to understand the smartphone addiction problem by analyzing the difference in data between self-reporting and automatically collected data. The proposed data collection method for this project was using a mobile application. However, though it is automatically collected data, authors have pointed out a lot of limitations while using the mobile application. The model used by them for this problem is multiple linear regression decision tree which achieved accuracy of 89.7 percent. The authors concluded that the average usage of smartphone by per user is as high as six hours a day and the users are likely to check their phone as high as 300 times unconsciously.

In other research, the offline and online question-based survey was collected from different students. This research used KNN and logistic regression machine learning algorithms on the data collected. This study obtained the best results with KNN model with accuracy around 96 percent. This study suggests that smartphone addiction causes mental disorders (Baby & Priya, 2021).

The Table 1 shows the summary of work discussed above and the detailed comparisons between different research on smartphone addiction.

My observation of these studies is that the main purpose of them is to predict the harmful nature of smartphone addiction to human beings. Some researchers focus on a particular group of people whereas some other researchers focus on different problems varying depending on the demographic data. I opine that any study needs large amount of data to be able to draw accurate results and the results drawn on small set of data are not reliable.

Table 1

Comparison Summary of Previous Work Using Machine Learning

Authors	Dataset	Models Used	Highest Accuracy	Conclusions
Baby & Priya, 2021	Offline & online question-based survey	KNN & Logistic Regression	96% with KNN model	This study suggests that smartphone addiction causes mental disorders
Lee et al., 2018	Smartphone usage pattern & self-report information	Multiple linear regression decision tree	89.7 %	Average smartphone usage of a user is as high as six hours a day and smartphone users likely to check their phone as high as 300 times unconsciously
Chaudhury & Kumar, 2018	Questionnaire from engineering college students	Naïve Bayes, Support vector machine (SVM), and RBF classifier	81 % with SVM model	Negatively affects the students' academic performance
Lee & Kim 2021	Data collected by KISA for 29,712 people	Decision tree, random forest, and Xgboost	83 % with random forest model	Harmful effects on different age groups including infants & senior citizens

Note. The Table 1 is the tabular comparison summary of work discussed above. Research done by Baby Lee et al., (2018) is from *Digital screen addiction with KNN and -Logistic regression classification*, research by Lee et al., (2018) is from *Analysis of Behavioral Characteristics of Smartphone Addiction Using Data Mining*, research by Chaudhury & Kumar, (2018) is from *A Study on impact of smartphone addiction on academic performance*, research by Lee & Kim (2021) is from *Prediction of Problematic Smartphone Use: A Machine Learning Approach*.

Chapter 2 - Data and Project Management Plan

2.1 Data Management Plan

The data management plan is a crucial part of a project, and it describes data collection method, format of the data, and management of the data throughout the life cycle of the project. For this project, the user interactions with iPhone devices will be collected in the form of JSON format. This data includes each iPhone user's daily activity on how much time the user spent on the device and type of applications the user has accessed. This data also includes time spent by each user for each category of applications.

There is no data collection directly from the users. The device manufacturer Apple collects the data from remote smartphone devices. The data collection approach to collect data by the manufacturer will be telemetry. According to Altvater (2021), the telemetry is a method to collect data from various devices which are located at different locations, and it collects the data continuously for different users using electrical pulses and this raw data transmitted to different servers for further analysis.

For this project, Apple will provide us six months historical data of 500 users, and they manually labeled the data as addictive and not addictive. This dataset will be used to train our model. The classification models will be applied on this batch data to identify the most suitable model for such kind of data. Based on this analysis, the selected model can be leveraged to classify the new stream data into addicted or not addicted. The historical data collected for this project will be 25 GB. As Apple has provided us the data using Google cloud storage, the data will be accessible for long term directly from the cloud. The data on Google cloud platform will be accessed using python programming.

As part of data management process, we will secure the data and use the data efficiently. To assure the quality of the data, we will make sure the data is relevant, up to date, and accurate during data preparation phase. The data with missing user id, app id, or missing time will be removed from the batch data. The data with robotic events or duplicate events will also be removed during this process. The metadata for the dataset and evaluation results of this project is schema details. The schema details will have information like user id, device id, app id, etc.

Retention of data will be made based on the user's location. We will apply various compliance processes like GDPR to make sure that the personal data will be discarded after six months, and the data will not be shared with any other party. The device manufacturer will take care of compliance of users` consent. We will protect the identity of participants via anonymization. The data will be masked before processing and personal information like names, date of birth, email will be dropped. Cloud storage is protected by IAM roles which will secure the usage of data by user account. We will use the secure HTTPS layer to transfer data between dataflow and cloud storage for better security of the data. We will abide by the data privacy rules of the device manufacturer and the data will be provided directly into the cloud storage. We will also follow the Google cloud platform rules to maintain the privacy of data.

There will be three multi region cloud storage buckets used for this project. The three buckets are standard, coldline, and archive.

- The standard bucket will store the batch data.
- The backup of batch data will be stored on Coldline Storage bucket for disaster recovery purposes.
- The evaluation results will be stored in archive bucket for long time access.

We will use the bigquery to store the evaluation results will also be stored on bigquery for short period of time. The evaluation results will be stored on cloud storage for long term and users` raw data will discarded after six months from the google cloud storage. Both Google cloud storage and bigquery storage will be charged according to the usage. Our team will be responsible for backup and recovery of data. The data science division team in Apple will be responsible for data management. For completion of this project, various service licenses such as dataflow, cloud storage, project management tool, etc. will be required. These service licenses will be costed according to the usage.

2.2 Project Development Methodology

We will use two methodologies for this project. The first methodology is Agile which will help us to complete the project efficiently. The second methodology is CRISP-DM which ensures completion of tasks in a sequence using different phases. The CRISP-DM methodology integrates with Agile wherein all phases of CRISP DM will be an Agile sprint.

There will be five different phases in CRISP DM and each phase will give us specific results at the end of the phase. The results of one phase will be used by next phase in progressive and iterative way. At the final phase we will be able to compare the model prediction with other models to identify the best model to solve the smartphone prediction. Below are the methodology phases for smartphone addiction using machine learning:

Business Understanding phase - This phase begins with an attempt to understand the need for smartphone prediction. In this phase, we will identify the objectives, scope, requirements, project plan, project architecture, and hardware and software requirements for this project. The deliverables of this phase are:

- Project plan smartsheet along with the start date and end date of each item.
- Functional requirement documents that specify details of the requirements along with the scope and limitations.
- Project architecture document that has visual representation of different layers in the project.

Data Understanding phase - In this phase, the historical smartphone logs are provided by the manufacturer. These logs will be stored on Google cloud storage for data analysis. The deliverable of this phase is:

- Summary report of the dataset and indicating descriptive features of the dataset.

Data Preparation phase – In this phase, the data will be cleaned and enriched. The process of cleaning and enriching will result accurate data by suppressing noisy data. The bigquery tables that are required to store the enriched data created in this phase. By the end of this phase, the cleaned batch data will be moved to bigquery tables. The deliverables of this phase are:

- Summary report of the data that is being saved to bigquery.
- Apache beam python code that is used to complete this phase.

Modeling phase – In this phase, various classification models such as logistic regression, naïve bayes, etc. will be applied on the enriched data. The deliverables of this phase are:

- Detailed report on the results of the various classification models that were applied on the data.
- Python code using scikit library to apply various classification models.

Evaluation phase – In this phase, we will compare results from different models to evaluate the model with highest accuracy. The deliverables of this phase are:

- Detailed report showing the different models and their accuracy percentages.
- Python evaluation code using scikit learn library to perform the above task.

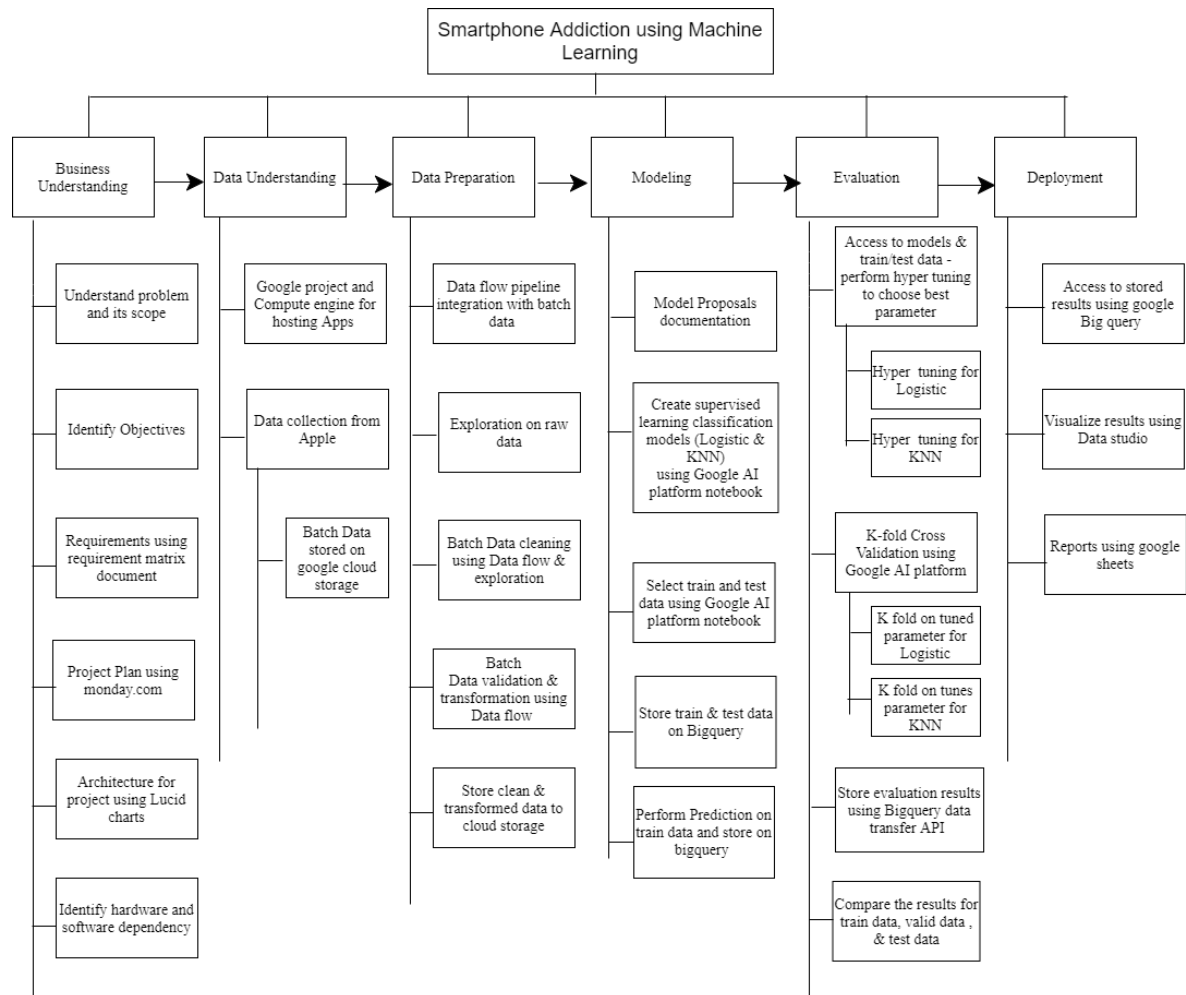
Deployment phase – This is the final phase of the project and in this stage, the visualizations of the results are performed. The visualization tool data studio will be used to represent the results. The deliverables of this phase are:

- Data studio dashboard report showing evaluation results.
- Final project report that has key findings, conclusions, and future enhancements.

2.3 Project Organization Plan

The project organization plan defines the different tasks which will be performed to achieve the project completion. This project will be executed in different phases - business understanding, data understanding, data preparation, modeling, evaluation, and deployment. Each of the phases have a series of tasks that are performed in an orderly fashion to complete the phase. Figure 4 shows the work down structure of this project to show the phases and decomposition of each phase into different work packages and deliverables.

Some of the major tasks from work breakdown structure are understanding problem, defining scope and objectives, batch data collection, data cleaning, data validation, and delivering the best model with its prediction results. Each activity shown in Figure 4 is a work package and it can be delivered by a single person. Every activity in Figure 4 delivers a specific deliverable at the end of the activity.

Figure 4*Work Breakdown Structure*

Note. The Figure 4 shows the work breakdown structure with different phases of CRISP DM.

2.4 Project Resource Requirements & Plan

To complete the tasks mentioned under work break down structure, we will need various tools, licenses, and cloud services. Table 2 represents the hardware and software requirements to complete the project. Table 3 represents the cost estimation of different services and licenses. The total duration of the project is around two months and the total cost to complete the project will be 197.58 USD.

Table 2*Hardware and Software Requirements.*

Hardware	Version
Instance server	n1-standard-1 (1 vCPU, 3.75 GB memory)
Apache beam server	n1-standard-2 (2 vCPU, 7.5 GB memory)
Cloud storage	1 standard bucket, 1 coldline bucket, 1 archive bucket
Software	Version
Python	3.7.3
Apache Beam	2.25.0
Operating System	Linux (Debian 10)

Note. The data for hardware and software is from

<https://cloud.google.com/products/calculator>. Copyright 2021 by Google.

Table 3*Monthly Cost Estimate and Licenses Requirements*

Service	Specifications	Justification	Estimated cost (USD)
Compute engine	n1-standard-1	To host different applications	40.97
Cloud storage	standard Storage	To store batch data	0.60
Cloud storage	coldline Storage	Disaster recovery of batch data	0.12
Cloud storage	archive Storage	Long-term storage of results	1.54
Dataflow	n1-standard-2	To clean and validate batch data	17.62
Bigquery	-	To store clean data and results from prediction	0.94
Monday.com	standard plan	To manage project	10
Lucidchart	team plan	To draw flowcharts, diagrams, etc.	27
Data studio	-	To visualize results from predictions	0
Total			98.79

Note. The data for compute engine, cloud storage, dataflow, bigquery, and data studio are from

<https://cloud.google.com/products/calculator>. Copyright 2021 by Google. The data for

Lucidchart is from <https://lucid.app/pricing/lucidchart#/pricing>. Copyright 2021 by Lucidchart.

The data for Monday.com is from <https://monday.com/pricing>. Copyright 2021 by Monday.

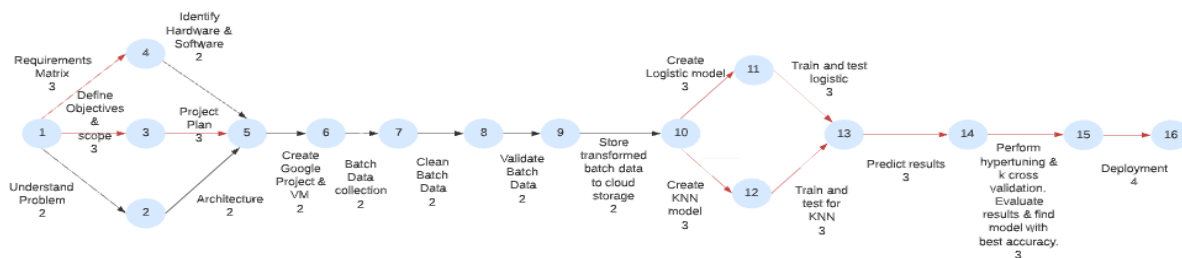
Table 3 shows various services which will be used for this project. There are various cloud storage buckets like standard, coldline and archive will be used. These storage buckets will be used for batch data storage, disaster recovery, and for long-term storage of results which can be accessed in future. We would also need Google dataflow for data preparation and validation and results of model evaluation will be stored on bigquery. The model comparison can be done using data studio. The overall services are hosted using compute engine. Other than Google cloud services we will also need the project management tools like Monday.com for project tasks tracking and lucid charts for creating architectural diagrams required for the project.

2.5 Project Schedule

The project schedules will identify all the milestones and activities that need to be accomplished to successfully deliver the project. This information can be represented using PERT chart and Gantt chart. Figure 5 is the PERT chart representing various milestones and activities. Each node in the Figure 5 represents a milestone and the arrow connecting two nodes represents an activity. Each node predecessor to a node represents the dependency to that node.

Figure 5

PERT Chart



Note. Figure 5 shows the PERT chart for project. The red lines shown in Figure 5 represents the critical path for the project.

Table 4*Calculation Results for Estimated Time*

Activity Name	Activity	Predecessor	Duration (Days)			
			O	M	P	Estimate
A	Understand Problem	-	1	2	3	2
B	Define Objectives	-	1	3	5	3
C	Requirement Matrix	-	2	3	4	3
D	Architecture	A	1	2	3	2
E	Project Plan	B	1	3	5	3
F	Identify Hardware & Software	C	1	2	3	2
G	Create Google project & Virtual machine	D, E, F	1	2	3	2
H	Batch data collection	G	1	2	3	2
I	Clean Batch data	H	1	2	3	2
J	Validate batch Data	I	1	2	3	2
K	Store transformed data to cloud storage	J	1	2	3	2
L	Create Logistic model	K	1	3	5	3
M	Create KNN model	K	1	3	5	3
N	Train & test Logistic model	L	1	3	5	3
O	Train & test KNN model	M	1	3	5	3
P	Predict results	N, O	1	3	5	3
Q	Evaluate	P	1	3	5	3
R	Deployment	Q	2	4	6	4

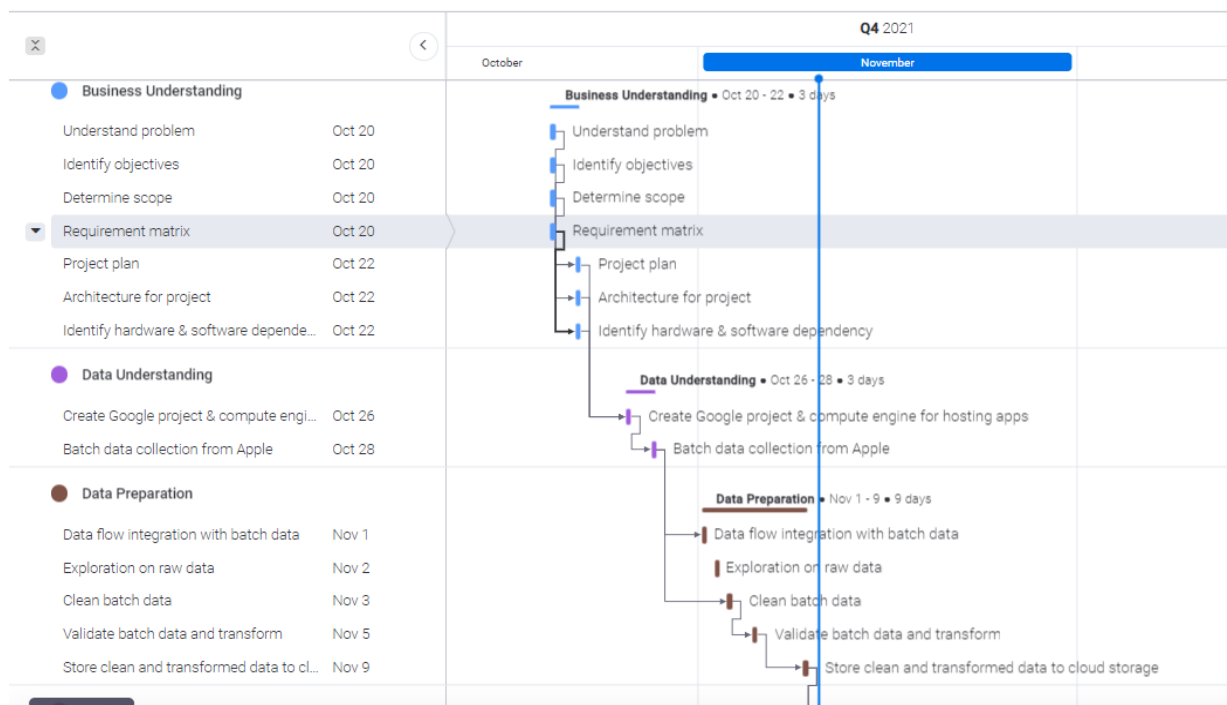
Note. Table 4 shows the calculation results of estimated time using most likely time (M), optimistic time (O), and pessimistic time (P) for each activity in Figure 2.

Table 4 shows that the estimated time for each task is not more than three days. We will involve total four team members for this project. Thus, completion of tasks in short time will not be an issue.

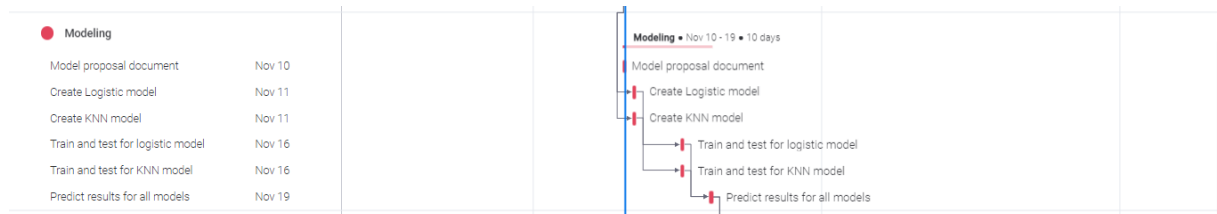
Figure 6, Figure 7, and Figure 8 shows the Gantt chart representing project schedule with phases, activities, timeline, and status of deliverables in months. The connecting lines here shows the dependency of each task. Each activity performed for this project ranges from one to four days. If we change the representation to days, it shows project timeline as Figure 9, Figure 10, Figure 11, and Figure 12 which is in more detailed manner.

Figure 6

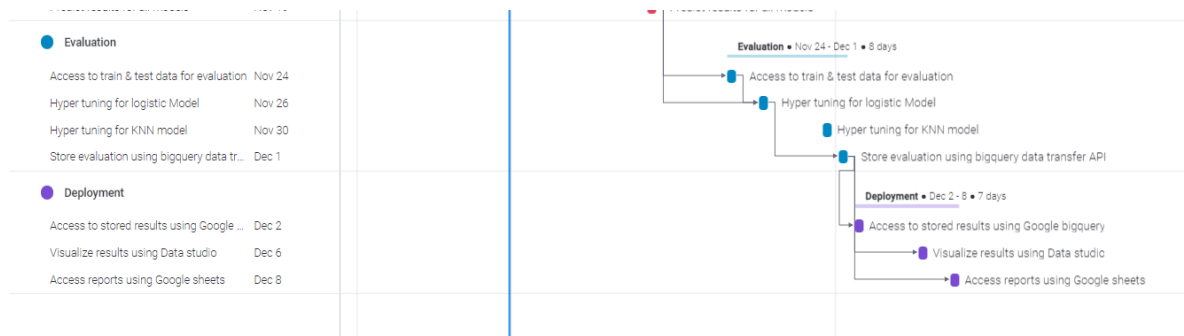
Month's View of Business Understanding, Data Understanding, and Data Preparation



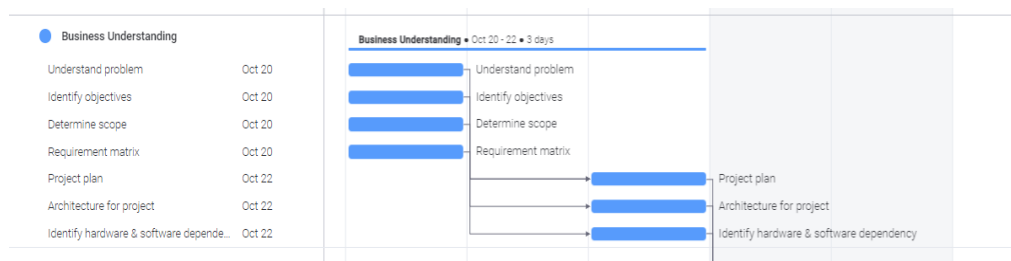
Note. The Figure 6 shows the month view of Gantt chart for business understanding, data understanding, and data preparation phases.

Figure 7*Month's View of Modeling*

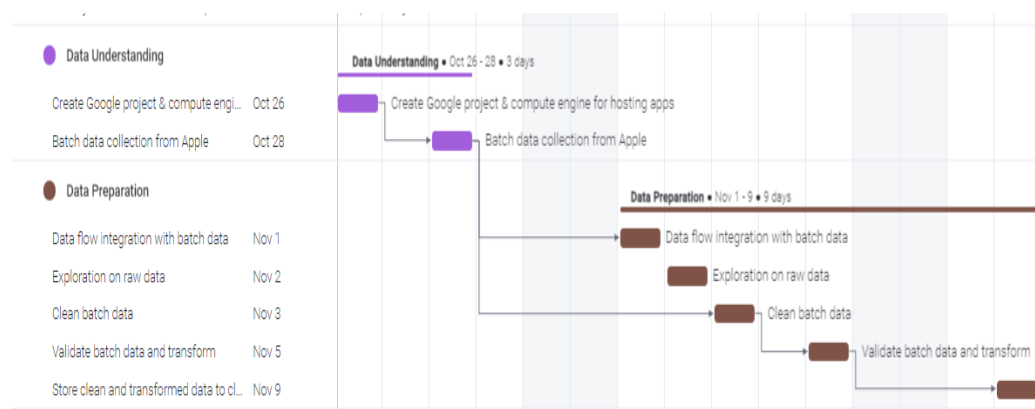
Note. The Figure 7 shows the month view of Gantt chart for modeling phase.

Figure 8*Month's View of Evaluation Phase and Deployment Phase*

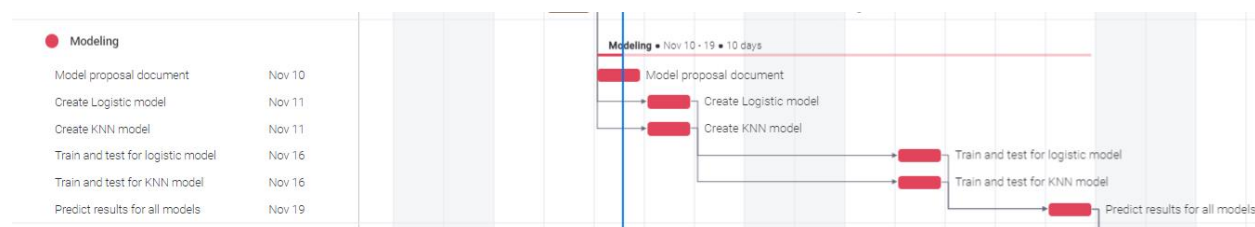
Note. The Figure 8 shows the month view of Gantt chart for evaluation and deployment phases.

Figure 9*Gantt Chart in Days for Business Requirement*

Note. The Figure 9 shows the day view of Gantt chart for business understanding phase.

Figure 10*Gantt Chart in Days for Data Understanding, and Data Preparation*

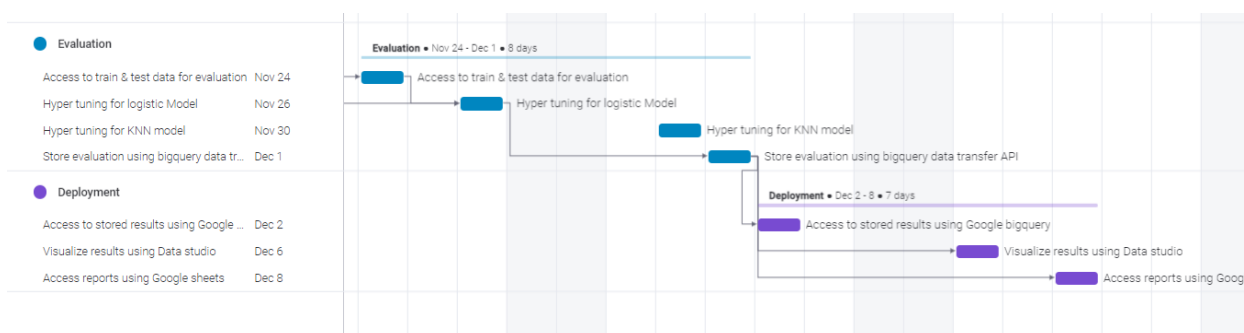
Note. The Figure 10 shows the day view of Gantt chart for data understanding and data preparation phases.

Figure 11*Gantt Chart in Days for Modeling*

Note. The Figure 11 shows the day view of Gantt chart for modeling phase.

Figure 12

Gantt Chart in Days for Evaluation, and Deployment

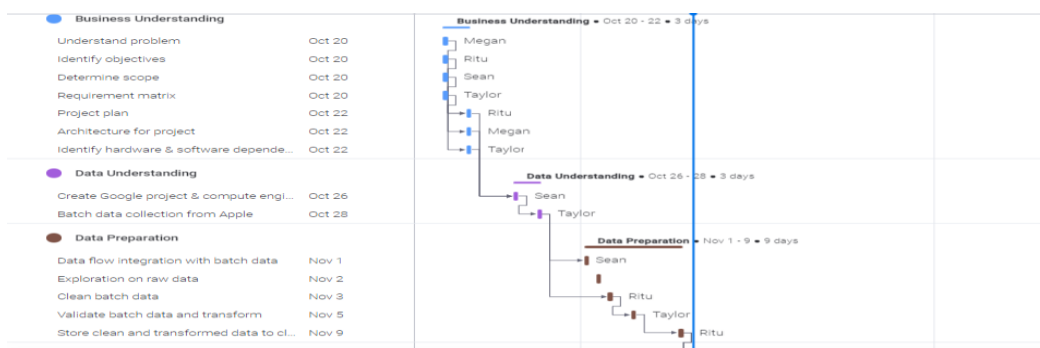


Note. The Figure 12 shows the day view of Gantt chart for evaluation and deployment phases.

The Gantt chart representation in Figure 13 and Figure 14 are with respect to team members of the project. The team members for this project are Ritu, Sean, Meghan, and Taylor.

Figure 13

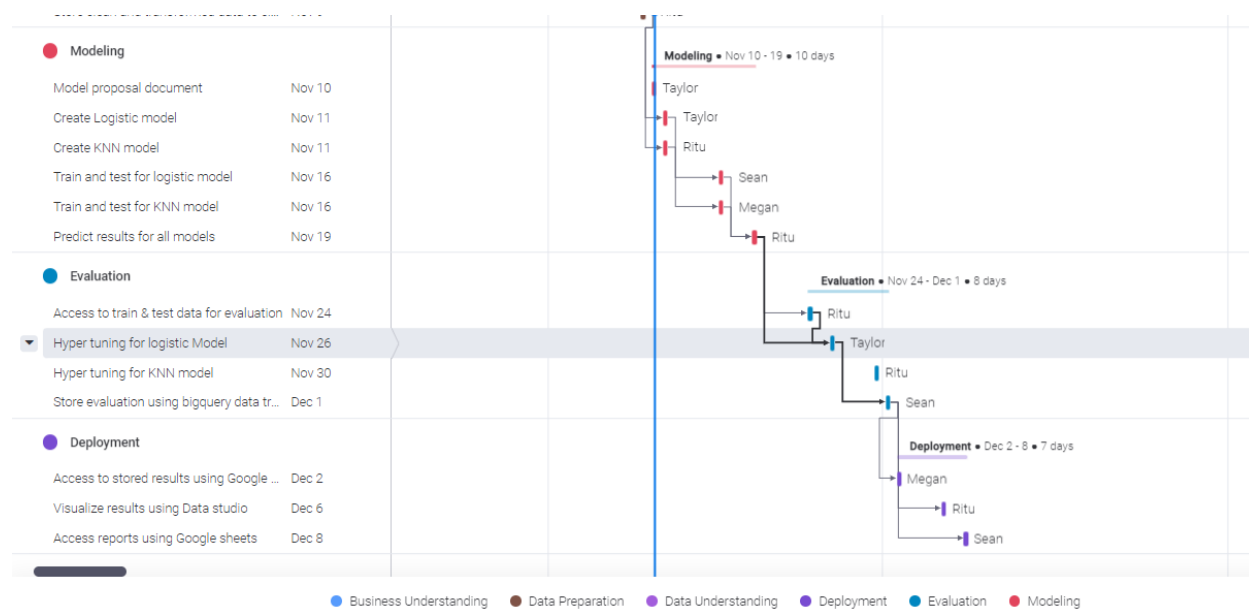
Gantt Chart with Respect to Team Members for Business Understanding, Data Understanding, and Data Preparation



Note. The Figure 13 shows the team members working on different tasks for phases – business understanding, data understanding, and data preparation.

Figure 14

Gantt Chart with Respect to Team Members for Modeling, Evaluation, and Deployment



Note. The Figure 14 shows the team members working on different tasks for phases – modeling, evaluation, and deployment.

Chapter 3 - Data Engineering

3.1 Data Process

In this phase, we will explain the overall architecture for the project. We will create architecture diagram showing all the Google cloud service and how they are being utilized to complete different tasks in the project.

The data is collected from manufacturer Apple. Figure 15 shows the data pipeline architecture diagram and different Google cloud services used for completion of this project. The raw data provided by Apple is stored on Google cloud storage which is a standard bucket. The batch raw data will be backed up on Coldline Storage bucket for disaster recovery purposes.

The batch data will be accessed from the standard bucket for cleaning and transformation. The cleaning and transformation are performed using Google dataflow service to eliminate the data with inconsistencies like duplicate values, missing values, inconsistent measure etc. In dataflow, we would use Jupyter notebook to perform cleaning and transformation steps. As the data with missing values is less than two percent of the data, we would drop the rows with missing values. In dimensionality reduction, we would drop unused fields like app id, device id, user id, first name, last name, from date, and to date to improve model performance. The cleaned and transformed data will be stored on Google cloud storage and it will be utilized for creating test and train sets.

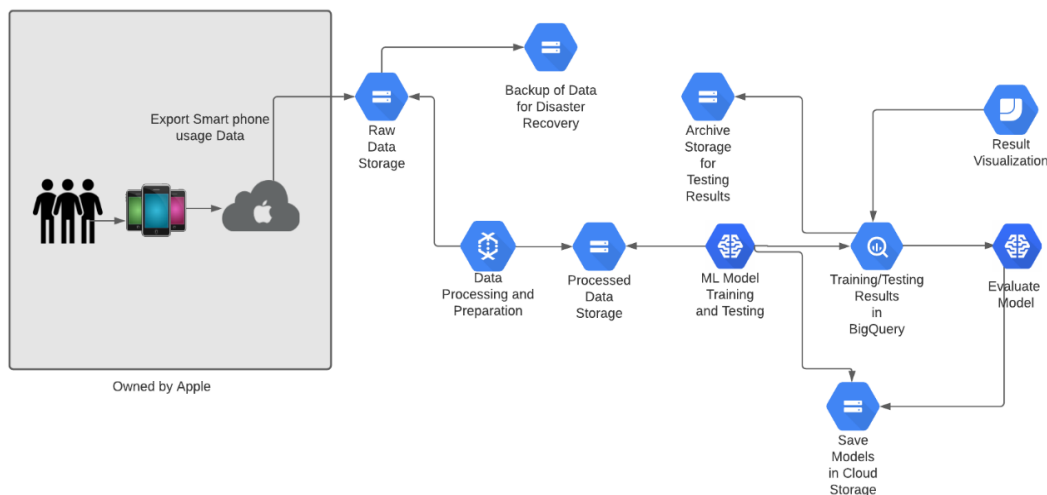
The Google cloud machine learning platform will be used to divide the dataset in train and test sets using Jupyter notebook platform. The train and test sets will be stored on Google bigquery. The prediction results on train dataset will also be performed using Google cloud machine learning platform for which we will store results on bigquery.

The different machine learning models that are applied on the data are stored on Google cloud storage. Finally, the evaluation part includes hyper optimization of parameters and k fold cross validation for which output is stored on Google bigquery.

The data studio will have access to the prediction results stored on bigquery to visualize the results. The prediction results will also be stored in google archive storage for long term access.

Figure 15

Architecture Diagram for Data Pipeline and Machine Learning



Note. The Figure 15 shows the architecture diagram for the project.

3.2 Data Collection

The source of the dataset is the historical data collected by Apple over six months. We referred screen time data.xlsx by Bailey (2019) to prepare sample raw dataset shown in Figure 16. The data is collected for 500 users and the data size will be 25 GB. This dataset will be sufficient for smartphone addiction analysis using machine learning. The raw data collected will

be provided in the form of csv file and the file will be placed on Google cloud storage for our access. The raw data file will have 504 rows and 15 columns. Figure 16 shows sample from raw dataset with all the fields present in the dataset.

Figure 16

Sample from Raw Dataset Showing Parameters

	AppID	DeviceID	UserID	First_Name	Last_Name	Gender	Age	From Date	To Date	Social Networking	Entertainment	Gaming	Productivity	Total Screen Time	Addicted/Not-addictive
1	AP001001	DV001001	UI001001	John	Anderson	M	25	5/17/20	10/16/20	450	160	180	90	880	1
2	AP001002	DV001002	UI001002	William	Davis	M	21	5/17/20	10/16/20	300	720	180	0	1,200	1
3	AP001003	DV001003	UI001003	James	Miller	M	28	5/17/20	10/16/20	300	260	150	85	795	1
4	AP001004	DV001004	UI001004	Charles	Smith	M	22	5/17/20	10/16/20	500	815	180	0	1,495	1
5	AP001005	DV001005	UI001005	George	Johnson	M	40	5/17/20	10/16/20	100	258	0	500	858	0
6	AP001006	DV001006	UI001006	Frank	Williams	M	50	5/17/20	10/16/20	129	369	0	120	618	1
7	AP001007	DV001007	UI001007	Joseph	Brown	M	60	5/17/20	10/16/20	149	200	0	122	471	1
8	AP001008	DV001008	UI001008	Thomas	Jones	M	61	5/17/20	10/16/20	50	75	0	200	325	0
9	AP001009	DV001009	UI001009	Henry	Garcia	M	45	5/17/20	10/16/20	350	300	0	145	795	1
10	AP001010	DV001010	UI001010	Robert	Lopez	M	30	5/17/20	10/16/20	200	250	0	99	549	1
11	AP001011	DV001011	UI001011	Edward	Anderson	M	35	5/17/20	10/16/20	180	200	0	98	478	1
12	AP001012	DV001012	UI001012	Mary	Davis	F	25	5/17/20	10/16/20	500	154	0	55	709	1
13	AP001013	DV001013	UI001013	Anna	Miller	F	21	5/17/20	10/16/20	450	350	0	0	800	1
14	AP001014	DV001014	UI001014	Emma	Smith	F	28	5/17/20	10/16/20	600	165	0	66	831	1
15	AP001015	DV001015	UI001015	Elizabeth	Johnson	F	22	5/17/20	10/16/20	300	650	190	0	1,140	1
16	AP001016	DV001016	UI001016	Minnie	Williams	F	40	5/17/20	10/16/20	389	366	0	190	945	1
17	AP001017	DV001017	UI001017	Margaret	Brown	F	50	5/17/20	10/16/20	289	269	0	189	747	1
18	AP001018	DV001018	UI001018	Ida	Jones	F	60	5/17/20	10/16/20	196	300	0	154	650	1

Note. Figure 16 shows raw dataset with various parameters.

The Figure 16 shows the sample of raw dataset with various parameters - app id, device id, user id, first name, last name, gender, age, from date, to date, social networking, entertainment, gaming, productivity, total screen time, and addictive/not-addictive.

Below are the fields which are shown in Figure 16 and Figure 17.

- AppID, DeviceID - These fields represent the information about the smartphone device. The datatype for these fields is string.
- UserID, First_Name, Last_Name – These fields uniquely identify the user. The datatype for these fields is string.

- Gender, Age – These fields represent the factors that will classify the users. The datatype for Gender is string and data type of age is integer.
- From Date, To Date – These fields represent the start date and end date of the data being collected. The datatype for these fields is date.
- Social Networking, Entertainment, Gaming, Productivity, Total Screen Time – These fields represent the time spent by the users in different categories of application. The unit of measurement is in hours.
- Addicted/Not-addictive – This field contains the labeling of the data which shows if the user is addicted to smartphone or not. The datatype for these fields is integer.

Figure 17

Summary of Raw Data Fields with Datatypes

```

RangeIndex: 504 entries, 0 to 503
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype
---  -
 0   AppID               504 non-null    object
 1   DeviceID            504 non-null    object
 2   UserID              504 non-null    object
 3   First_Name          504 non-null    object
 4   Last_Name           504 non-null    object
 5   Gender              503 non-null    object
 6   Age                 503 non-null    float64
 7   From Date           504 non-null    datetime64[ns]
 8   To Date              504 non-null    datetime64[ns]
 9   Social Networking    504 non-null    int64
10   Entertainment        504 non-null    int64
11   Gaming               504 non-null    int64
12   Productivity         504 non-null    int64
13   TotalScreenTime      504 non-null    int64
14   Addicted/Not-addictive 504 non-null    int64
dtypes: datetime64[ns](2), float64(1), int64(6), object(6)

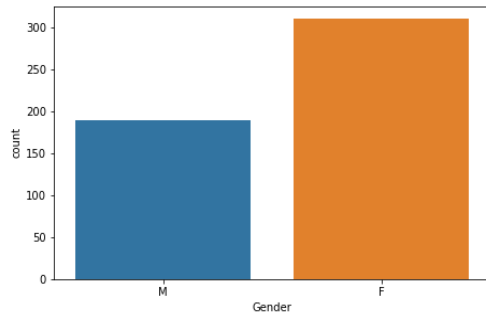
```

Note. The Figure 17 shows the different parameters along with its data type.

The raw data is being analyzed and explored using various graphs created with Python are shown below.

Figure 18

Gender Distribution in Dataset

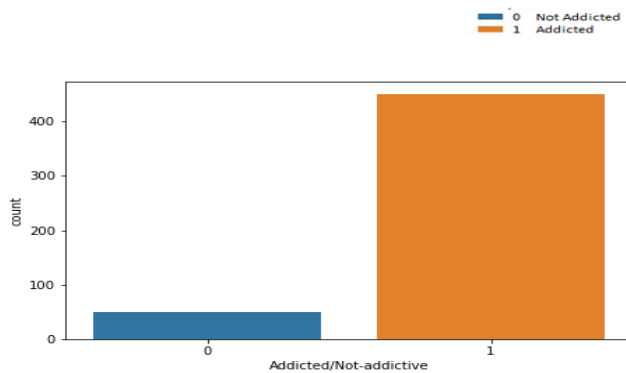


Note. In Figure 18 blue bar represents male and orange bar represents female.

Figure 18 shows the gender distribution for our dataset. It indicates that there are more female smartphone users in our dataset. This exploration will help us to understand the male and female distribution for usage of smartphone.

Figure 19

Addicted and Not Addicted Users by Count

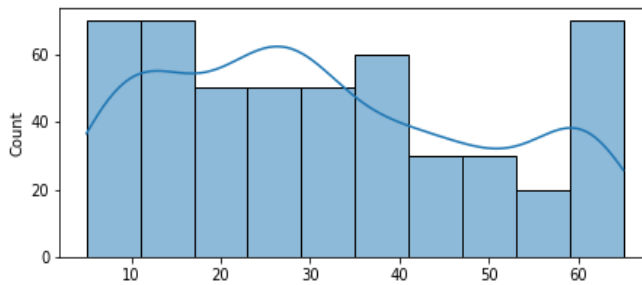


Note. In Figure 19 blue bar shows not addictive users and orange bar represents addictive users.

Figure 19 shows the addicted and not addicted users by count in our dataset. It indicates that most of the users are addicted to smartphone. This exploration will help us to understand the number of addicted and not addicted users of smartphone.

Figure 20

Distribution of Age

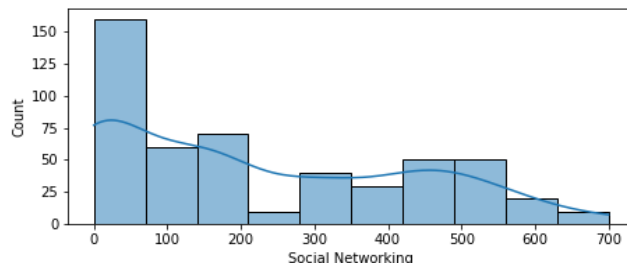


Note. Figure 20 shows distribution for age.

Figure 20 represents the distribution of age for our dataset. The data represents all the ages from five to 65. The distribution for age is mostly uniform. This exploration will give us the idea of age distribution of the dataset. It can also help us to understand maximum and minimum users by the age range.

Figure 21

Distribution of Social Networking

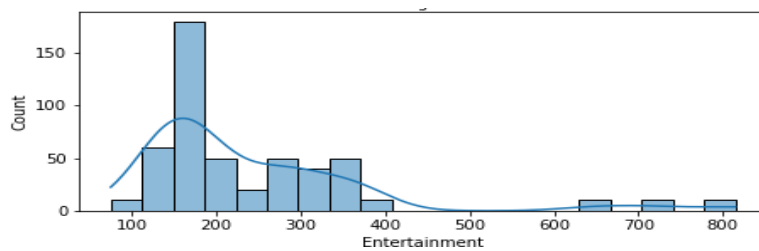


Note. Figure 21 shows social networking distribution for the dataset.

Figure 21 shows the distribution of social networking for our dataset. The distribution for social networking is unimodal and is skewed to right. Social networking is one of the major factor for smartphone addiction prediction. Thus, this exploration will help us how social networking contribute to the smartphone addiction.

Figure 22

Distribution of Entertainment

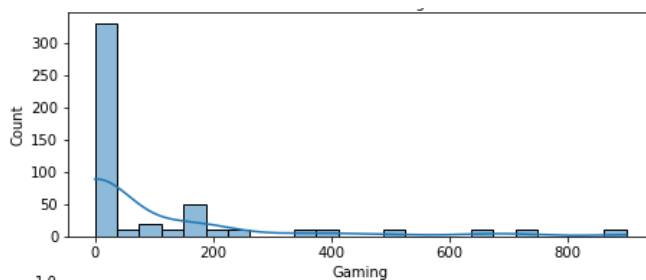


Note. Figure 22 shows entertainment distribution for the dataset.

Figure 22 shows the distribution of entertainment for our dataset. The distribution for entertainment is unimodal and is skewed to right. This is another major factor for smartphone addiction which will help us to understand how entertainment is contributing to the smartphone addiction.

Figure 23

Distribution of Gaming

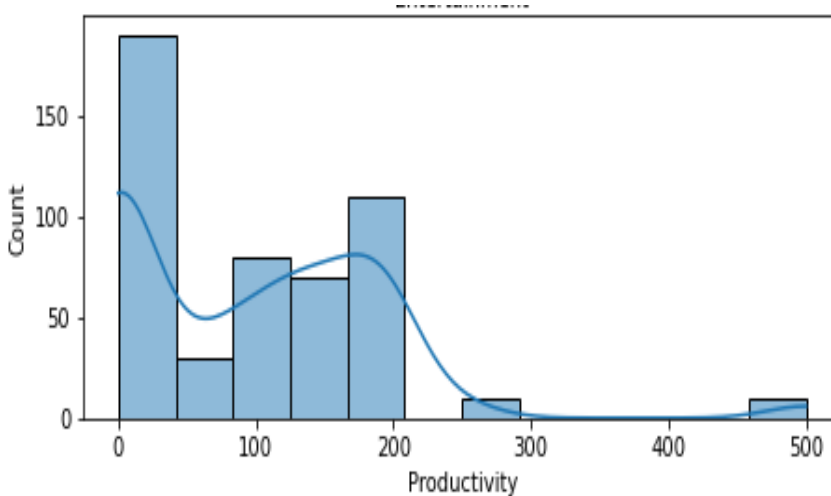


Note. Figure 23 shows gaming distribution for the dataset.

Figure 23 shows the distribution of gaming for our dataset. The distribution for entertainment is exponential.

Figure 24

Distribution of Productivity



Note. Figure 24 shows productivity of users' distribution in dataset.

Figure 24 shows the distribution of productivity for our dataset. The distribution for entertainment is unimodal and is skewed right. Productivity is one of the major factor that needs to be considered to predict the smartphone addiction correctly. Considering, only the total time spent on smartphone without factoring productivity time will lead into wrong prediction results. Therefore, it is important to explore the productivity feature.

3.3 Data Pre-processing

This phase will involve the cleaning of raw dataset. The raw dataset consists of issues like missing values, duplicate values, etc. Such issues need to be addressed before modeling phase. Cleaning of data also results in more accurate results and less processing time. Figure 25 shows the data quality report for Categorical and Continuous variables after cleaning. We have

dropped the fields AppID, DeviceID, UserID, From Date, and To Date. The attributes AppID, DeviceID, and UserID are used for identification and adding them can affect the performance of model. Thus, we will drop these attributes from the categorical features as these features will not be helpful in the modeling. The 500 rows for fields From Date and To Date have same value as 5/17/20 and 10/16/20. Adding these fields in modeling phase will not add any value. Thus, we will also drop From Date and To Date fields from the dataset using Python.

Figure 25

Data Quality Report for Categorical and Continuous variables After Cleaning

	Categorical Feature	total	count	miss%	card	mode	mode freq	mode pct	2nd mode	2nd mode freq	2nd mode pct
0	Gender	500	500	0.0	2	F	310	62.0	M	190	38.0

	Continous Feature	total	count	miss%	card	min	1st qrt	mean	median	3rd qrt	max	std
0	Age	500	500	0.0	26	5	16.0	31.58	29.5	45.0	65	17.881935
1	Social Networking	500	500	0.0	30	0	22.0	225.58	180.0	425.0	700	203.975596
2	Entertainment	500	500	0.0	33	75	150.0	244.40	191.0	300.0	815	146.415710
3	Gaming	500	500	0.0	15	0	0.0	103.60	0.0	150.0	900	201.372888
4	Productivity	500	500	0.0	27	0	0.0	97.82	94.0	180.0	500	97.814250
5	TotalScreenTime	500	500	0.0	47	150	458.0	671.40	611.5	800.0	1500	304.083870
6	Addicted/Not-addictive	500	500	0.0	2	0	1.0	0.90	1.0	1.0	1	0.300000

Note. The Figure 25 shows that there are no missing values after cleaning the data. The figure also shows the other parameters like cardinality, total values, etc. after cleaning the data.

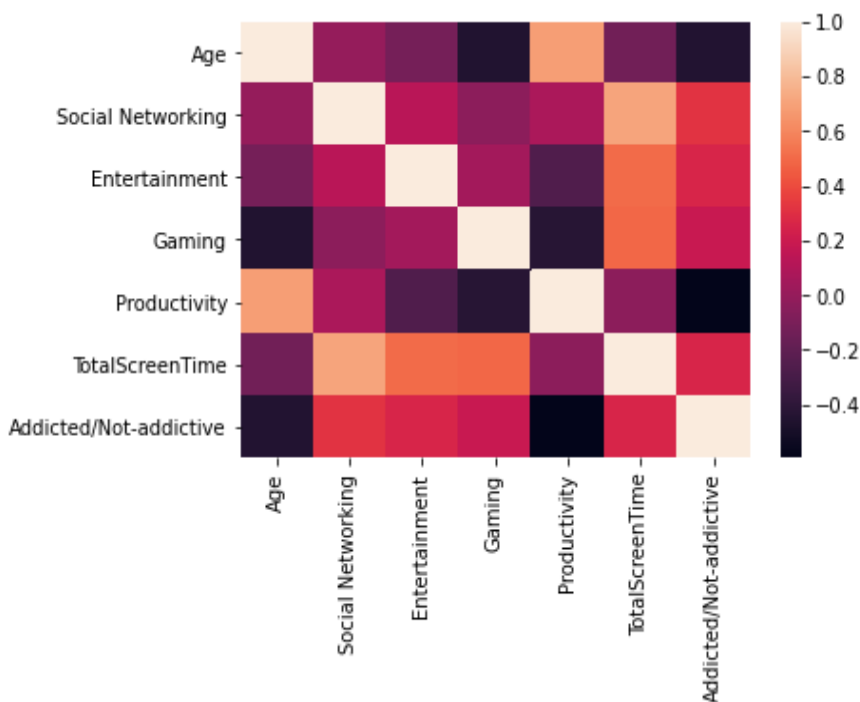
During data pre-processing phase we will make sure of below validations:

- There are no duplicate entries for combination of user id and device id.

- The data is complete with information like age, gender, total screen time, etc.
- The data is consistent. The total screen time should match the sum of all categories - social networking, entertainment, gaming, and productivity.
- The data is uniform. All the data values for social networking, entertainment, gaming, productivity, total screen time will be represented in hours.

Figure 26

Heat Map to Show Correlation Between Different Features



Note. Figure 26 shows the heat map for dataset after data cleaning.

The correlation for different files can be seen in heat map shown in Figure 26. The method used to calculate correlation among different features is Pearson correlation coefficient. The correlation graph helps us to understand the relation among different features in the dataset. The graph also helps us to understand the linear relationship between two features of a given

dataset. The light peach color in the heat map shows the correlation with high values between two features. For example, age and total time screen have strong correlation.

3.4 Data Transformation

After pre-processing step, only gender field is in categorical form with M and F binary values. The machine learning algorithms works on numerical data and therefore gender field needs to be transformed. The data transformation is done in two steps which involves, label encoding and feature scaling. Both the steps would result in numerical values of features which are accepted by machine learning algorithms.

- Label encoding – In this step, the gender values M and F will be converted to 1 and 0 respectively. The code to do label encoding and encoded results are shown in Figure 27.

From the figure we can see that all the fields are in numerical format. The

LabelEncoder() is the predefined function under sklearn library. The gender column has binary values M and F, thus choosing label encoding is good choice here for data transformation.

Figure 27

Gender Values M and F Converted to 1 and 0 Respectively

```
In [30]: 1 le = LabelEncoder()
          2 df['Gender'] = le.fit_transform(df['Gender'].astype(str)) # Male = 1 and female = 0

In [31]: 1 df.head()
```

Out[31]:

	Gender	Age	Social Networking	Entertainment	Gaming	Productivity	TotalScreenTime	Addicted/Not-addictive
0	1	25	450	160	180	90	880	1
1	1	21	300	720	180	0	1200	1
2	1	28	300	260	150	85	795	1
3	1	22	500	815	180	0	1495	1
4	1	40	100	258	0	500	858	0

Note. The Figure 27 shows the gender column transformed into one and zero values.

- Feature Scaling – The wide range of data can increase the processing time. Thus, converting data within a range is very important task before applying machine learning algorithms. In this step, we transform the descriptive features to standardize the data and bring all the descriptive features to the same range. Figure 28 shows code which is used the StandardScaler class from sklearn library. This step will help to make the training process faster.

Figure 28*Feature Scaling Code and Results*

```

In [34]: 1 # Feature scaling
          2
          3 from sklearn.preprocessing import StandardScaler
          4 sc = StandardScaler()
          5 X_train = sc.fit_transform(X_train)
          6 X_test = sc.transform(X_test)

In [45]: 1 print(X_train)

[[-0.79399923 -1.13352805 -1.11138162 ... -0.02249309 -1.03991394
  -1.34974055]
 [ 1.25944706 -0.1385587 -0.11914748 ... -0.51289487 -0.0213991
  -0.4107124 ]
 [ 1.25944706  1.51972356 -0.37216718 ... -0.51289487  0.21522556
  -0.66861449]
 ...
 [-0.79399923  1.51972356 -0.13899216 ... -0.51289487  0.54444248
  -0.07676224]
 [-0.79399923  1.79610394 -0.81371138 ... -0.51289487  0.50329036
  -0.36111584]
 [-0.79399923 -0.91242375  1.86532081 ...  3.01799792 -1.03991394
   1.73370031]]

```

Note. The Figure 28 shows the feature scaling performed using Python and its results.

After performing encoding and feature scaling, the descriptive features and target feature are converted to numerical values. The Figure 28 shows the final transformed results which will be used by machine learning models. The results can be accessed from Google cloud standard bucket for further usage.

3.5 Data Preparation

Figure 29 shows the values for X and y. X represents the descriptive features for the dataset and y represents the target feature for the dataset. The descriptive feature consists of fields gender, age, social networking, entertainment, gaming, productivity, total time screen and target feature consists of addictive/not addictive field. Figure 29 shows the python code where we provide the field index numbers in the form of slicing to perform this step. The code also uses the iloc from pandas library which will do indexing of the features.

Figure 29

Descriptive and Target Features

```
X = df.iloc[:, :-1].values # Gender to Total Screen Time
y = df.iloc[:, -1].values #Addicted/Not-addictive
```

Note. The Figure 29 shows the descriptive features and target feature using Jupyter notebook.

Figure 30 represents the splitting of train and test data. The sklearn library train_test_split will be used to divide the dataset as train and test sets. The train dataset will contain 80 percent of the complete dataset and rest will be test dataset. Figure 31 shows the sample of dataset from X_train and X_test set. Figure 32 shows the sample of dataset from y_train and y_test set.

Figure 30

Train and Test Split

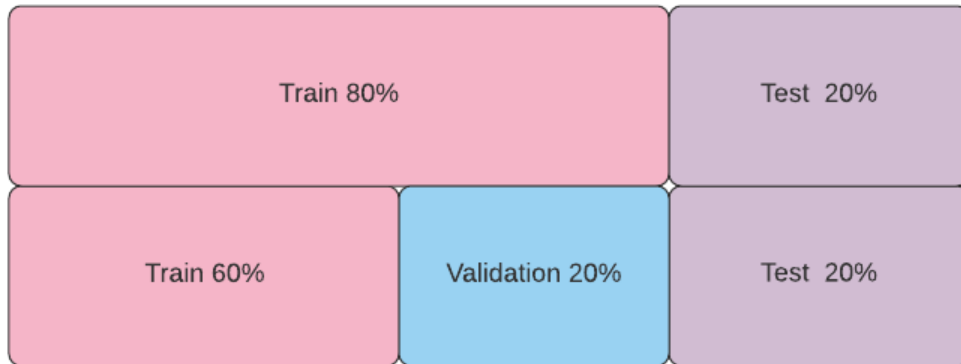
```
# Train and test data
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

Note. The train and test split using Python.

Figure 33

Split of Data – Train Set, Test Set, and Validation Set



Note. The Figure 33 shows data splits as train set, test set, and validation set.

The Figure 33 shows the overall data preparation where we divide the data into 80 percent of train set and 20 percent of test set. We further divide train set into 20 percent of validation set and 60 percent of train set. In modelling phase this validation set will be used to perform the k cross validation for model validation. The hyper optimization of parameters will also be performed before applying k cross validation. The validation set will help us to understand the performances by different models. It will also help us to choose the best model for smartphone addiction.

3.6 Data Statistics

The raw data is collected, pre-processed, and transformed in various stages. The raw data contains fields like AppID, DeviceID, UserID, gender, from date, to date, social networking, entertainment, gaming, productivity, total screen time, and addicted/not addicted. Figure 34 shows the data quality report for categorical and continuous variables on raw data which shows the statistics performed on the raw dataset. The missing values for age and gender fields are

below two percent and thus we have dropped rows with missing values in cleaning part of our dataset. As the missing values are less dropping those rows would not affect the prediction results. Due to privacy concerns, we have dropped the users' first name and last name before checking data statistics.

Figure 34

Data Quality Report for Categorical and Continuous variables Before Cleaning

	Categorical Feature	total	count	miss%	card	mode	mode freq	mode pct	2nd mode	2nd mode freq	2nd mode pct
0	AppID	504	54	89.285714	54	NaN	450	89.285714	AP001050	2	0.396825
1	DeviceID	504	504	0.000000	501	DV001050	2	0.396825	DV001052	2	0.396825
2	UserID	504	504	0.000000	500	UI001050	2	0.396825	UI001051	2	0.396825
3	Gender	504	503	0.198413	3	F	313	62.103175	M	190	37.698413

	Continous Feature	total	count	miss%	card	min	1st qrt	mean	median	3rd qrt	max	std
0	Age	504	503	0.198413	27	5.0	16.0	31.652087	30.0	45.0	65.0	17.874653
1	Social Networking	504	504	0.000000	31	0.0	22.0	223.956349	180.0	425.0	700.0	203.976486
2	Entertainment	504	504	0.000000	34	5.0	150.0	243.214286	182.0	300.0	815.0	146.507983
3	Gaming	504	504	0.000000	15	0.0	0.0	102.777778	0.0	150.0	900.0	200.782750
4	Productivity	504	504	0.000000	28	0.0	0.0	98.226190	98.0	180.0	500.0	97.815953
5	TotalScreenTime	504	504	0.000000	48	11.0	458.0	668.174603	605.0	800.0	1500.0	305.293681
6	Addicted/Not-addictive	504	504	0.000000	2	0.0	1.0	0.894841	1.0	1.0	1.0	0.306758

Note. The Figure 34 shows the missing values for fields gender and age.

Under pre-processing step, we dropped the fields AppID, DeviceID, UserID, from date, to date as these fields would not be helpful for our model. We made sure that the data is consistent, complete, uniform, and valid in the pre-processing step. This can also be seen in Figure 25 which explains the data quality report for categorical and continuous variables after cleaning.

Under transform step we are left with fields -gender, age, social networking, entertainment, gaming, productivity, total screen time, and addicted/not addicted. Using label

encoder, the gender column values are encoded as binary values 0 and 1. Later, we use standard scalar class to standardize the data for all the fields - gender, age, social networking, entertainment, gaming, productivity, total screen time, and addicted/not addicted. Figure 35 shows the statistics performed on the transformed dataset.

Figure 35

Statistics After Transforming the Dataset

	Gender	Age	Social Networking	Entertainment	Gaming	Productivity	TotalScreenTime	Addicted/Not-addictive
count	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.0000
mean	0.380000	31.580000	225.580000	244.400000	103.600000	97.820000	671.400000	0.9000
std	0.485873	17.899844	204.179878	146.562345	201.574563	97.912212	304.388411	0.3003
min	0.000000	5.000000	0.000000	75.000000	0.000000	0.000000	150.000000	0.0000
25%	0.000000	16.000000	22.000000	150.000000	0.000000	0.000000	458.000000	1.0000
50%	0.000000	29.500000	180.000000	191.000000	0.000000	94.000000	611.500000	1.0000
75%	1.000000	45.000000	425.000000	300.000000	150.000000	180.000000	800.000000	1.0000
max	1.000000	65.000000	700.000000	815.000000	900.000000	500.000000	1500.000000	1.0000

Note. Figure 35 shows the statistics like count, mean, standard deviation, etc. performed on transformed data.

Chapter 4 - Model Development

4.1 Model Proposals

The two models which will be used for prediction of smartphone addiction are logistic regression and k nearest neighbor (KNN).

- Logistic Regression

Figure 36

Algorithm for Logistic Regression

```

Step1: Function grad (predictor_attributes, target_attribute, weights)
    {
        Calculate gradient_descent;
        Return weights + learning_rate * gradient_descent;
    }
Step2: Normalize the dataset;
Step3: Repeat
    {
        Weights = grad (params);
        Update weights;
    } until convergence
Step4: z = dot product of predictor variables and updated weights;
Step5: prediction_limit = sigmoid function (z);
Step6: Predict the target class
  
```

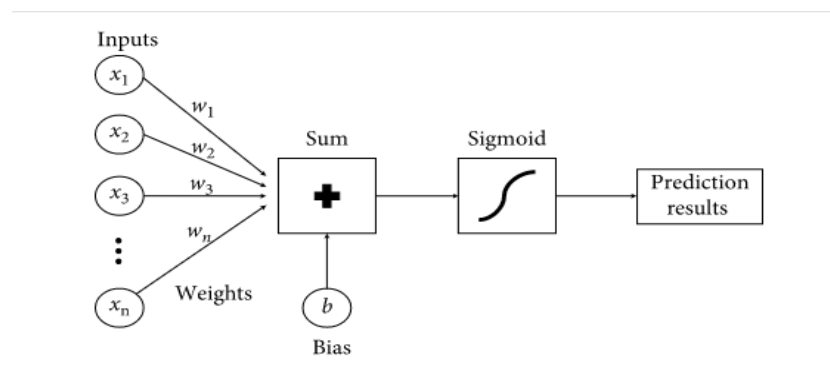
Note. Figure 36 is logistic regression algorithm. Reprinted from *An Intelligent and Energy Efficient Wireless Body Area Network to Control Coronavirus Outbreak*, by Bilandi, N., Verma, H. K., & Dhir, R. (2021). Copyright 2021 by Arabian Journal for Science and Engineering.

Figure 36 shows the algorithm for logistic regression model. The algorithm shows various steps which can be explained as below:

- Step one – A function is created which takes values like x values which is independent variable, y values which is dependent variable, and weights. This function will update the weights to the closest possible optimal values by using different gradients.
- Step two - This step is to normalize the dataset. The method used for normalization is min-max normalization.
- Step three – This step will utilize the function mentioned in step one and keep updating the weights.
- Step four – This step will compute the dot product of x and updated weights from step three.
- Step five – Sigmoid function is applied on the results of step four.
- Step six – This step will predict the target class after applying sigmoid function.

Figure 37

Flow Chart for Logistic Regression Model



Note. The Figure 37 is logistic regression model flow chart. Reprinted from *A comparative analysis of machine learning algorithms to predict Alzheimer's disease* by Bari Antor, M., Jamil, A. H. M. S., Mamtaz, M., Monirujjaman Khan, M., Aljahdali, S., Kaur, M., Singh, P., & Masud, M. (2021). Copyright 2021 by Journal of Healthcare Engineering.

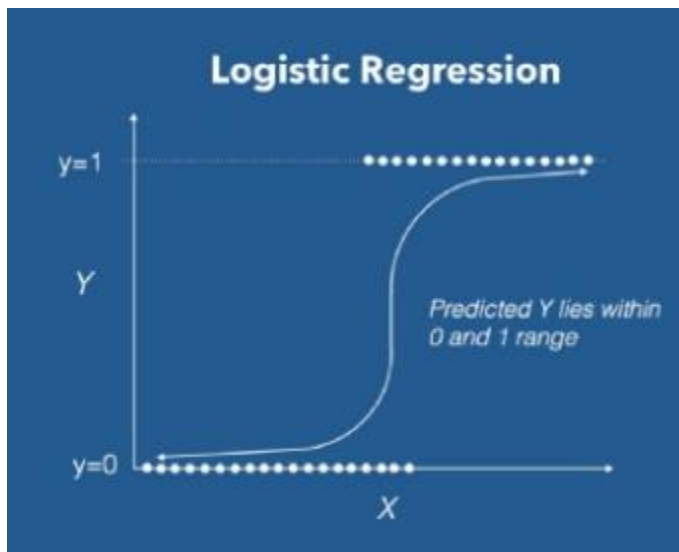
There are various descriptive features which is provided as - age, social networking, entertainment, gaming, productivity, and total screen time. This can be seen as x_1, x_2, x_3 in Figure 37 and each of them are assigned with a weight. For sum shown in Figure 36, we will do the summation of dot product of x and weight.

$$\text{Sum} = x_1 \cdot w_1 + x_2 \cdot w_2 + \dots + b$$

Here b is the variable that represents the leaning of the classifier. To covert the sum into probability the sigmoid function will be applied on the sum. The probability is always between zero and one. The probability above 0.5 will be categorized as addictive and the probability with below 0.5 will be categorized as not addictive. The distribution of sigmoid function can be seen in Figure 38.

Figure 38

Sigmoid Function



Note. Figure 38 is logistic regression sigmoid function. Reprinted from *Introduction to logistic regression* by Pant, A. (2021). Copyright 2021 by Pant, A.

- k nearest neighbor (KNN)

Figure 39

Algorithm for KNN Model

Input : Q , a set query points and \mathcal{R} , a set of reference point;
Output: A list of k reference points for each query point;

```

1 foreach query point  $q \in Q$  do
2   compute distances between  $q$  and all  $r \in \mathcal{R}$ ;
3   sort the computed distances;
4   select  $k$ -nearest reference points corresponding to  $k$  smallest distances;

```

Note. Figure 39 is algorithm for KNN model. Reprinted from *A software tool for fast and scalable knn computation using GPUs* by Arefin, A. S., Riveros, C., Berretta, R., & Moscato, P. (2012). Copyright 2012 by PLoS ONE.

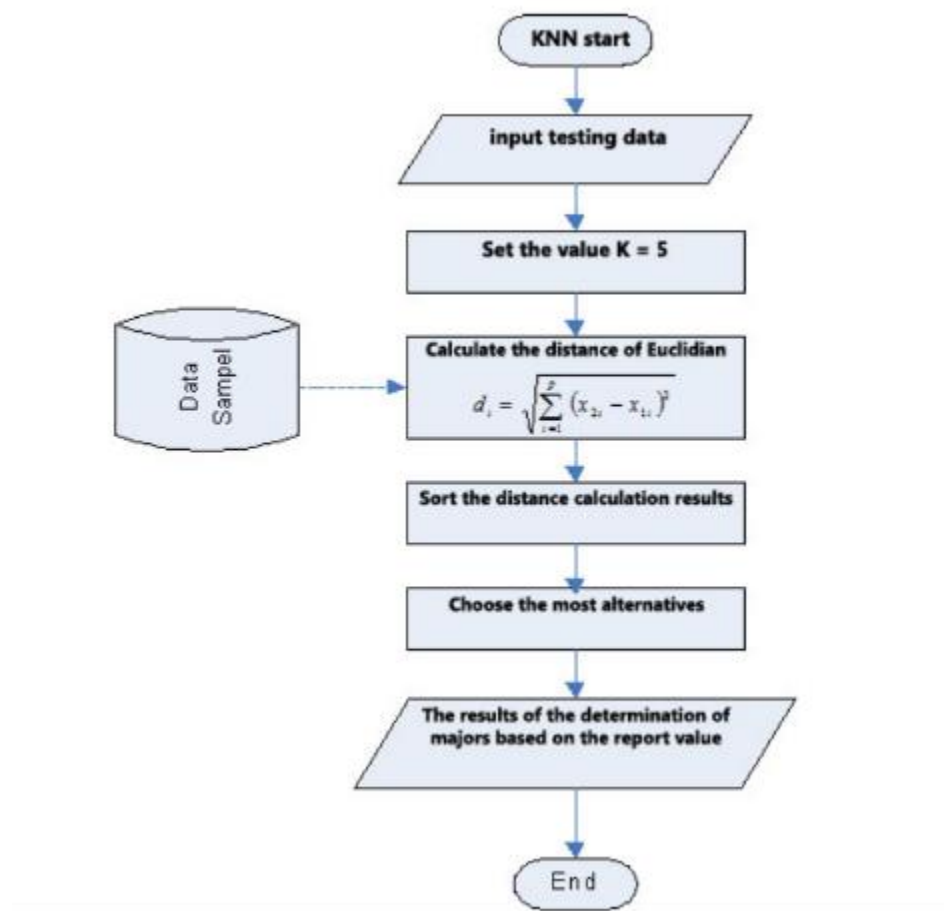
Figure 39 shows the algorithm for KNN model. This algorithm takes input as query Q point for which we want to do the prediction and \mathcal{R} is the reference point in the dataset. r represents the different datapoints within the dataset. This algorithm produces k number of nearest points for which the prediction must be done.

- Step one - Create a loop to include step two to step four for every prediction point q within prediction points Q .
- Step two - Compute the Euclidean distances from points q to r .
- Step three - Arrange all the Euclidian distances in ascending order.
- Step four - Pick produces k number of nearest points for which the prediction must be done.

The above step four will identify each query point in the dataset whether it should belong to classify zero or classify one. If it is a classifier one, then the user is addictive and if the classifier is zero it is not addictive.

Figure 40

Flow Chart for KNN Model



Note. The Figure 40 demonstrates the KNN model flow chart. Reprinted from *Optimization of K Value at the K-NN algorithm in clustering using the expectation maximization algorithm* by Lubis, Z., Sihombing, P., & Mawengkang, H. (2020). Copyright 2020 by IOP Conference Series: Materials Science and Engineering.

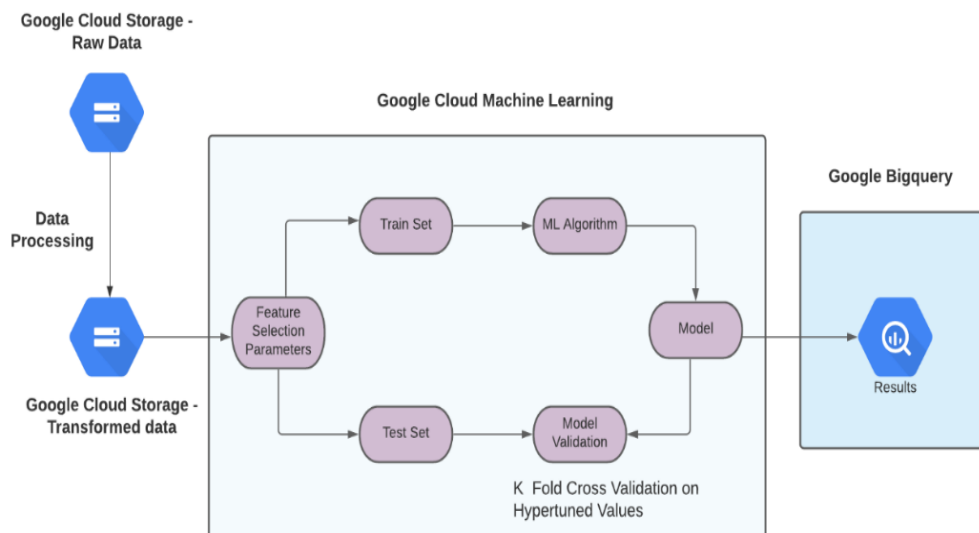
Figure 40 shows the flow chart for KNN where we input the test set and choose the k value as 17. We calculate the Euclidian distance of the new data points from each dataset point by the formula shown in Figure 40. Later, we sort the datapoints from closest to farthest according to the distance. The label which has counted the highest weight will be prediction. In output the model should be given as addictive or not addictive user as result.

4.2 Model Supports

The platform used for machine learning is Google cloud platform. Figure 41 shows the architecture explaining each step for model development.

Figure 41

Architecture for Machine Learning using Google Cloud Platform



Note. The Figure 41 shows the architecture for platform used for machine learning.

The Figure 41 demonstrates the architecture for model development. This involves three major cloud services.

The raw data is stored in a standard Google cloud storage, and it is cleaned and transformed and then transferred into another standard Google cloud storage which will be used by the model. The transformed data have the selected features as age, social networking, entertainment, gaming, productivity, total screen time and target as addicted/not-addictive. The data values are also normalized using min max normalization technique.

In Google cloud machine learning service (AI Platform), we will use Jupyter notebook to separate the data into two sets as train set and test set. Figure 41 shows the Machine learning (ML) algorithm which uses the train data to look patterns into the train data. This pattern learning aims to form the model that could provide predictions on the test set.

Model validation shows the validation performed on logistic regression and KNN model. The method used for validation is k cross validation which is performed on best hyper tuned parameters. Finally, we store the results on Google bigquery.

4.3 Model Comparison and Justification

For building the logistic regression model we used LogisticRegression class from sklearn library. Figure 42 shows the object created for LogisticRegression class as classifier1. There are various parameters which can be given to the model. The two parameters which we will use are penalty and C. The penalty uses two techniques of regularization as L1 and L2 which can help to reduce overfitting. The other parameter C can help us to control adjust penalty strength. Under hyper optimization in later phases, we will use various parameter value combinations and will try to find out the combination with highest accuracy score.

Figure 42

Logistic Model Using sklearn Library

```

1 from sklearn.linear_model import LogisticRegression
2
3 #build
4 classifier1 = LogisticRegression(random_state = 0, penalty='l2', C=1)
5
6 #train
7 classifier1.fit(X_train, y_train)
8
9 #predict
10 y_pred_logistic = classifier1.predict(X_test)

```

Note. Figure shows the logistic regression model details.

Figure 43

Logistic Regression Strengths and Limitations

Strengths	Limitations
<ul style="list-style-type: none"> • Easy to implement, easy to interpret the results, and efficient to train • It can easily be extended to multinomial regression and uses probabilistic approach • It provides the measure of prediction in the form of coefficient, and it also provides the direction (positive or negative coefficients) 	<ul style="list-style-type: none"> • It may lead to overfitting if the data is lower than the number of features • It assumes that there is linearity between dependent and independent variables • This model works only on linearly separable data and will not work on nonlinear data

Note. The Figure 43 shows the strengths and limitations of linear regression model. Reprinted from *Advantages and Disadvantages of Logistic Regression* by GeeksforGeeks (2021).

Copyright 2021 by GeeksforGeeks.

Figure 43 shows various strengths and limitations for logistic regression. The data used for this project is linear as the smartphone addiction depends on various factors like age, social networking, entertainment, gaming, productivity, and total screen time. Thus, linear model should be one of the best models for this project.

Limitations in Figure 43 shows issues the issue of overfitting due to dataset size. However, the dataset used is large enough to provide faster results. Logistic regression model is a widely used model in the industry for similar datasets that is used in this project. Thus, above reasons justify choosing logistic regression as one of the models for smartphone addiction prediction.

For building the KNN model we used KNeighborsClassifier class from sklearn library. In second line of code shown in Figure 44, we can see the object creation for KNeighborsClassifier class as classifier2. There are various parameters which can be given to the model. The two parameters which we will use are n_neighbors for k value and metric for distance calculation. Under hyper optimization in later phases, we will use various parameters combinations and will try to find out the combination with highest accuracy score.

Figure 44

KNN Model

```

1  from sklearn.neighbors import KNeighborsClassifier
2
3  #build
4  classifier2 = KNeighborsClassifier(n_neighbors = 17, metric = 'euclidean')
5
6  #train
7  classifier2.fit(X_train, y_train)
8
9  #predict
10 y_pred = classifier2.predict(X_test)
11

```

Note. The figure shows the KNN model details.

Figure 45*K-Nearest Neighbor Strengths and Limitations*

Strengths	Limitations
<ul style="list-style-type: none"> • It is simple to implement the model and this model makes no assumptions • This model constantly evolves and improves based on the historical data • This model can be used for both regression and classification 	<ul style="list-style-type: none"> • This model slows down as the dataset size increases • The major limitation of model is to choose optimal k value. Often it is hard to choose optimal k value • It will not predict well on the data with a greater number of input variables

Note. The Figure 45 shows the strengths and limitations of k-nearest neighbor's model.

Reprinted from *How does knn algorithm work? What are the advantages and disadvantages of knn?* by MLNERDs. Copyright 2020 by MLNERDs.

Figure 45 shows various strengths and limitations for KNN model. The data is for this project is historical data collected by Apple. Also, this data is labelled data which also makes k-nearest neighbors as next best choice for this project.

It is easy to train data using KNN and it is easy to implement it for problems where the output is binary. In our project, the data is labelled with binary values of addictive and not addictive. It is easy to interpret the results using KNN and provides great results for historical data used in this project.

4.4 Model Evaluation Methods

The project is based on classification techniques logistic regression and KNN. For this project, the model evaluation for logistic regression and KNN is done using accuracy and f1 score.

- **Accuracy** – The accuracy for model prediction can be calculated as the fraction of accurately predicted events and total events. Figure 46 shows the formula for accuracy:

Figure 46

Model Accuracy Formula

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Note. Formula for model accuracy. Reprinted from *Measuring Performance in Classification Models* by Decowski, R. (2021). Copyright 2021 by RStudio-pubs

The calculation of Accuracy will be done in Python. For accuracy we will use the python inbuilt function `accuracy_score`. In this function, we will provide models` respective parameters as `y_test`, `y_pred`. The accuracy uses the parameters of confusion matrix shown in Figure 47.

This can be further explained as below:

- True positive (TP) – This will be the results where the prediction is positive, and the user is addictive to smartphone.
- True negative (TN) – This will be the results where the prediction is negative, and the user is not addictive to smartphone.

- False positive (FP) - This will be the results where the prediction is positive, but the user is not addictive to smartphone.
- False negative (FN) - This will be the results where the prediction is negative, but the user is addictive to smartphone.

Figure 47*Confusion Matrix*

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Note. The Figure 47 demonstrates the parameters of confusion matrix. Reprinted from “Analyzing the Performance of the Classification Models in Machine Learning” by Rajan, S. (2020). Copyright 2020 by Rajan, S.

- **f1 score** – This will be calculated with the harmonic average of recall and precision. The Figure 48 shows the formula for calculations of f1 score.

Figure 48*F1 Score Formula*

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Note. Figure 48 shows f1 score formula. Reprinted from *Accuracy, Precision, Recall or F1?* by Shung, K. P. (2020). Copyright 2020 by Shung, K. P.

The formula in Figure 49 shows that the f1 calculation uses the precision and recall for its results.

Figure 49

Recall and Precision Formula

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

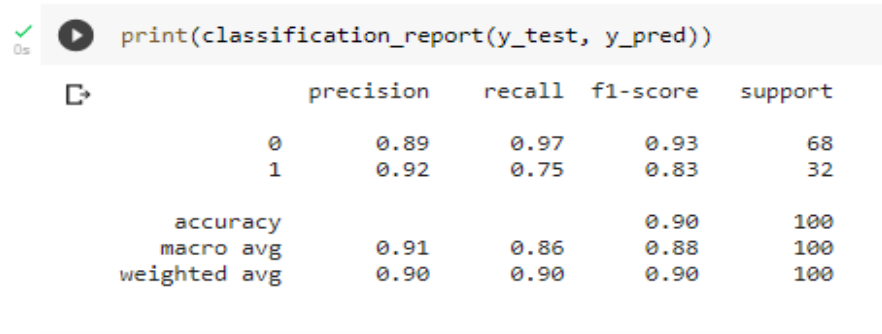
Note. Figure 49 shows formula for recall and precision using parameters of confusion matrix.

Reprinted from *Accuracy, Precision, Recall or F1?* by Shung, K. P. (2020). Copyright 2020 by Shung, K. P.

The calculation of f1 score will be done in Python. For this we will use the python inbuilt function `classification_report`. The function will take the input parameters as `y_test` and `y_pred` for respective models. The classification report also provides us the results of precision and recall. Figure 50 and Figure 51 shows model results for logistic regression and knn before applying k fold cross validation. The figure shows the precision, recall, f1-score, accuracy, and support. However, for our project evaluation, we will only consider f1 score and accuracy score.

Figure 50

Classification Report for Logistic Regression Model Before K Fold Cross Validation



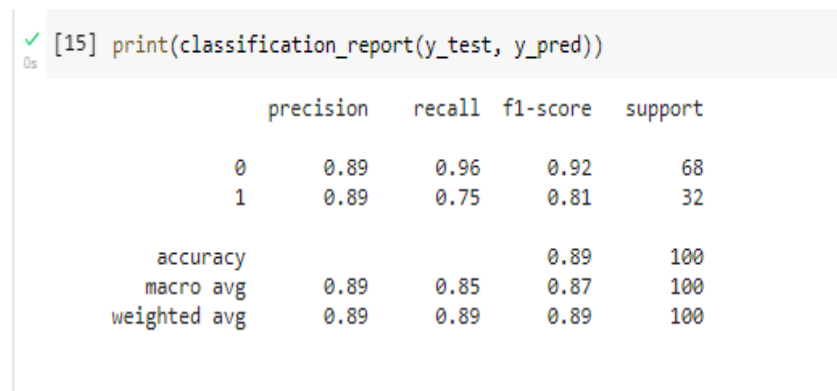
```
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.89	0.97	0.93	68
1	0.92	0.75	0.83	32
accuracy			0.90	100
macro avg	0.91	0.86	0.88	100
weighted avg	0.90	0.90	0.90	100

Note. The Figure 50 shows the accuracy and f1 score calculated for logistic model.

Figure 51

Classification Report for KNN Model Before K Fold Cross Validation



```
[15] print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.89	0.96	0.92	68
1	0.89	0.75	0.81	32
accuracy			0.89	100
macro avg	0.89	0.85	0.87	100
weighted avg	0.89	0.89	0.89	100

Note. The Figure 51 shows the accuracy and f1 score calculated for KNN model.

4.5 Model Validation and Evaluation Results

The model validation and validation are performed in three steps:

- Train the models on training set – This step includes training logistic model and KNN model using train data set.

- Perform hyper tuning and select the best parameter to perform k cross validation – This step will tune hyperparameters for selection of best model parameters. The grid search is used for each model to perform validation. The grid search will use various combinations of the hyperparameters, and its different values chosen by us. The performance results are given for various combinations and thus be able to identify the best among all the combinations. For logistic regression penalty is used as a hyperparameter, which takes values of L1 and L2. The other hyperparameter used is c, which takes values of zero, four, and 10. The results shows that the best hyperparameters for logistic regression are penalty = L2 and c = one.

Similarly, for KNN the hyperparameters used are n_neighbors and metric.

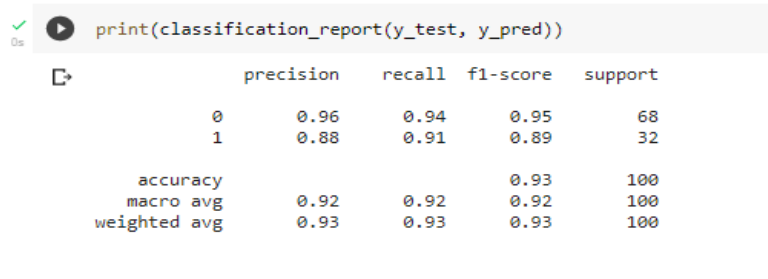
N_neighbors parameter takes values from one to 21 and metric parameter takes values of Manhattan, Minkowski, and Euclidean. The results shows that the best hyperparameters for KNN are n_neighbors = 17 and metric = Euclidean.

Once we have the best hyperparameters, we apply the cross validation with library function GridSearchCV. The cross validation further divides train set into training data and validation data. The GridSearchCV is provided by parameters - estimator = model, scoring = accuracy, which is measure for the performance, cv = 10 which is number of folds used for cross validation. The results will be evaluated using confusion matrix, accuracy score, and f1 score for each model.

- Compare the results obtained from trained data, validation data, and test data - At the end, we compare evaluation results like confusion matrix and f1 score for various models (on train data, validation data, and test data) and come up with the best model for prediction.

Figure 52

Classification Report for KNN Model After K Cross Validation



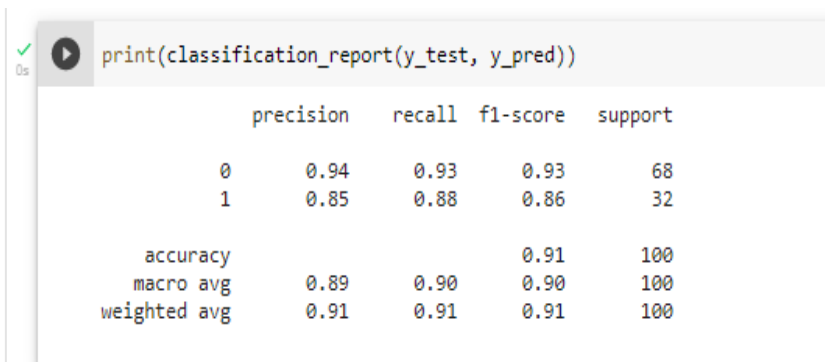
```
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.96	0.94	0.95	68
1	0.88	0.91	0.89	32
accuracy			0.93	100
macro avg	0.92	0.92	0.92	100
weighted avg	0.93	0.93	0.93	100

Note. The Figure 52 evaluation metrics for Logistic regression model after k cross validation.

Figure 53

Classification Report for Logistic Regression Model After K Cross Validation



```
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.94	0.93	0.93	68
1	0.85	0.88	0.86	32
accuracy			0.91	100
macro avg	0.89	0.90	0.90	100
weighted avg	0.91	0.91	0.91	100

Note. The Figure 53 shows evaluation metrics for KNN model after k cross validation.

Figure 52 shows the classification report for KNN with f1 score values as 0.95 and 0.89 for zero and one respectively. Figure 53 shows the classification report for logistic regression model with f1 score values as 0.93 and 0.86 for zero and one respectively. These results are taken after hyper optimization of parameters is performed following k fold cross validation.

Table 5*Results Summary Table for Logistic Regression and KNN Model*

Model	Evaluation metrics	Results	
		Before Tuning	After Tuning and k Cross Validation
Logistic regression	F1 score - 0	0.93	0.95
	F1 score - 1	0.83	0.89
	Accuracy	90	93
KNN	F1 score - 0	0.92	0.93
	F1 score - 1	0.81	0.86
	Accuracy	89	91

Note. The Table 5 demonstrates the summary results from different models.

Table 5 shows that f1 score for both the models is improved after performing k cross validation. The table also shows that the best model for prediction is logistic regression with highest accuracy score of 93 percent. The logistic regression model also outperformed with the f1 score of 0.95 for zero and 0.89 for one.

References

- Advantages and Disadvantages of Logistic Regression*. (n.d.). GeeksforGeeks. Retrieved November 13, 2021 from <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/?ref=lbp>
- Altwater, A. (2021, March 30). *What Is Telemetry? How Telemetry Works, Benefits of Telemetry, Challenges, Tutorial, and More*. Stackify. <https://stackify.com/telemetry-tutorial/>
- Arefin, A. S., Riveros, C., Berretta, R., & Moscato, P. (2012). GPU-FS-kNN: A Software Tool for Fast and Scalable kNN Computation Using GPUs. *PLoS ONE*, 7(8), e44000. <https://doi.org/10.1371/journal.pone.0044000>
- Baby S. G., and Priya R. (2021). Digital Screen Addiction with KNN and -Logistic Regression Classification. *Materials Today: Proceedings*. <https://doi.org/10.1016/j.matpr.2020.11.360>
- Bailey, T. L. (2019, June 3). *Screen Time Data*. data.world. Retrieved October 29, 2021, from <https://data.world/taylynners04/screen-time-data/workspace/file?filename=Screen+Time+Data.xlsx>
- Bari Antor, M., Jamil, A. H. M. S., Mamtaz, M., Monirujjaman Khan, M., Aljahdali, S., Kaur, M., Singh, P., & Masud, M. (2021). A Comparative Analysis of Machine Learning Algorithms to Predict Alzheimer's Disease. *Journal of Healthcare Engineering*, 2021, 1–12. <https://doi.org/10.1155/2021/9917919>
- Bilandi, N., Verma, H. K., & Dhir, R. (2021). An Intelligent and Energy-Efficient Wireless Body Area Network to Control Coronavirus Outbreak. *Arabian Journal for Science and Engineering*, 46(9), 8203–8222. <https://doi.org/10.1007/s13369-021-05411-2>

Chaudhury, P., & Kumar Tripathy, H. (2018). A Study on impact of smartphone addiction on academic performance. *International Journal of Engineering & Technology*, 7(2.6), 50.

<https://doi.org/10.14419/ijet.v7i2.6.10066>

Choose your plan. (n.d.). Lucidchart. Retrieved October 8, 2021, from

<https://lucid.app/pricing/lucidchart#/pricing>

Choueiry, G. (2021). *Interpret the Logistic Regression Intercept*. Quantifying Health.

<https://quantifyinghealth.com/interpret-logistic-regression-intercept/>

Decowski, R. (2021). *Measuring Performance in Classification Models*. Rstudio-Pubs.

<http://rstudio-pubs->

static.s3.amazonaws.com/370944_96c386c03ac54ef3bec4535d49e92890.html

De-Sola Gutiérrez, J., Rodríguez de Fonseca, F., & Rubio, G. *Cell-phone addiction: A review*.

Front. Psychiatry, 7, 175. <https://doi.org/10.3389/fpsy.2016.00175>

GeeksforGeeks. (2020c, September 2). *Advantages and Disadvantages of Logistic Regression*.

<https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/?ref=lbp>

Google Cloud Platform Pricing Calculator. (2021, October 7). Google Cloud. Retrieved October 8, 2021, from <https://cloud.google.com/products/calculator>

How does KNN algorithm work? What are the advantages and disadvantages of KNN? (n.d.).

Machine Learning Interviews. Retrieved November 13, 2021 from

<https://machinelearninginterview.com/topics/machine-learning/how-does-knn-algorithm-work-what-are-the-advantages-and-disadvantages-of-knn/>

- Lee, J., & Kim, W. (2021). Prediction of Problematic Smartphone Use: A Machine Learning Approach. *International Journal of Environmental Research and Public Health*, 18(12), 6458. <https://doi.org/10.3390/ijerph18126458>
- Lee M, Han M, Pak J. (2018). Analysis of Behavioral Characteristics of Smartphone Addiction Using Data Mining. *Applied Sciences*, 8(7):1191. <https://doi.org/10.3390/app8071191>
- Lubis, Z., Sihombing, P., & Mawengkang, H. (2020). Optimization of K Value at the K-NN algorithm in clustering using the expectation maximization algorithm. *IOP Conference Series: Materials Science and Engineering*, 725(1), 012133. <https://doi.org/10.1088/1757-899x/725/1/012133>
- MLNerds. (2020, October 26). *How does KNN algorithm work? What are the advantages and disadvantages of KNN?* Machine Learning Interviews. <https://machinelearninginterview.com/topics/machine-learning/how-does-knn-algorithm-work-what-are-the-advantages-and-disadvantages-of-knn/>.
- Mok, J.Y., Choi, S.W., Kim, D.J., Choi, J.S., Lee, J., Ahn, H., Choi, E.J., & Song, W.Y. (2014). [Latent class analysis on the internet and smartphone addiction in college students.](https://doi.org/10.2147/NDT.S59293) *Neuropsychiatric Disease and Treatment*, 10, 817–828. <https://doi.org/10.2147/NDT.S59293>
- Pant, A. (2021, December 7). *Introduction to Logistic Regression - Towards Data Science.* Medium. <https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148>
- Rajan, S. (2020, July 27). *Analyzing the Performance of the Classification Models in Machine Learning.* Medium. <https://towardsdatascience.com/analyzing-the-performance-of-the-classification-models-in-machine-learning-ad8fb962e857>

Shung, K. P. (2020, April 10). *Accuracy, Precision, Recall or F1? - Towards Data Science*.

Medium. <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>

Supercharge your teamwork. (n.d.). Monday.Com. Retrieved October 8, 2021, from

<https://monday.com/pricing>