**Machine learning Report**

**Abstract:**

This project focuses on developing a prediction model for game application ratings in an application store. By analyzing key attributes such as customer rating, customer count, and in-app purchases, the model aims to predict the rating that a game application can achieve over time. The predictive analysis of these attributes assists in identifying the best game genre to be released in a particular area for obtaining a high customer rating. This project can provide valuable insights for game developers and app store owners to improve their offerings and enhance customer satisfaction.

**Research Questions:**

* Applications to be released in order to gain market attention.
* Predicting rank of game appliaction in today’s life.

**GitHub Repository Address:**

<https://github.com/MaheshGundagoni99/Machine-Learning-Final-Project>

**Data Description:**

This project is related to Application rating prediction ; It contains a dataset with

17007 data samples and 16 features:

• Icon URL : It represents the url of the icon which displays while we search in the app store. It is a string datatype.

• Average user rating : It represents the average of all the ratings given by the user.It is a float datatype.

• User rating count : It represents the the count of the users who gave the feedbacks.

It is a float datatype.

• Price : It represents the cost of the application for the user to download the

application. It is a float datatype.

• In app purchases : It represents the cost of other purchases in the application. It is a

float datatype.

• Description : It shows the information about the application. It is a string datatype.

• Developer : It represents the name of the developer who developer the application.

It is a string datatype.

• Age rating : It represents the rating of the age. It is a string datatype.

• Languages : It represents all the languages in which the application is displayed. It

is a string datatype.

• Size : It represents the memory consumed by the application. It is float datatype.

• Primary genres : It represents the category of game to which it belongs to. It is a

string datatype.

• Genres : It represents the sub category of game to which it belong to. It is a string

datatype.

• Original Release Date : It represents the release date of the applications. It is a date

time datatype.

• Current version Release Date : It represents the release date of the latest version of

the application. It is a date time datatype.

**Source of Dataset:**

<https://github.com/sujithkumar05/app-store-data/blob/main/appstore_games.csv>

**Project Plan:**

Project plan consists of two steps

1. Data pre-processing

2. Model construction

3. Optimisation

4. Model Evaluation

Firstly, we are going to apply Pre-Processing techniques that are Data Cleaning,

Data Normalisation, Data Reduction, Data Transformation[1].

• Data Cleaning : Data cleaning in data mining is the process of detecting and

removing corrupt or inaccurate records from a record set, table or database.

Either you can ignore the tuple or fill the missing values manually.

• Data Normalisation : Database normalisation is the process of structuring a

database, usually a relational database, in accordance with a series of so-called

normal forms in order to reduce data redundancy and improve data integrity.

• Data Reduction : Data reduction is the transformation of numerical or

alphabetical digital information derived empirically or experimentally into a

corrected, ordered, and simplified form.

• Data Transformation : Data transformation is the process of converting data

from one format or structure into another format or structure. It is a

fundamental aspect of most data integration and data management tasks such

as data wrangling, data warehousing, data integration and application

integration.

**Algorithm**:

In this project, I have build three models

• **K Nearest Neighbor Classifier** :

The k-nearest neighbors (KNN) algorithm is a simple, supervised

machine learning algorithm that can be used to solve both

classification and regression problems. It's easy to implement

and understand, but has a major drawback of becoming

significantly slows as the size of that data in use grows.

neigh = KNeighborsClassifier(n\_neighbors=50)

neigh.fit(X\_train, y\_train)

• **Support Vector Machine** :

SVM works by mapping data to a high-dimensional feature space

so that data points can be categorized, even when the data are

not otherwise linearly separable. A separator between the

categories is found, then the data are transformed in such a way

that the separator could be drawn as a hyperplane.

from sklearn import svm

sv\_clf = svm.SVC()

sv\_clf.fit(X\_train, y\_train)

• **Logistic Regression Classifier :**

Logistic regression is basically a supervised classification

algorithm. In a classification problem, the target variable(or

output), y, can take only discrete values for a given set of

features(or inputs), X. Contrary to popular belief, logistic

regression is a regression model.

from sklearn.linear\_model import LogisticRegression

log\_clf = LogisticRegression(max\_iter=1000)

log\_clf.fit(X\_train, y\_train)

**Measuring Performance:**

• **K Nearest Neighbor Classifier** :

Accuracy : It is one of the metric for evaluating the model.

accuracy\_score(y\_test, y\_predicted)

0.73220064724919

Here we got 73.2 % accuracy for the model.

• **Support Vector Machine (SVM)** :

Accuracy : It is one of the metric for evaluating the model.

accuracy\_score(y\_test, y\_predicted)

0.7346278317152104

Here we got 73.5 % accuracy for the model.

• **Logistic Regression Classifier** :

Accuracy : It is one of the metric for evaluating the model.

y\_predict = log\_clf.predict(X\_test)

accuracy\_score(y\_test, y\_predicted)

0.7338187702265372

Here we got 73.4 % accuracy for the model.

**Future Enhancement:**

Data that has been collected was totally related game genre, instead in future we can collect all types of genre applications without bias and bring an inference like which application is most popular, and which type of can should be released in up coming days so that it can sustain the market and with stand in it by considering application size, price, genre.

**Conclusion:**

Based on the analysis of user behavior and preferences, it can be inferred that game applications with no or low cost are more popular among users. The majority of users tend to rate the games between 4 to 5, indicating high satisfaction levels. Additionally, the average application size of the games is also a key factor in determining user interest. These insights can be beneficial for game developers and app store owners to optimize their offerings and attract a larger audience.