APPSTORE PREDICTOR
ON PLAY STORE
MACHINE LEARNING PROJECT

SUBMITTED BY:-
MAHAK MAHAWAR

DATE:30 JAN 2024

TABLE OF CONTENTS

ABSTRACT

This project aims to analyse data about various applications available on Google Play Store.Based on data like application category, size, price, number of installs, content

rating, review count , reviews, prediction of how successful an android application will be on the Google Play Store is made. This is achieved by predicting the likely application rating on google play store. For doing this, Linear Regression model, SVM model and Random Forest regression model are used for predicting the rating. Also models are evaluated by comparing the predicted results against the actual results by the use of mean squared error & mean absolute error.

INTRODUCTION

NEED OF THE SYSTEM

For developing a good android application, it is better to be aware of the characteristics that makes an application successful on that platform. This system helps one know how well their application will work on Google Play Store based on features of the application and what improvements can be made to make that application a hit on Playstore platform. It will also help developers in improving existing applications to achieve higher customer satisfaction levels and better reviews and ratings on Play Store.

APPLICATIONS OF PROPOSED SYSTEM

● It can be used to predict rating of an application available on Google Play Store, based on current ratings of other applications.
● It can be used to predict the success of a new application on Google Play Store. One can simply add this new application's details in the testing set and get the Results.

CHALLENGES IN DEVELOPMENT

● The columns 'category' and 'genre' store almost the same data. If two explanatory variables in a model are highly linearly related, it poses a problem called multicollinearity. Together, these columns have nearly the same effect on the final result. So considering them both can affect the result. Therefore, we dropped 'genre' column from the dataset.
● The dataset contained columns like 'Last Updated', 'Current version', 'Android Version', which do not play any part in the app ratings on Play Store. So we dropped these columns by using dataset.drop (labels=[]) function.
● In data preprocessing stage, an error was incurred because of the rows which had NULL values. Thus, we applied 'dataset.dropna()' function to remove the rows which had NULL values in them.
● We encountered an error of approximately 65% while using SVM model. It was so because we were initially performing feature scaling in SVM model. We overcame this error by removing feature scaling and re-applying the model. Without feature scaling, an error of approximately 20% was there.

WORKING OF PROPOSED SYSTEM

Proposed system uses Machine learning algorithms to predict the rating of the application of google play store based on their features. Branch of Machine learning

used here is supervised Learning which needs a human to “supervise” and tell the computer what it should be trained to predict for, or give it the right answer. We feed the computer with training data containing various features, and we also tell it the right answer. Supervised learning can solve two problems- Classification & Regression. For the said problem, regression is used so as to predict application rating. Machine learning problem in supervised learning can be solved in three stages which are - Data Preparation, Training & Testing, & evaluation of used models. For the regression problem, most commonly used regression models are used which are - Linear Regression, SVM Model & Random forest regression model. Linear regression is the simplest of regression model which is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). Support vector machines (SVMs, also support vector networks [1]) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. When used alone, decision trees are prone to overfitting. However, random forests help by correcting the possible overfitting that could occur. Random forests work by using multiple decision trees — using a multitude of different decision trees with different predictions, a random forest combines the results of those individual trees to give the final outcomes.