Slide 1: Hi we are team 7 from SC2 and today we will be presenting on our mini project.

Slide 2: This would be a short overview of the presentation. We will showcase how we prepared the data, followed by a few key data analysis. We will then go through the model creation process and finally end of the presentation with our conclusion and insights

Slide 3: Firstly, what are the motivations for this project? According to Steam, more than 10 000 games are released just last year. With the pandemic, causing everyone to stay at home, gaming is set to be one of the booming industries in this day and age. However, games nowadays easily cost around $70 and players are finding it increasingly difficult to find the ‘right’ game for themselves.

Slide 4: With these issues, we have came up with two main problem statements, How can we tell whether a game is good? And How do we find games most suited for us?

Slide 5: In order to answer these questions, we will focus on our chosen dataset, the Steam dataset. Steam is a video game digital distribution service and storefront by Valve. Its popularity can be seen as there are more than 50 000 games under its name.

Slide 6: The main dataset we will be using will be from Kaggle steam dataset, which is uncleaned and contains 2 CSV files. Another dataset we will be using will be steam user’s dataset, which is in a SQL file and we will also be calling steam’s developer APIs to gather more information about the user.

Slide 7: We will now move on to explain how we cleaned both datasets. The first dataset we cleaned up is the Kaggle Steam Dataset. There are 2 CSV files, steam\_app\_data and steamspy\_data. We merged the relevant columns of both the excel files. We then drop rows of duplicates and fill missing data with blanks or appropriate data. We converted dates to date time format and also fixed headers for the columns. Since some data is in dictionary format, we used literal evaluation to convert them to strings and joined them together with semicolons. We also created rating variable using Wilson Score Interval from yes or no recommendations

Slide 8: This is the final dataset after cleaning up the data.

Slide 9: For the steam user dataset, we used google cloud platform to host the SQL file on their SQL server. We then connected to the database and did SQL queries with conditions to fetch data that we need and exported it to excel file. We then called steam’s developer’s API to get more information on user's games owned and ratings. We then Merged Both dataset together using steam user ID and exported it. This is the final dataset on the bottom left.

Slide 10: To gain insights from our dataset, we will now analyse some key features in our dataset which are shown below.

Slide 11: Firstly, gaming keywords. We made a wordcloud with the various game descriptions to gain a better understanding of the words used for the game descriptions. A bar plot is then used to visualize the frequency of tags on all the games. With this, we can clearly see that most games come with the tag puzzle or 2D games.

Slide 12: For game ratings, we mainly used boxplots, to show the relationship between achievements and multiplayer compatibility against ratings. We discovered that having achievements in the game would result in a higher rating. This would mean that we could use achievements as one of the variables for our models. On the other hand, multiplayer compatibility does not seem to have much impact on ratings.

Slide 13: Moving on the game quality, we used a bar chart to visualize the number of games released per year. The obvious increasing trend shows a clear sign of booming game industry. A line graph is then use to show the median rating of games per year which tell us that there is a declining median quality of games. However, we used a bar graph for games above the rating of 80, which surprisingly shows an increasing trend. Thus, this tell us that even thought median rating has dropped throughout the years, quality games are on the rise.

Slide 14: For game genres, we first used a violin plot for the genre distribution against rating. However, there is little to no correlation between the genres and rating. We then looked further into the count of games per genres in another bar chart and we discovered that Indie, Casual and Adventure have the highest counts.

Slide 15: To find out more about the relationship between the variables, we used a correlation matrix of all the variables except for genres. Some notable variables with correlation are listed here. Like we have seen before multiplayer compatibility has almost no correlation to its rating.

Slide 16: To answer the two questions that we have, we prepared two solutions, a rating predictor, and also game recommendation system.

Slide 17: Firstly, for the rating predictor, we used two models, the regression models and classification models. For regression models and classification models, these are the models and techniques we used. For this presentation, we will look at some of these models and techniques.

Slide 18: Before we start with our models, we will first use feature selection and K-fold cross validation to come up with our variables as well as the training and test set. We used SelectKBest to find out the top variables and choose them for our models. This reduces the computational cost of modeling. We then standardize the data using StandardScaler to account for input values with differing scales. We then split the data into 80:20 train test data set. Applying K-fold Cross Validation partitions would allow us to have a more generalized model.

Slide 19: Next, we will take a look at Gradient boosting. Gradient boosting uses a loss function to optimize the model, a weak learner to make predictions as well as an additive model to add weak learners to minimize the loss function. After training the model, we obtained a R^2 value of 0.2 for the test set. The root mean square error turned out to be 21.3. This model has a higher R^2 value and lower RMSE value shows that it a more accurate model than the other Regression models. This model also has optimized hyperparameters using GridSearchCV.

Slide 20: Moving on to the classification model, we first prepare data for classification by creating a new column, a new class called “is\_good” based on ratings. However unequal distribution of the data would reduce the performance of our models. We will thus use resampling as a way to mitigate the issue. We used the Synthetic Minority Oversampling Technique for oversampling and Edited Nearest Neighbour for undersampling to achieve a more even distribution of our new class.

Slide 21: We will then apply the Random Forest Classifier where the model will fit a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. Using this, we achieved an accuracy of 0.77 on our test set. This is significantly better than our Regression model.

Slide 22: To answer “How do we find games most suited for us?”, we came up with a game recommendation system. Our system will do 2 types of recommendation, firstly, content-based recommendation, which will be using game metadata, such as game plot and related genres. Secondly, Collaborative filtering recommendation, which uses game rating for all users.

Slide 23: For content-based recommendation, we used the steam dataset to make recommendations based on content. We used text data such as short descriptions, genres and developers. We also did further cleaning of data by removing spaces and removing stop words such as "a" and "the". We also used Count Vectorizer to vectorized text data and also calculated the cosine similarity of their favorite game with all other games. The top most similar game will then be recommended.

Slide 24: For Collaborative Filtering Recommendation, we will be using the Steam User Dataset. We used Truncated Singular Value Decomposition on our data. What this does is that it builds a model based on the past behaviour of users. In this way, the model finds an association between the users and the items. The Model is then used to predict the rating for the games in which the user may be interested.

Slide 25: Singular Value Decomposition, decomposes a matrix into constituent arrays of feature vectors corresponding to each row and each column. We’re able to better estimate the ratings of user and the matrix will then represent a generalized view of users' "tastes". visualising our data using t-SNE, we can see that SVD is finding points close to each other within different dimensions and grouping them up

Slide 26: The last step is then to calculate the similarities by using Pearson's correlation coefficient for each game. This way, we can find the targeted game closest to other games. And the system will recommend games with the highest correlation to the user’s “taste”.

Slide 27: Lastly, we will move on to our conclusion where we will share our data-driven insights, our learning outcomes, and the final conclusion for our project.

Slide 28: For our data-driven insights, we noticed that our classification model was significantly more accurate than our regression model. We deduced that it would be easier to predict discrete values as compared to continuous values. We found out that the top models for our data set seems to be Gradient boosted Regression for the regression model and the Random Forest Classifier for classification. However, we also realized that there are many missing factors required to be predicting the rating of games such as the budget for the game. Moreover, the exponential increase in games throughout the years create a possibility of skewed data, explaining the low correlation between variables.

Slide 29: For our learning outcomes, we tried new ways to obtain data such as learning how to set up a Google Cloud SQL server, use SQL queries and call APIs to get JSON objects. For methods to handle data, we used new methods such as SMOTE and SelectKBest to improve accuracy for our modelling. We also used new models for our solutions such as Random Forest and Gradient Boosting.

Slide 30: Finally, the outcome of our project, with the rating predictor, even when we used classification model, the accuracy that we achieved might still not be ideal. This would mean that ratings are more volatile to external factors. However, we believe that this rating predictor can still be a gauge for both gamers and game creators on the quality of a game. As for our recommendation system, we were able to show users which games are more suited for their tasters based on their most played games. Thus, they would be able to easily filter out games for themselves.

Slide 31: Coming to the end of the presentation, we hope that our models can help gamers find the perfect game for themselves. Thank you.