**IML\_HACKATHON\_2021**

**Detailed Report**

**Ehud Dahan, Natan Goldstein, Rachel Asael, Neria Cohen**

**AIM**

Our goal is to develop a model that can predict the box office revenue and average viewer rating of movies before their official release. The prediction will be based on a dataset containing information about 5,000 movies. Revenue will be predicted as an integer, while the viewer rating will be predicted as a float between 0 and 10, with one decimal place.

**DATASET DESCRIPTION**

**Dataset description:** The provided dataset contained around 5,000 movies, each with 22 features. (ID, collection to which the movie belongs (if indeed), budget, genre, link to homepage, original language, original title, overview, number of viewers who ranked the movie, production companies, production countries, release date, runtime, spoken language, the stage of production, tagline, title, keywords, cast, crew.)

**CHALLENGING CHARACTERISTICS OF THE DATASET**

* **Variety of Data Formats**: The dataset contains different formats for various movie characteristics. Some data, like budget, is numeric, while others, like the overview, are long strings. Additionally, some columns contain lists, such as genres, crew, production companies, keywords, cast, and spoken languages.
* **Processing Non-numeric Data**: Non-numeric data had to be processed and transformed into a usable format for modeling.
* **Missing Data**: Many values were missing in the dataset. For example, numerous revenue values were absent, posing a challenge for accurate predictions.
* **Non-useful Variables**: Some features could be directly inferred from others, making them redundant and less useful for the model.

**DATA CLEANING AND PRE-PROCESSING**

* First, we needed to **divide the provided data into training and testing sets**. We did this by sorting all the data according to revenue and then selecting every fourth entry to be in the test set, with the remaining data going into the training set.
* To handle **missing fields**, we decided to fill them using the average of the dataset. This approach provided better results than filling missing values with zeros and helped maintain the balance of the features.
* We handled **nonnumeric data** by converting categorical features into dummy variables. For example, if a movie's genre was categorized as "Action," "Comedy," or "Drama," we created separate binary variables for each genre category. For each movie, the corresponding dummy variable is set to 1 if the movie belongs to that genre and 0 otherwise. This transformation allowed us to include these categorical features in our numerical models, enabling the model to understand and utilize this information effectively.

**CONSIDERATIONS THAT GUIDED OUR DESIGN OF THE LEARNING SYSTEMS AND DESCRIPTION OF THE CHOSEN ALGORITHM**

**Random Forest Option**

Initially, we considered using the Random Forest model due to its ability to handle complex and nonlinear relationships. However, further analysis revealed that our data exhibited predominantly linear relationships between the features and the target variables (revenue and viewer ranking). Given this linearity, we decided that linear regression would be more appropriate.

Random Forest is an ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy and robustness. It does this by randomly sampling subsets of the data and selecting random subsets of features for each tree, which helps in reducing overfitting. However, in our case, using a simpler model like linear regression allowed us to effectively capture the linear trends in our data without the risk of overfitting associated with more complex models.

**Nearest Neighbors Option**:

We also considered using the k-nearest neighbors (k-NN) algorithm to predict revenue and viewer ratings. k-NN predicts a target value based on the average values of the closest movies in the feature space, and could also be used to impute missing data by leveraging local similarity among data points.

However, we rejected k-NN due to potential overfitting if 'k' is too small. Given the linear nature of our data, linear regression was more suitable. For imputing missing data, we chose to use the general average of the available data to maintain robustness and avoid complexity.

**Correlation Matrix Analysis**

Finally, we created a correlation matrix to analyze relationships between variables in our dataset. A correlation matrix shows correlation coefficients between pairs of variables, indicating the strength and direction of linear relationships.

We found that variables such as budget and the number of votes had strong linear correlations with revenue and viewer ranking. This informed our decision to use a linear regression model, which is well-suited for capturing these linear patterns and making accurate predictions based on the identified correlations.

**VARIOUS METHODS WE TRIED AND THEIR RESULTS**

Not all the features in the dataset were necessary for predicting film ranking and revenue. Some features were insignificant, while others overlapped with more important features. To reduce the number of features while maintaining good prediction accuracy, we used several methods:

**Scree Plot (PCA)**:

* **Explanation**: PCA creates new uncorrelated features called principal components by combining original features. A scree plot is a graphical representation used in PCA to determine how many principal components to retain. It shows the variance captured by each principal component. The point where the plot starts to level off indicates the optimal number of components that capture the most significant patterns and variations in the data.
* **Example**: Principal Component 1 (PC1) might combine budget and number of votes, capturing the overall popularity and financial aspect. Principal Component 2 (PC2) might combine runtime and genre, capturing the content and type of the movie.

**Lasso Method**:

* **Explanation**: Lasso regression (Least Absolute Shrinkage and Selection Operator) adds a penalty to the absolute values of the regression coefficients. If a feature has a very low coefficient (meaning it doesn't add much value to the prediction), the penalty makes it more favorable to exclude that feature. Therefore, the coefficient and the feature will be excluded from the model. This helps in selecting only the most important features while reducing the complexity of the model.
* **Example**: If we apply Lasso to our dataset, it might retain features like budget and number of votes as important predictors, while shrinking coefficients of less relevant features like cast size and crew size to zero.

**Correlation Matrix**:

* **Explanation**: We created a correlation matrix to find key features with strong linear relationships to the target variables and to identify redundant features. A correlation matrix shows correlation coefficients between pairs of variables, indicating the strength and direction of their relationships. This helped us see that budget and number of votes were strongly correlated with revenue and ranking, making them important features.

**Common Sense**:

* **Explanation**: We applied domain knowledge to select features we believed were critical for making accurate predictions.

These methods helped us streamline our model by focusing on the most relevant features.

**PREDICTION OF THE GENERAL MODEL ERROR WE EXPECT OUR SYSTEM TO HAVE:**

We expect that the more noisy or incomplete the data we receive, the harder it will be to make accurate predictions, resulting in larger errors.

**IMAGES FROM OUR LEARNING PROCESS**

A graph showing the growth of training

Description automatically generatedA graph showing a plot of design matrix

Description automatically generated![A graph of a number of data

Description automatically generated with medium confidence]()

**Hackathon summary for interviews:**

"As part of a Machine Learning course, I participated in an extra credit hackathon where our team developed a model to predict box office revenue and viewer ratings for movies using a dataset of thousands of movies with tens of features. We addressed challenges such as diverse data formats, nonnumeric data, missing values, and redundant features through data cleaning and preprocessing. We split the data into training and test sets, filled missing values with the dataset's average, and converted categorical data into dummy variables. Although we initially considered using Random Forest and k-nearest neighbors, we opted for linear regression due to the data's linear nature. We used a correlation matrix to identify key features, PCA for dimensionality reduction, and Lasso regression for final feature selection. These steps ensured our model's accuracy. This experience enhanced my skills and fueled my excitement for machine learning."

**סיכום בעברית לראיון**

* במסגרת קורס Machine Learning השתתפתי בהאקתון שבו פיתחתי יחד עם הצוות שלי מודל לחיזוי הכנסות ודירוג של סרטים בהתבסס על מאגר נתונים של אלפי סרטים עם עשרות תכונות.
* עשינו Data Cleaning וPre-Processing כדי להתמודד עם אתגרים כמו פורמטים שונים של נתונים, נתונים לא-מספריים, ערכים חסרים ותכונות מיותרות.
* חילקנו את הנתונים לTraining Set ו-Test Set, מילאנו ערכים חסרים והמרנו Categorical Data ל-Dummy values.
* למרות שבתחילה שקלנו להשתמש ב-Random Forest וב-K-Nearest-neighbors , החלטנו להשתמש ברגרסיה לינארית בגלל האופי הליניארי של הנתונים.
* השתמשנו במטריצת קורלציה לזיהוי תכונות מפתח, בPCA לצמצום הממדים וב-Lasso Regression לבחירת התכונות הסופיות.
* בחוויה זו התנסיתי ביישום התיאוריה של ML והתלהבתי מאוד.

**לקורות חיים:**

• Machine Learning Hackathon: Developed a predictive model in Python to forecast box office revenue and viewer ratings for movies using linear regression. Managed data cleaning and preprocessing, applied PCA for dimensionality reduction, and utilized both a correlation matrix and Lasso regression for feature selection. (ML Hackathon, 2021)