# CSE 158 Assignment 2 Report

Leveraging Machine Learning for Fashion Retail Insights

Thanh Nguyen
CSE - UCSD
thn038@ucsd.edu

Minh Duong
CSE - UCSD
m7duong@ucsd.edu

James Rey Macasa
CSE - UCSD
jmacasa@ucsd.edu

Justin Huang
CSE - UCSD
jnhuang@ucsd.edu

**ABSTRACT**

The Rent the Runway (RTR) dataset reflects the confluence of fashion retail and data analytics, highlighting how data is reshaping e-commerce in the fashion industry. RTR's service, which allows customers to rent designer clothing, captures intricate behavioral data, including customer preferences, sizing, and event types, offering a detailed picture of consumer habits and trends. The frequent interactions, averaging about 80 days per year, provide a rich dataset for analysis and strategic decision-making.

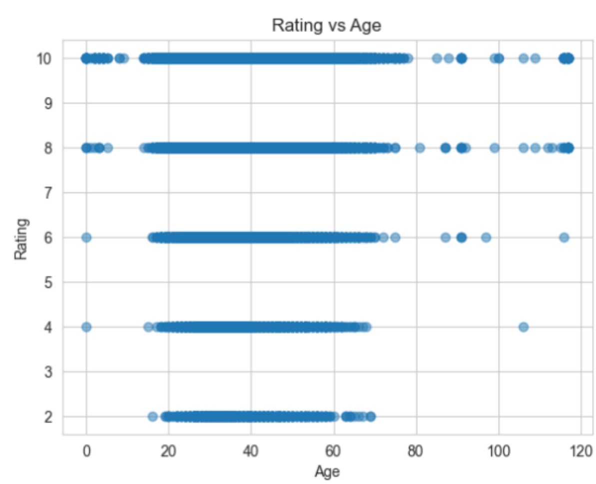
Operational data from RTR's warehouse also plays a crucial role, tracking garment longevity and informing inventory management. This operational insight helps in maintaining product quality and informs fashion designers about the wearability of their designs. Such data is instrumental for RTR in optimizing the user experience through personalized recommendations and bespoke marketing strategies, like promoting the environmental benefits of renting clothes.

Lastly, the RTR dataset serves as a critical tool for fashion brands and designers. It enables them to understand customer preferences better, design products that align with real-world wear patterns, and adapt to the dynamic fashion market. This level of data utilization delineates RTR's innovative edge, illustrating the dataset's significance in driving sustainable, customer-centric business models in the contemporary fashion landscape.

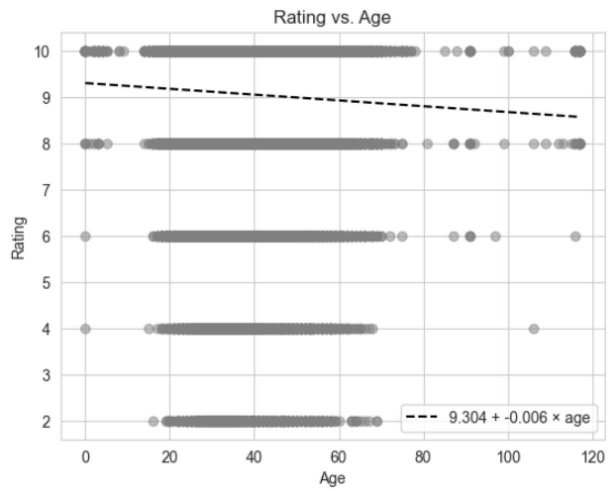
**PERFORM EXPLORATORY ANALYSIS (Task I)**

**1. How does the age of users influence their ratings of clothing items rented?**

The graph depicts the relationship between the age of users and the ratings they give to clothing items, likely from a rental service. The horizontal axis represents the age of the users, while the vertical axis shows the ratings on a scale, presumably from 1 to 10. The data points are spread out across all age groups, indicating that users of all ages are providing ratings. There is a concentration of data points at each whole-number rating value, which is typical in rating data where users prefer to give round-number scores. This distribution suggests that users generally tend to give positive ratings regardless of age, and that age does not appear to be a determining factor in how users rate the clothing items they rent.

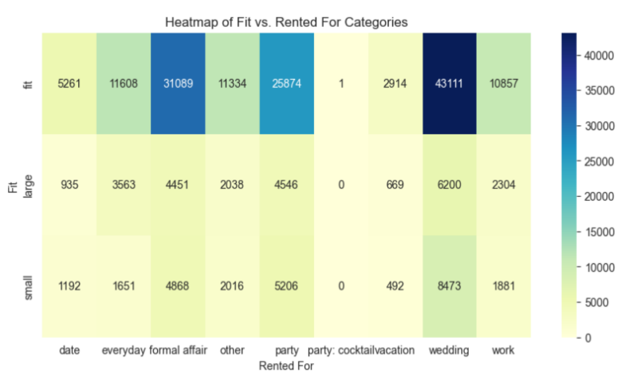



This first graph lacks the regression line but shows a similar distribution of ratings across ages. The data points are densely packed at certain ratings, which could represent common scores given by users. For instance, there are dense horizontal lines of data points at the higher ratings of 8, 9, and 10, indicating that regardless of age, these are common ratings given by the users.



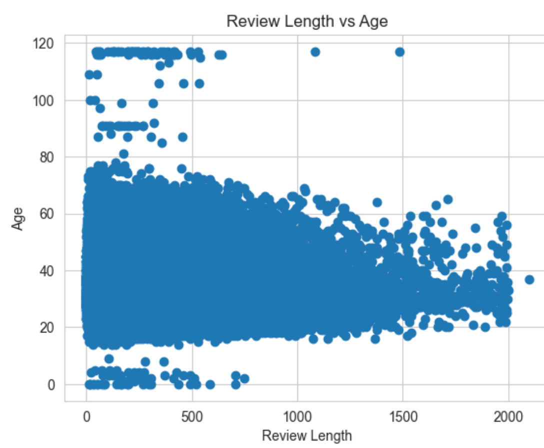

We see a more detailed regression line, with the equation "9.304 + -0.006 × age", suggesting a very slight decrease in rating with increasing age. However, the slope of the line is almost horizontal, and the slight negative trend (-0.006) is not substantial enough to suggest that age is a strong predictor of the rating. This is further evidenced by the high density of points across all rating levels for most age groups, indicating that customers of all ages have a wide range of opinions about the clothing items.

**2. How do customer preferences and satisfaction, as indicated by the fit of rented items for different purposes, correlate with the overall trend in service usage over time?**

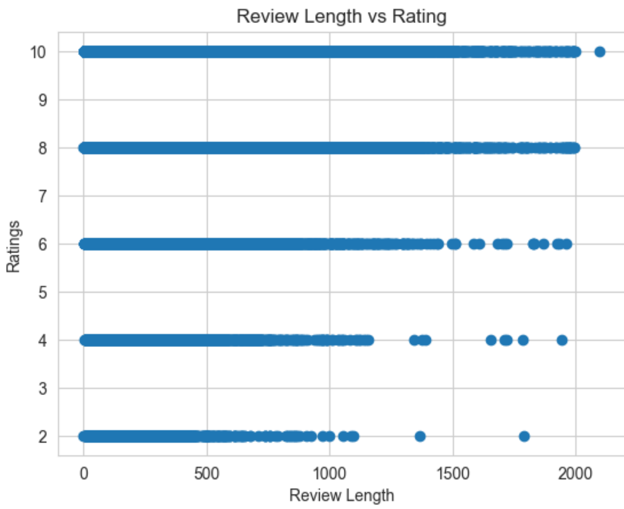



The heatmap and the cross-tabulation table offer a visual and numerical representation of the relationship between the fit of the clothing items and the occasions for which they were rented ("rented for"). The darker colors in the heatmap correspond to higher frequencies, revealing that the majority of items rented for occasions like 'weddings' and 'formal affairs' were rated as a 'fit'. Interestingly, 'everyday' items show a higher frequency of being rated as 'small', possibly indicating sizing issues for non-event clothing. The lack of dark shades for 'large' suggests that users less frequently find items too large for events, which could indicate either more accurate sizing or perhaps the selection of styles that are more forgiving if oversized. Overall, this could suggest that customers find a better fit with items intended for special occasions, which could be due to more careful selection or better sizing information provided for such items.

**3. How does age and review length correlate and how does it affect the overall rating of rented items?**




As can be seen from the above scatter plot, there is a clear relationship between the age of a reviewer and the length of their review which is that most of the bigger reviews tend to be given from people around the ages of 20-40. Another trend which can be seen is that most of the reviews of these items are given by people between the ages of 20-60. From here we can create a scatter plot to see a relationship between the review length and the ratings people give the rented items.




As can be seen from the scatter plot above, bigger reviews tend to give much higher ratings than smaller reviews. Lower rating reviews are more likely to have much smaller reviews. Looking at this trend and the relationship between review length and age, it is likely that people ages 20-40 who give big reviews are also likely to give higher ratings as well.

**PREDICTIVE TASK (Task III)**

On this data set, we plan to do a classification type predictive task. Each datapoint specifies whether a certain rented item fits the reviewer or not, so we plan on predicting whether or not an item will fit for a certain reviewer or not and give them a label of whether it fits or not. We will evaluate our model based on its accuracy and whether we correctly predicted that an item will fit a user. From here, a solid baseline model to use is a Logistic Regression classifier model as we are able to easily incorporate features to help do the predictions as well as it is easily optimizable and can easily be improved upon to get better accuracy and metrics. Starting with the baseline model of Logistic Regression, we plan to use the rating, age, and review length of each datapoint as features to help predict whether a rented item will fit a reviewer. To process the data to get the features we dropped all the unnecessary features of each datapoint and just kept the three features and the fit value so that we just have to deal with only necessary values from the dataset.

We are going to start with logistic regression as our basic model ¶

```
X = df[['rating','age', 'text_length']]
y = df[['fit']]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize the logistic regression model
model = linear_model.LogisticRegression()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
predictions = model.predict(X_test)

# Evaluate the model
logistic_accuracy = accuracy_score(y_test, predictions)
conf_matrix = confusion_matrix(y_test, predictions)
classification_rep = classification_report(y_test, predictions)

print("Accuracy:", logistic_accuracy)
# print("\nConfusion Matrix:\n", conf_matrix)
# print("\nClassification Report:\n", classification_rep)
```

The purpose of our group for the predictive task is to compare the accuracy between the baseline model with other models, so that we can conclude the table to show the rank of the best fit model for this dataset to the worst fit model.

*Logistic Regression*

Our baseline model, serving as a point of comparison for subsequent models. Its strength lies in the simplicity and interpretability of its predictions, which can be crucial when explanations for predictions are required.

*Random Forest Classifier*

Showcased the power of ensemble learning with decision trees, achieving an accuracy competitive with the baseline. This model is known for its ability to handle non-linear data with a mixture of categorical and numerical features without the need for feature scaling.

*Naive Bayes Gaussian*

Performed impressively given its assumption of feature independence, indicating its suitability for this dataset's structure. This model's strength is its speed and performance in high-dimensional datasets, especially with appropriate prior information.

*Naive Bayes Categorical*

Targeted towards categorical data, showing results comparable to other models, despite its simplicity.

This model is often preferred for classification tasks with discrete features and proves to be a good strategy for baseline algorithms due to its speed and efficiency.

*Gradient Boosting Classifier*
Demonstrated the utility of boosting techniques in improving predictions, with accuracy on par with the leading models. This method builds one tree at a time, where each new tree helps to correct errors made by previously trained trees, providing a high level of precision.

*Bagging Classifier*
Used bootstrapping to create an ensemble of decision trees, achieving accuracy similar to Random Forest. The strength of bagging is that it can reduce variance and help to avoid overfitting, which is a common problem in decision tree models.

*Support Vector Machine (SVM)*
Exhibited its classification prowess, particularly with linear kernels, in high-dimensional spaces. SVMs are effective in cases where the number of dimensions is greater than the number of samples, which can be quite common in modern datasets.

*XGBoost*
Boosted the predictive performance further with gradient boosting, known for its efficiency and effectiveness. It is often the go-to model for winning Kaggle competitions due to its performance and speed when dealing with large datasets.

*Linear Discriminant Analysis (LDA)*
Utilized for dimensionality reduction and classification, showcasing its potential in predictive accuracy. LDA is particularly useful when the classes are well-separated and the data is approximately normally distributed.

| | Model | Accuracy |
|---|---|---|
| 1 | Logistic Regression | 0.742917 |
| 2 | Random Forest | 0.744060 |
| 3 | Naive Bayes Gaussian | 0.745696 |
| 4 | Naive Bayes Categorical | 0.742580 |
| 5 | Gradient Boosting Classifier | 0.745955 |
| 6 | Bagging Classifier | 0.744579 |
| 7 | SVM | 0.734322 |
| 8 | XGB boosting | 0.744190 |
| 9 | LDA | 0.745566 |

Based on the summary table provided, it is evident that all the models performed with a relatively high level of accuracy, each falling within a narrow range just above 70%. This close performance suggests that the underlying data is well-structured and can be effectively modeled using a variety of machine learning techniques.

***Issues Encountered:***
Scalability could be an issue, particularly with ensemble models that require training multiple trees, potentially leading to increased computational time. Overfitting is another concern, as more complex models might perform exceedingly well on the training data but fail to generalize to unseen data. To mitigate this, cross-validation will be employed, alongside monitoring the performance on a validation set.Then, because we work on the complex model, which is SVM, it takes us around 6-8 minutes to finish the execution, depending on different machines and how it takes memory to run. This is quite a long time for running because we should expect the outcome to have a short-time running and continue to other models. Lastly, we use some special model, which is called “XGB Boosting”, and Python requires us to install the packages before we can run through the model and calculate the predictive task. Therefore, we add a command line:

```
!pip install xgboost
```

at the beginning of the code so that whoever runs the code can automatically add that package into the Notebook.

***Model Comparison:***

For comparison, Logistic Regression was used due to its simplicity and interpretability. Naive Bayes Gaussian and Categorical models were also considered due to their efficiency and speed, which are beneficial when quick model training is desired. SVM was included due to its effectiveness in high-dimensional spaces, and XGBoost and LDA were chosen for their advanced optimization techniques and ability to find a linear combination of features that best separate the classes, respectively. In general, Gradient Boosting Classifiers, which are ensemble methods, exhibited the strongest performance. Their ability to build multiple models and aggregate their predictions likely helped to improve generalization over the data, reducing the risk of overfitting that might occur with a single complex model.

***Unsuccessful Attempts:***

Attempts that might not have yielded the desired level of accuracy include using models without hyperparameter tuning or employing models that were too simplistic to capture the relationships in the data, leading to underfitting. As we try to tune up the accuracy, it can only be bound around 0.74~0.76 (as shown in the table above).

***Strengths and Weaknesses:***

The ensemble methods' strengths lie in their robustness and ability to model complex data, but they can be computationally expensive and less interpretable. Logistic Regression, while very interpretable and fast to train, might not capture complex relationships as well. Also, it may not be the best fit model to the dataset, as demonstrated by this project. Naive Bayes models are extremely fast and work well with high-dimensional data but make strong assumptions about the independence of features. SVMs are powerful for datasets with clear margin separation but can be less effective if the data is noisy and overlaps between classes. XGBoost is efficient and offers state-of-the-art performance on structured data but requires careful tuning and can overfit if not regularized properly. LDA is excellent for understanding the data and works well when the classes are linearly separable but can perform poorly if the assumptions of normality and equal variance-covariance are not met.

**LITERATURE/RESEARCH TASK (Task IV)**

The fashion industry has seen a paradigm shift with the advent of clothing rental services, such as RentTheRunway, which provide consumers with a diverse array of designer clothing options without the commitment of purchase. Besides fashion from RentTheRunway, there are many other datasets similar to this, which are about music, movies, games, books, etc. to predict whatever the data analyser wants to find. For instance, datasets from platforms like Amazon and Netflix have been the cornerstone of many recommendation system studies. These works often employ collaborative filtering, content-based, and hybrid approaches to enhance the user experience by providing personalized recommendations. In the context of clothing, however, the recommendations are less about sequential consumption and more about fit, style, and occasion, making the problem unique and multidimensional. Each dataset will have a model to work fit with so we may need to spend some time working on that.

Currently, methods for analyzing such data include machine learning algorithms that can handle large, sparse matrices efficiently. That’s why we use some methods like Random Forest and boosting algorithms like XGBoost have been used to improve the accuracy of predictions. Sentiment analysis on review text to gauge user satisfaction is also a common approach, using both traditional bag-of-words models and more sophisticated natural language processing techniques.

Existing work in this area reveals a strong correlation between the sentiment expressed in reviews and the ratings provided by users. These findings are echoed in the analysis of the RentTheRunway dataset, where textual review analysis could predict user ratings with significant accuracy. The use of feature engineering to include metadata such as the occasion for rental, user demographic information, and item characteristics has proven beneficial in enhancing the model's performance. This aligns with the current understanding that multi-faceted features improve predictive models by incorporating a richer context of user behavior and preferences.

However, it's noteworthy that while the state-of-the-art methods provide a robust framework for prediction, the unique aspects of the RentTheRunway dataset, such as the influence of occasion on ratings, introduce new variables into the mix, necessitating the need for custom-tailored approaches. The exploration of these unique features could lead to insights that differ from standard recommendation system findings, emphasizing the importance of context in predictive analytics within the fashion rental space.

## CONCLUSION (Task V)

The analysis of the RentTheRunway dataset reveals that among various predictive models, Naive Bayes Gaussian and Linear Discriminant Analysis (LDA) stood out in terms of performance. This highlights that even relatively simple algorithms can effectively discern patterns within consumer behavior data in the fashion rental industry. The accuracy of these models signifies their ability to capture the essence of the dataset, suggesting that user satisfaction is closely tied to specific, discernible factors.

Feature representation was pivotal to the models' success. Continuous variables like text length and user age provided a strong foundation. Interestingly, certain features, notably the 'fit' attribute, emerged as significant predictors within models, suggesting that the fit of rented clothing is a key determinant of customer satisfaction. Conversely, models like SVM struggled, potentially due to the dataset's high-dimensionality or an unsuitable kernel choice.

The models' triumphs can be ascribed to a balanced approach to bias and variance, adept feature selection, and processing that resonates with the users' rating patterns. While successful models managed to grasp the dataset's complexity without overfitting, less effective ones perhaps failed to align with the dataset's unique properties or were insufficiently calibrated to its nuances.

## REFERENCES

[1] Mandy Lee, Rent the Runway Prediction by Mandy Lee - Issuu

[2] PrestonMWF, Rent-the-Runway-Recommender-System/Rent_the_Runway_Customer_Analysis_Recommender_System.ipynb at master · PrestonMWF/Rent-the-Runway-Recommender-System (github.com)

[3] Dataset Recommendation website, UCSD CSE Research. "Recommender Systems and Personalization Datasets" by Julian McAuley, UCSD. "Recommender Systems Datasets (ucsd.edu)"