Holly Cohan,Yesh Onipede, Yu Xia

Week 4 Report: Data Preprocessing and Preparation for Modeling

Imputing 'Price' for Missing Values

The target variable 'price' has 55,095 missing values out of approximately 693,000 total observations, accounting for about 8% of the data. Uber has a ride option called 'Taxi', which pairs riders with local taxi cab drivers. Every observation labeled 'taxi' is missing the price, and there are no missing values for any other type of ride. We considered simply removing all observations labeled 'Taxi' to remove the problem of the missing values, but this may affect our model's ability to generalize to the whole dataset, which would contain instances of taxis. Therefore, we initially planned to impute the missing values with prices based on the distance of the ride and the Boston Taxi rate. Taxi rates are available publicly on boston.gov at (https://police.boston.gov/taxi-rates/). Essentially, the passenger pays $2.60 for the first 1/7 mile of the ride, and $0.40 for each additional 1/7 mile.

However, we later found more information of the Taxi fare breakdowns on our Uber app (**Appendix 1.A**). It shows that the prices were calculated based on a base fare ($2.60), per minute rate ($0.47), and per mile rate ($2.80). Thus, the price is the sum of the base fare, distance fare and the time fare. We are not sure whether Uber charges exact taxi rates for their taxi option or if there are any additional fees. Since this is our best estimate, we will impute prices based on the price breakdown provided by Uber.

One challenge we encountered is that the duration of each ride is also not provided in the dataset, as well as the longitude and latitude for both source and destination. Since there is only one set of latitude and longitude in the dataset, we generated them using the package `geopy`. In addition, we used Mapbox Direction API to estimate the eta for those Taxi trips. Then, we use the estimated time and the distance to calculate the price. We also use our uber app to assess the imputation. For the record going from Haymarket to Back Bay, Uber estimated a price range $13-$18 (**Appendix 1.B**), and our imputed value is $18.05 (**Appendix 1.C**), which is pretty accurate.

Encoding Categorical Variables

We encoded the categorical variable using several different techniques based on the type and context of the variable.

- **Dummy-Coding:**

    - We used dummy-coding for *cab_type*, which indicates whether the car is an Uber or a Lyft.
- **One-Hot Encoding**:
    - We used one-hot encoding for *long_summary*, which contains a written summary of the weather conditions.
    - We used one-hot encoding for *source* and *destination*, which contain the locations that the ride starts from and ends at, respectively. The possible values of these variables are: [Haymarket Square, Back Bay, North End, North Station, Beacon Hill, Boston University, Fenway, South Station, Theatre District, West End, Financial District, Northeastern University].
    - We used one-hot encoding for *name*, which contains the type and level of the car that is called. The possible values of these variables are: [Shared, Lux, Lyft, Lux Black XL, Lyft XL, Lux Black, UberXL, Black, UberX, WAV, Black SUV, UberPool, Taxi].
- **Ordinal Encoding**:
    - We used ordinal encoding for *short_summary*, which contains a brief written summary of the weather conditions and has the following natural logical order to the possible values: [Clear, Foggy, Partly Cloudy, 'Mostly Cloudy, 'Overcast, Possible Drizzle, Drizzle, Light Rain, Rain].
    - We used ordinal encoding for *icon*, which contains another brief written summary of the icon which represents the weather conditions of the given day and has the following natural logical order to the possible values: [clear-day, clear-night, partly cloudy day, partly cloudy night, fog, cloudy, rain].
- **Variables Removed:**
    - We removed the *timezone* variable since every observation in our dataset is from Boston and therefore has the same timezone of ‘America/New_York’. Since every observation has the same value, this variable will not add any value to our modeling.
    - The *product_id* variable appears to have identical information to the *name* variable, so we decided to drop it as it does not provide any additional useful information and will add no additional value to our analysis.

PCA and Standardization

After encoding all the categorical variables, we were left with 109 features, which might work for more complex models. However, we are not going to extract valuable predictions and insights using simple

models like linear regression. Therefore, we conducted PCA to address high dimensionality and correlation identified from last week.

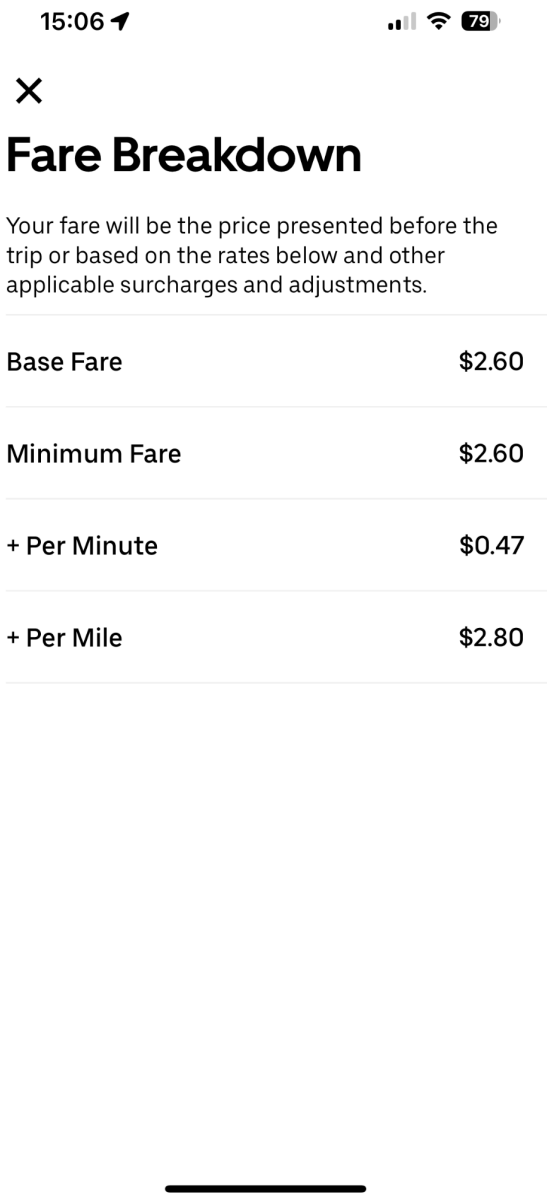
After encoding all the categorical variables, we were left with 109 features, which might work for more complex models. However, we are not going to extract valuable predictions and insights using simple models like linear regression. Therefore, we conducted PCA to address high dimensionality and correlation identified from last week.

We set a threshold of 0.95 for the cumulative explained variance ratio, which resulted in retaining **N** principal components. This threshold was chosen to ensure that we capture at least 95% of the variance in the dataset while reducing the dimensionality of our feature space. By selecting these components, we aim to minimize noise and redundancy in the data, thus enhancing the model's performance and interpretability.

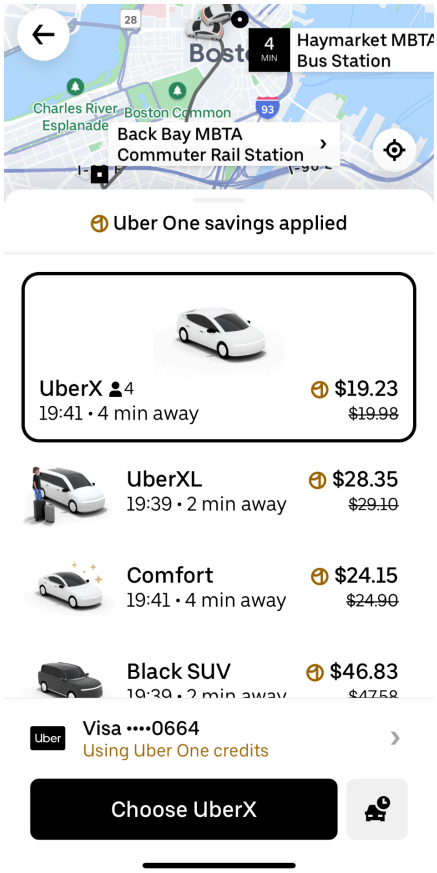
Before applying PCA, we standardized the features using StandardScaler. Standardization is crucial as it ensures that each feature contributes equally to the distance calculations involved in PCA. Without standardization, features with larger ranges could dominate the PCA results, leading to biased components that do not accurately represent the underlying structure of the data. We also saved the df locally after standardizing them for future convenience in case we want to use the processed data without PCA for some models.

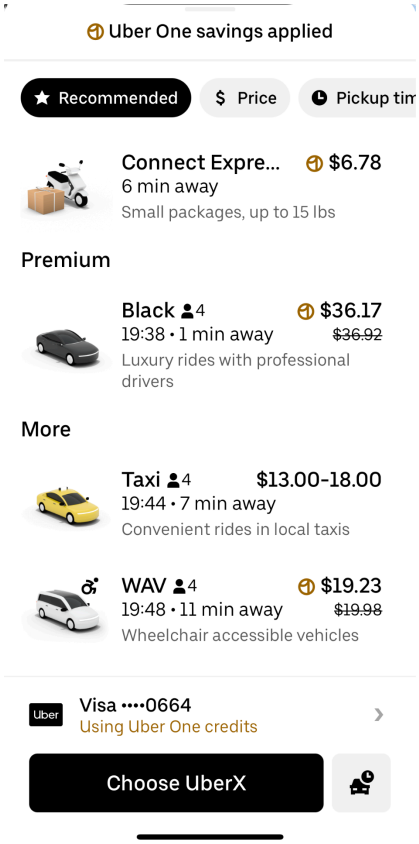
By combining PCA with standardization, we can effectively reduce dimensionality while preserving the most significant variance in our dataset, making it more suitable for linear regression and other predictive modeling techniques.

Appendix 1.A Uber Taxi Price Breakdown



Appendix 1.B Haymarket to Back Bay Uber Taxi Price





Appendix 1.C Haymarket to Back Bay Uber Taxi Price Imputation

```
df_taxi[["source", "destination", "price"]].head()
```

[17] ✓ 0.0s Python

	source	destination	price
39	Haymarket Square	Back Bay	18.046207
44	Haymarket Square	Beacon Hill	9.662757
61	Beacon Hill	Boston University	14.543172
71	Beacon Hill	South Station	14.665433
109	Theatre District	Northeastern University	848.548153