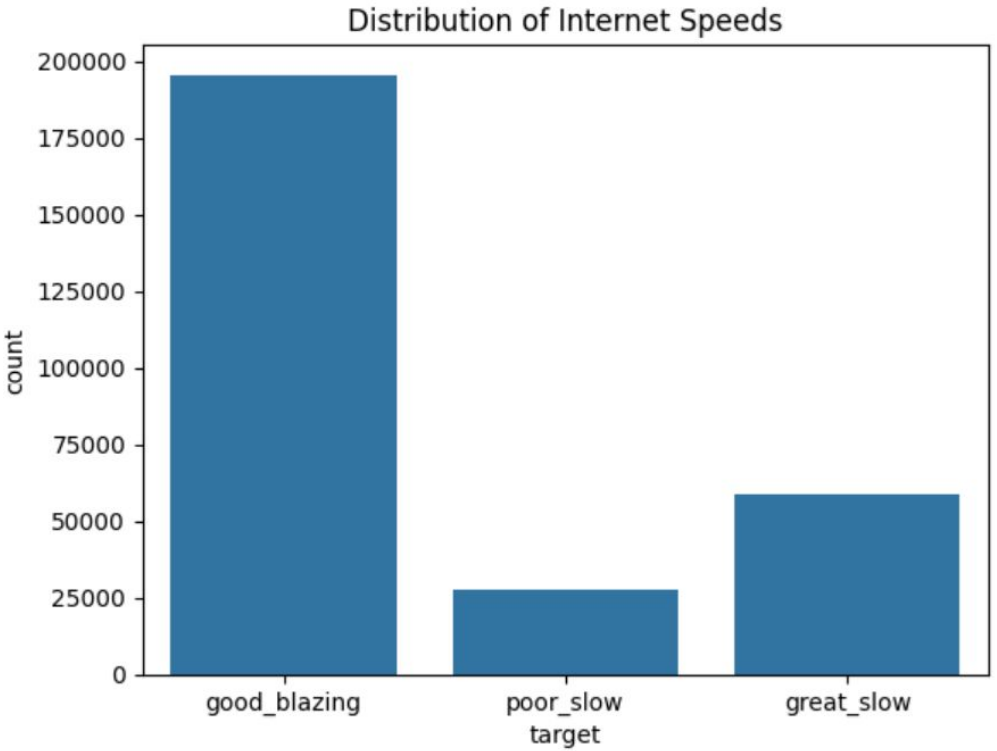
Building a predictive machine learning pipeline: Verizon Internet Service Offers

La’Tisa Ward and Lesley Frew
CS 624 Fall 2024

Problem Statement

Users want the fastest internet at the best price.

What features are best predictors of the Verizon internet offer (speed with price)?



Data Loading

```
df_verizon = pd.read_csv('speed_price_verizon.csv')
```

The entire dataset (282,622 rows) fits in memory, so we did not sample.

Initial Data Analysis

Review from last assignment:

This is a tabular with multiple variables.

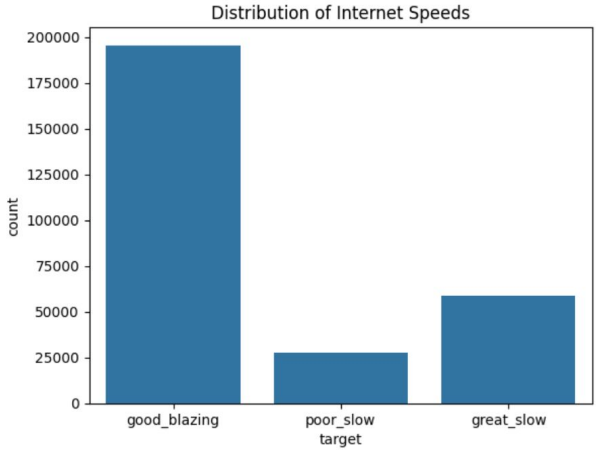
Numeric: speed_down, fastest_speed_price, race_perc_non_white, cost_per_mbps(=price/speed_down)...

String categorical: major_city, redlining_grade, speed_desc(=slow...blazing)...

Nulls: income_dollars_below_median (20k), speed_unit (58k), redlining_grade (69k)...

We did not identify any linear relationships.

New this assignment:



We added the cost_desc column which is great(<.10), good(<.25), or poor(else).

We added the target column which is the cost_desc concatenated with the speed_desc.

There is a **class imbalance problem** with our target classification column.

Data Cleaning

- We removed 3 classes with barely any membership

```
df_verizon = df_verizon.drop(df_verizon[df_verizon['target'] == 'poor_medium'].index)
df_verizon = df_verizon.drop(df_verizon[df_verizon['target'] == 'poor_fast'].index)
df_verizon = df_verizon.drop(df_verizon[df_verizon['target'] == 'poor_blazing'].index)
```

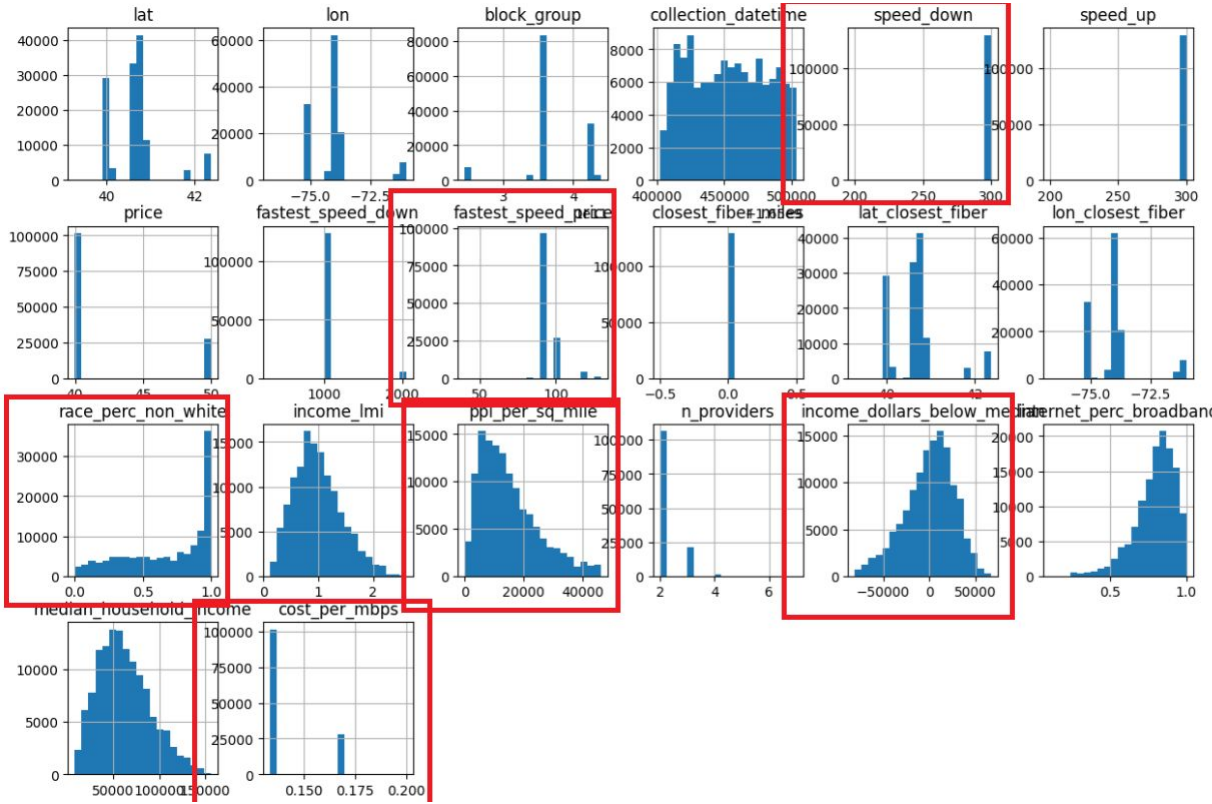
- We removed 89,110 outliers from the following columns:
  - speed_down, fastest_speed_price, race_perc_non_white, ppl_per_sq_mile, income_dollars_below_median, cost_per_mbps

- We removed 52,084 rows with null values from all columns.

```
df_verizon = df_verizon.dropna()
```

Data Visualization - distributions

After removing outliers, there is still skew in many of the features we have chosen.



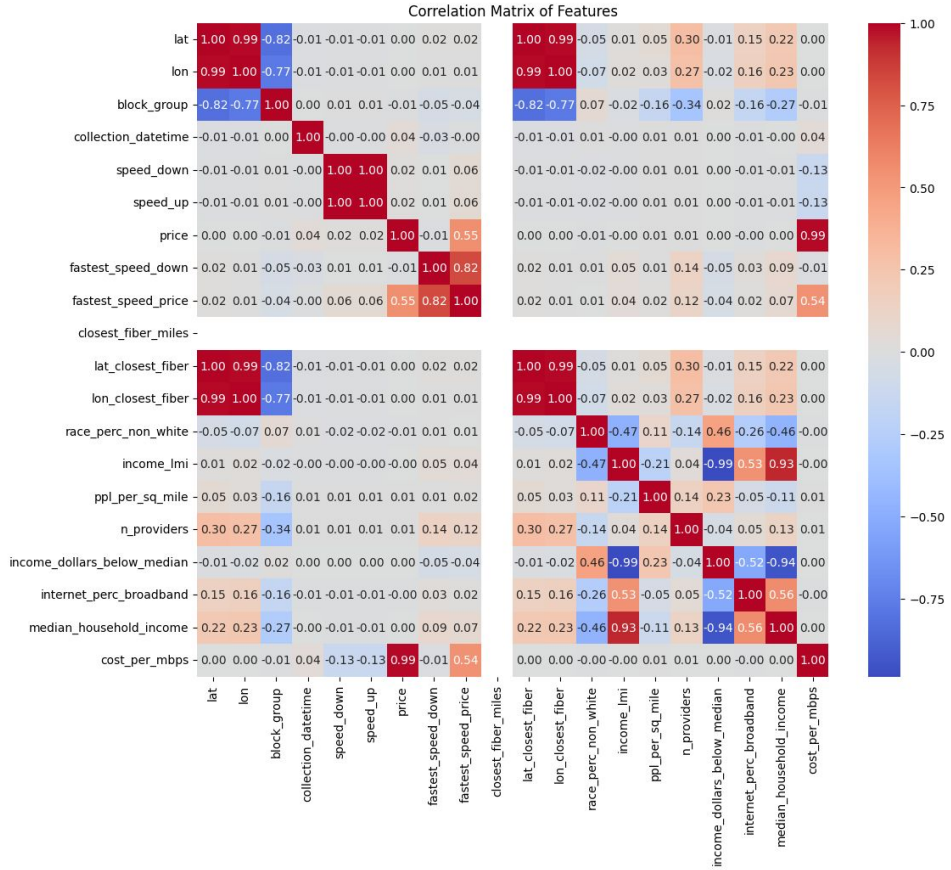
Data Visualization - correlations

Features with no correlation to price:

`speed_down`
`race_perc_non_white`
`ppl_per_sq_mile`
`income_dollars_below_median`

Positive correlations to price:

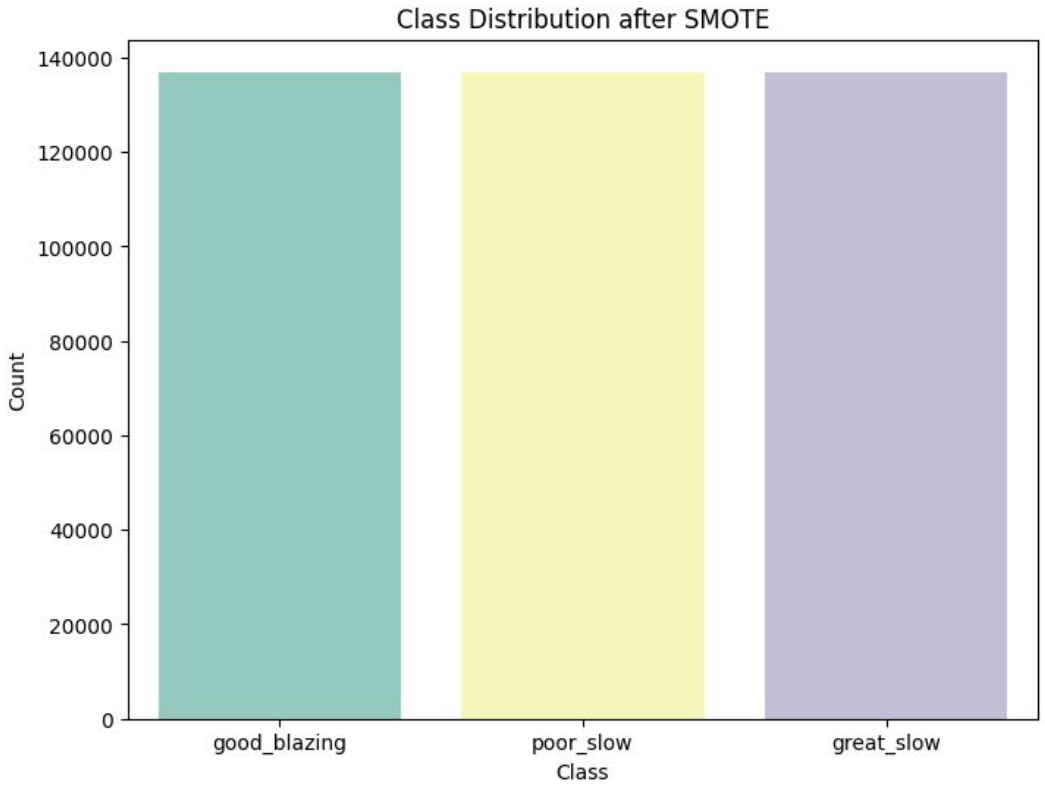
`Fastest_speed_price` (magnitude of .55)
`Cost_per_mbps` (magnitude of .99)



Class imbalance

We used SMOTE to fix the class imbalance problem.

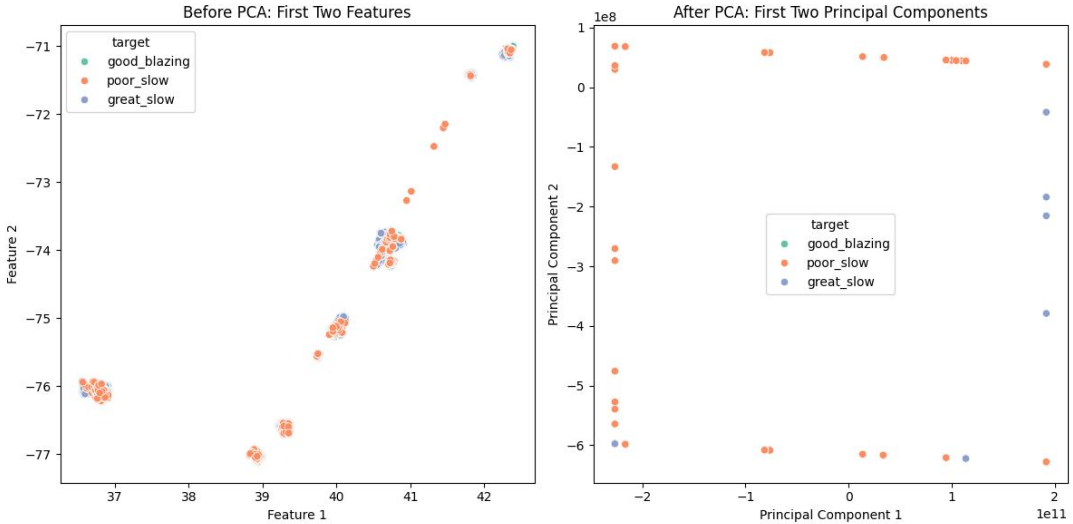
All poor_slow and great_slow were in outlier/na rows so we had to restore those and replace na with 0.



Feature Engineering

We used Principal Component Analysis and standardization on our columns.

We **removed** price/speed columns because we are trying to predict those.



Training

We used 5-fold cross validation for our training.

Fold 1:
Training set: 328800 samples
Validation set: 82200 samples

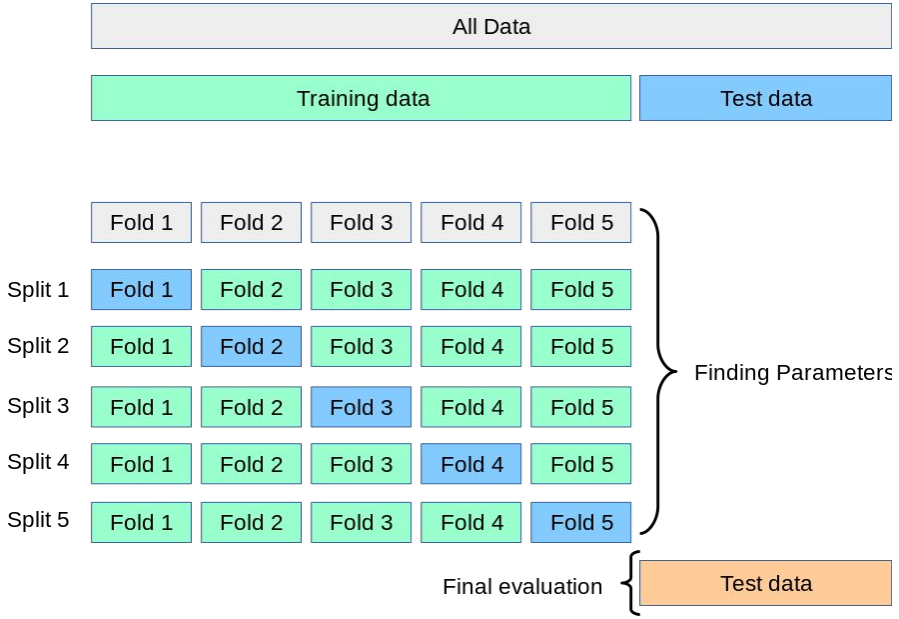
Fold 2:
Training set: 328800 samples
Validation set: 82200 samples

Fold 3:
Training set: 328800 samples
Validation set: 82200 samples

Fold 4:
Training set: 328800 samples
Validation set: 82200 samples

Fold 5:
Training set: 328800 samples
Validation set: 82200 samples

K-Fold Cross-Validation setup with 5 folds.

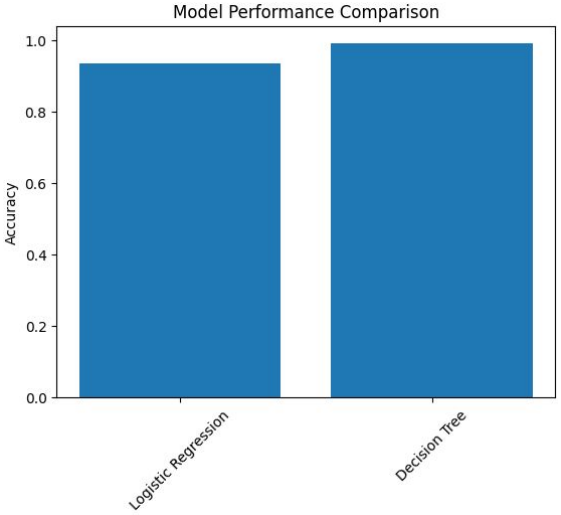


https://scikit-learn.org/1.5/modules/cross_validation.html

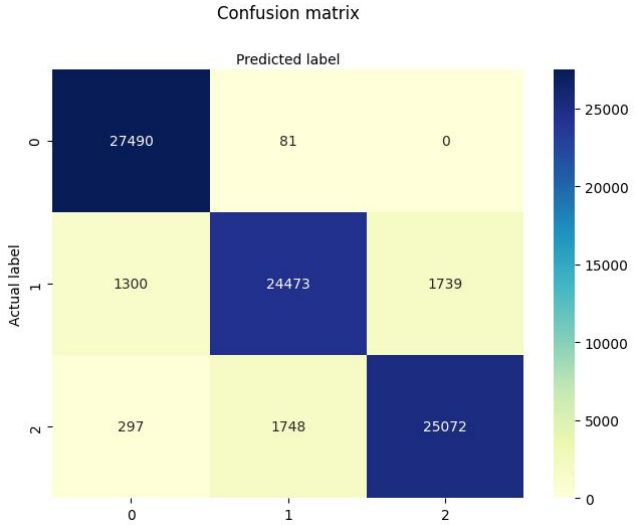
Performance Evaluation

We had good performance on simple models, so we just modeled Logistic and Decision Tree (most appropriate for this task). The other models were time prohibitive without sampling and would not offer better performance. The decision tree had over 1,200 decisions in it!

Model performance comparison



Confusion Matrix



Conclusion

Strengths and limitations of models:

- Strength: The models had good (>.90) accuracy
- Limitations: The decision tree model is likely overfit and unexplainable (what does Verizon actually use to make these offers?)

Future work or improvements:

- Try other models with sampling (KNN, random forest)
- Determine how to calculate a more explainable decision tree