AIRBNB BOOKING PREDICTION

Bentley Ou
11/01/19

BACKGROUND

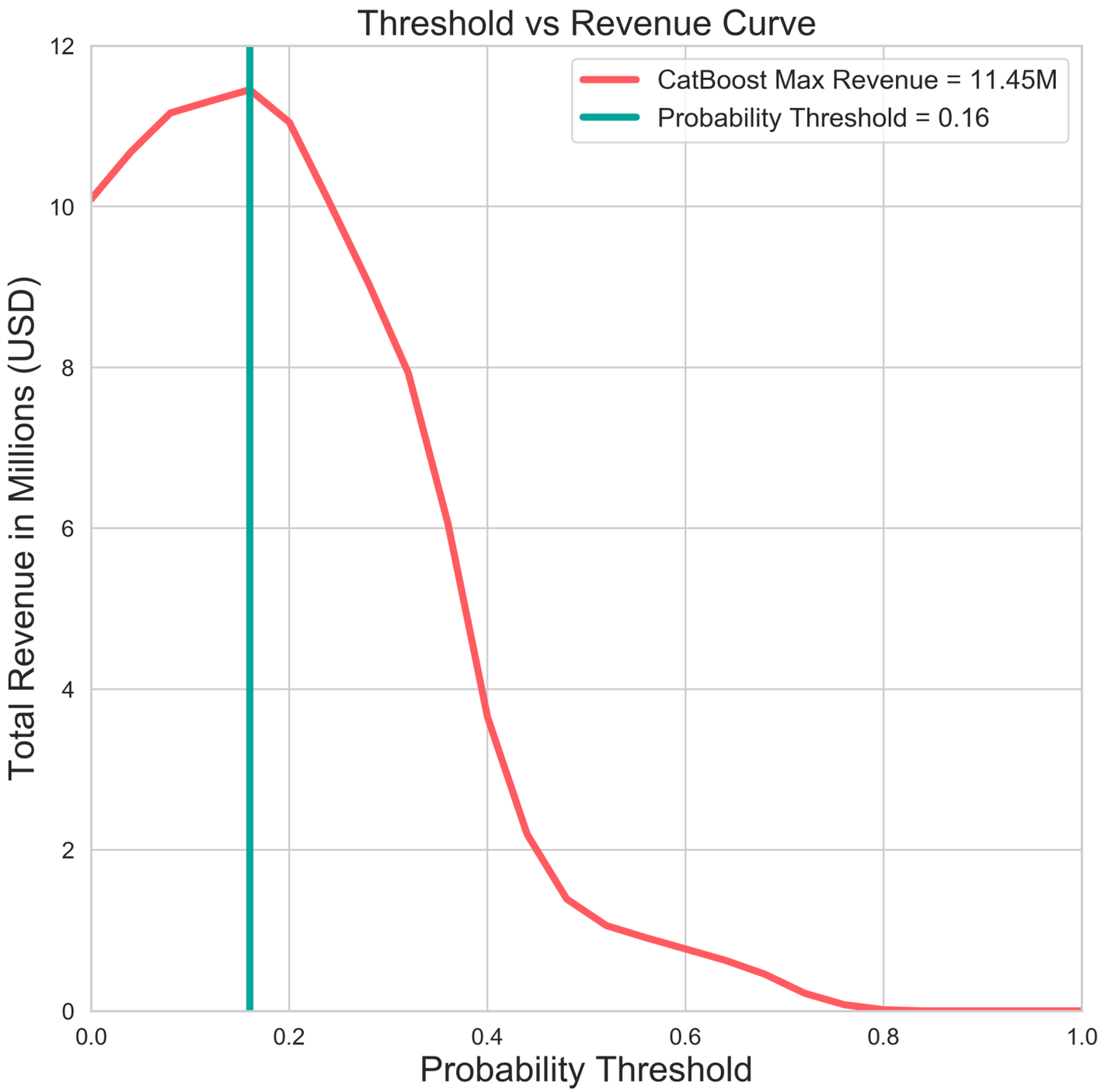
- In 2020, Airbnb is forecasted to have about **2.2 million** first time users.
- **23%** of first time users are expected to make a reservation.
- The average home reservation is around **$160/night**.
- AirBnb generates about **$32** of revenue per reservation on average
  - **17%** fees from bookers
  - **3%** fees from hosts

PROBLEM STATEMENT

- The Marketing team wants to launch a campaign initiative to effectively advertise to new users.
  - Cost of advertisement: $5
  - Immediately revenue per user acquisition: $32

- Objective: maximize revenue for the campaign
- We want to **maximize campaign revenue** by targeting first time users who are likely to make a reservation

RESULTS



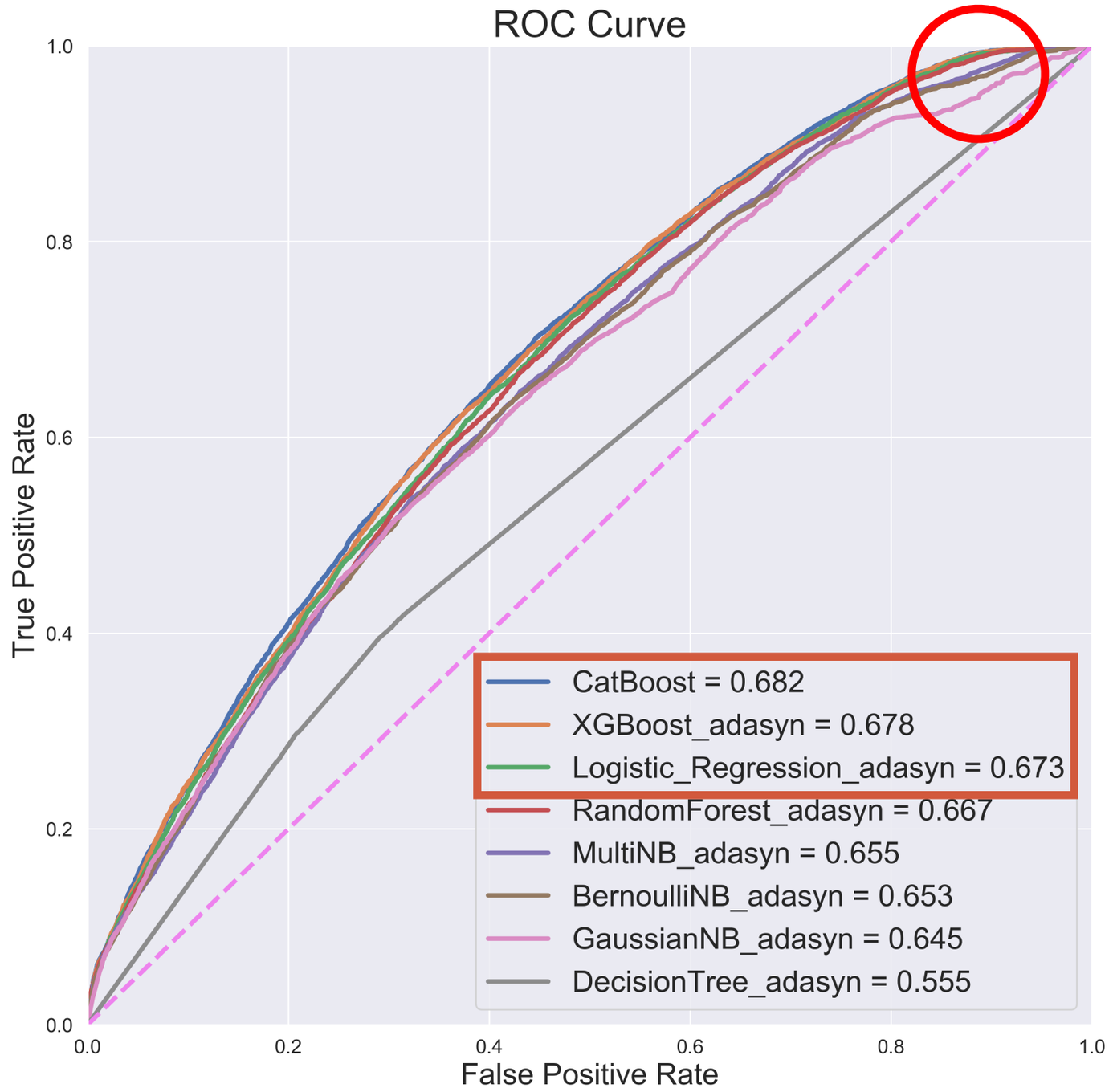
- Classifier:
  - **CatBoost Model**
- Targeted users:
  - **They have at least 16% chance of making a reservation**
- Maximum profit using this business constraint:
  - **$11.45M USD**

METHODS AND APPROACHES

- Collecting user data and preprocessing
  - age, gender, browser, device, signup flow, signup method, ad affiliation, etc…
- Labeling
  - Class 1: User who makes a reservation within 5 days of signing up for an account (minority class - 20%)
  - Class 0: User who do not make a reservation within 5 days of signing up for an account (majority class - 80%)
- Build a binary classification model
  - Logit model baseline = 0.6 AUC
- Validation and Evaluation
  - Choosing evaluation metrics (AUC)
  - Choosing probability threshold (business constraint)
- Model Deployment - Flask App

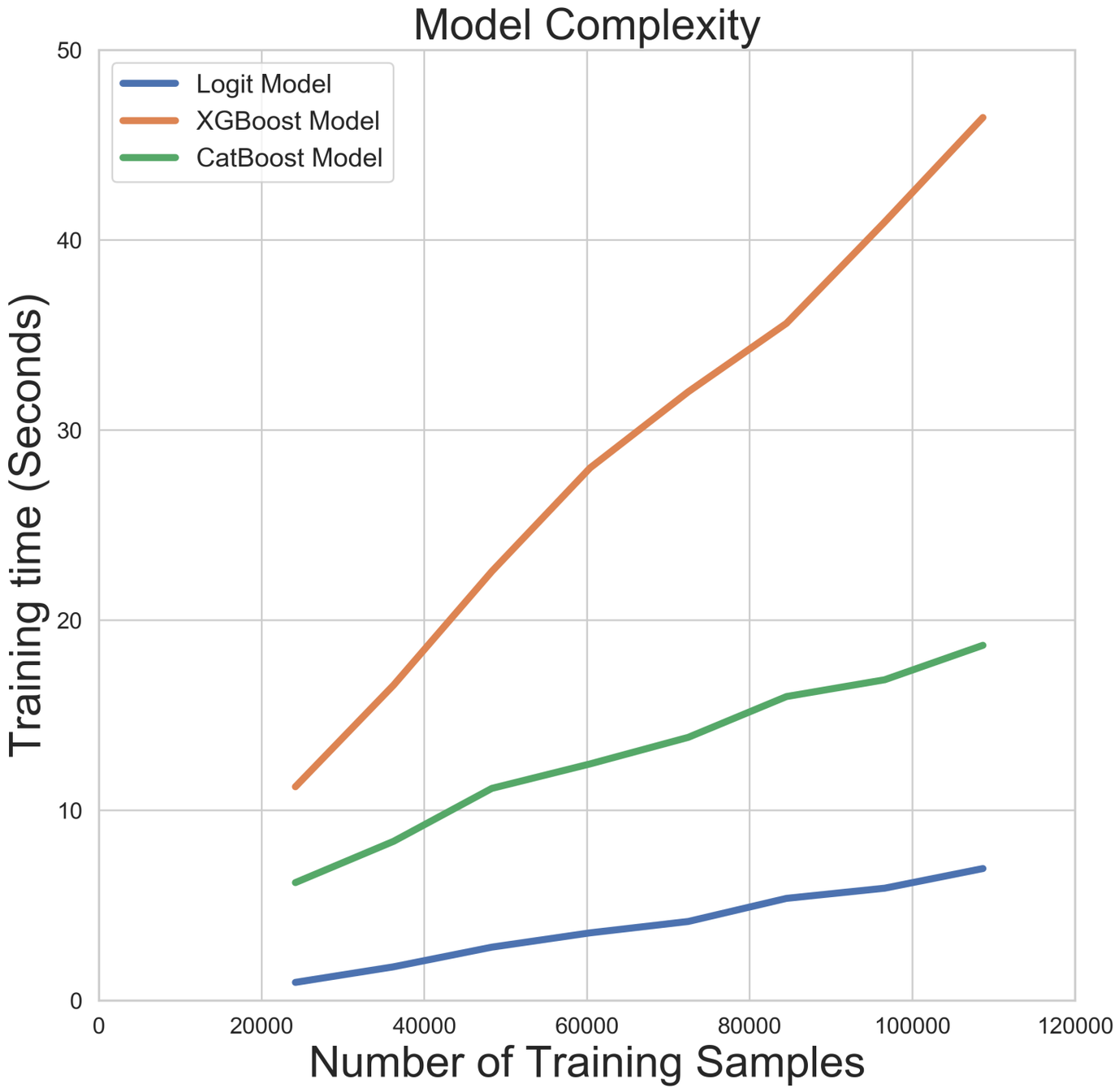
EVALUATION

- AUC
  - We want low threshold - high FPR and TPR
  - Operating at upper right corner
- Model candidates
  - CatBoost
  - XGBoost
  - Logit model



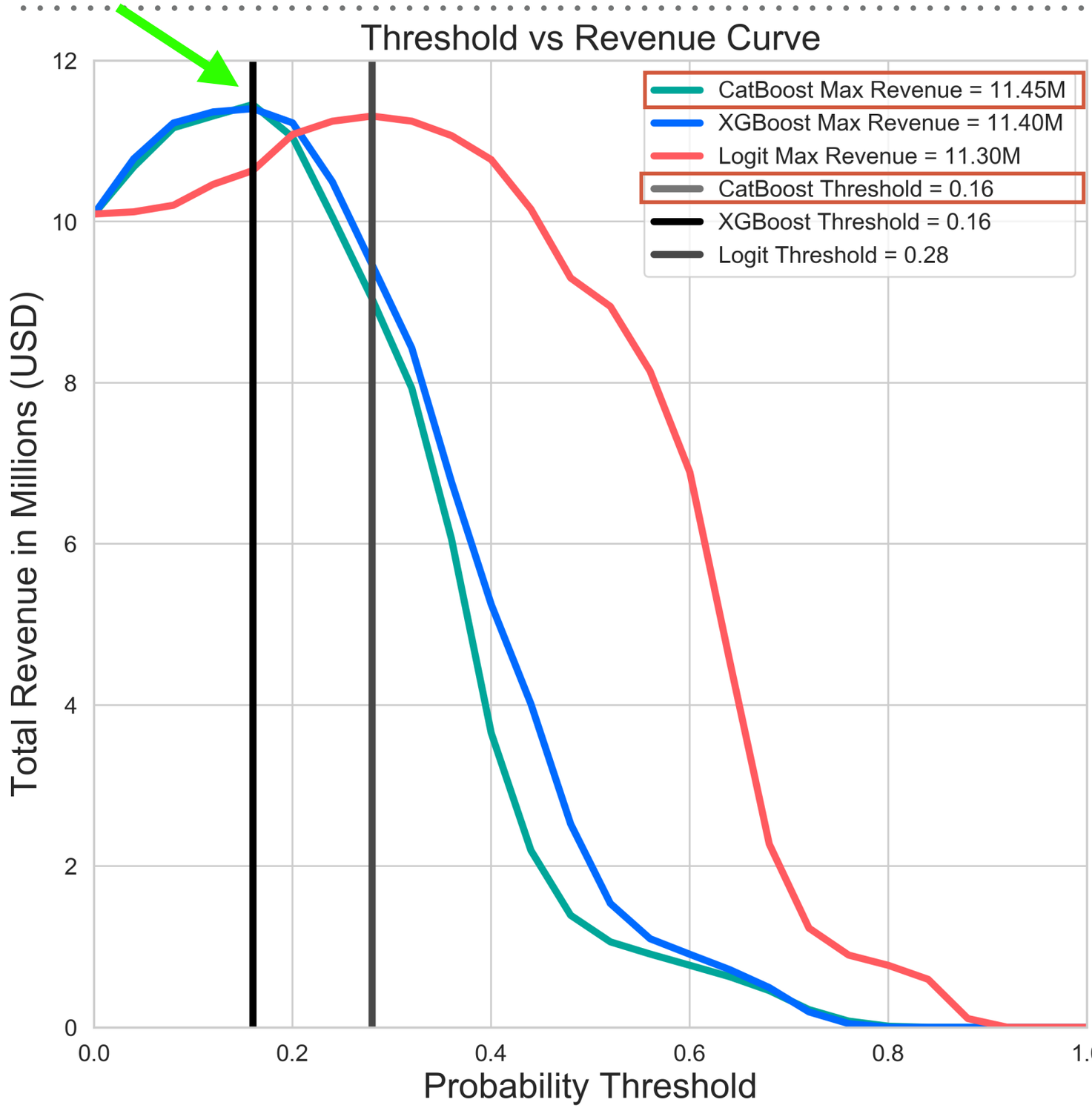
ROC and AUC for each classification model

MODEL COMPLEXITY COMPARISON



- CatBoost Model
  - 2.4x faster than XGBoost
  - Comparable to Logit Model
  - CatBoost rebuilds the data in a parallel manner
  - Gives up to 3x speedup compared to other tree-based models

REVENUE COMPARISON



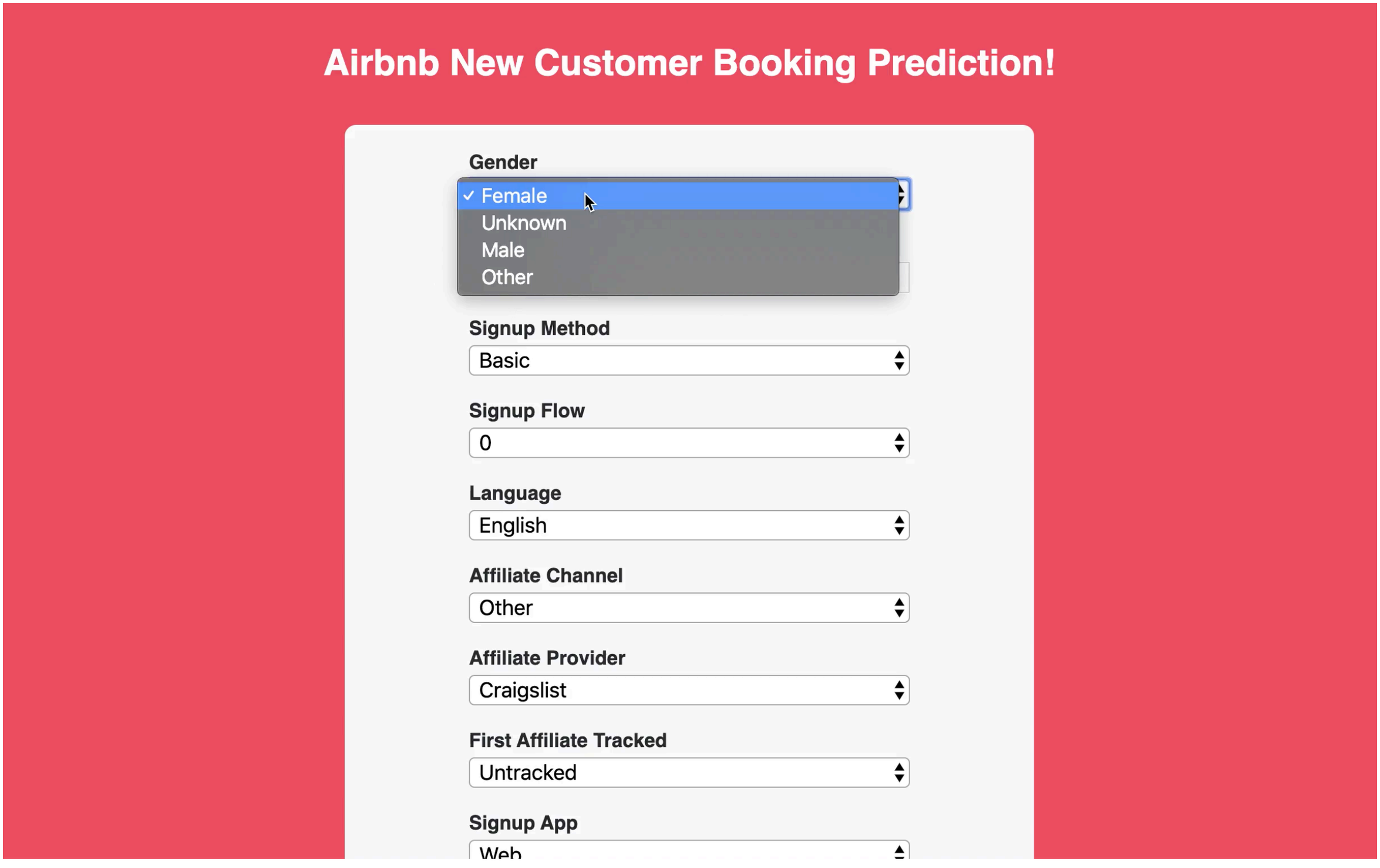
CatBoost Model

- Yields the most revenue
- More interpretable than XGBoost
  - Takes raw categorical features
    - No need for one-hot-encoding
  - 11/13 cat features

RECOMMENDATIONS – USER CONVERSION

- User experience on the web app increases conversion
  - All else constant - Users signing up on a web app have a 6 to 1 odd compared to users who are not
- Specific signup flows are better
  - 24, 3, 12

FLASK APP DEMO



CONCLUSION

- In 2020, a total of 2.2 million new users will be on AirBnb
  - We should target users who have at least 16% chance of making a reservation
- Web app experience and specific signup flows translate to higher user conversion
- Using the CatBoost classifier would yield an expected revenue of $11.45M

THANK YOU

Questions?

APPENDIX

FUTURE WORK

- In order to increase revenue
  - Improve model AUC of 0.682
- Obtain more features
  - User engagement information on the platform
  - User demographics

ECONOMICS

- Based on dataset sample frequency, 26% of all new users will make a booking
- AirBnb generates a revenue of about $32 from each nightly booking per user
- Confusion Matrix:
  - Cost for True Positives: $5 from ad campaign
  - Cost for True Negatives: $0
  - Cost for False Positives: $5
  - Cost for False Negatives: $0
  - Revenue for True Positives: $32 from booking revenue
  - Revenue for True Negatives: $0
  - Revenue for False Positives: $0
  - Revenue for False Negatives: $0

ECONOMICS CALCULATION

```
Cost_TP = cost_advertisement_per_user
Cost_TN = 0
Cost_FP = cost_advertisement_per_user
Cost_FN = 0

Revenue_TP = revenue_per_booker
Revenue_TN = 0
Revenue_FP = 0
Revenue_FN = 0

Total_revenue_cat = []

for threshold in np.linspace(0,1,26):
    TN = confusion_matrix(y_test_catboost, catboost_adasyn_clf.predict_proba(X_test_catboost)[:,1] > threshold)[0][0]
    TP = confusion_matrix(y_test_catboost, catboost_adasyn_clf.predict_proba(X_test_catboost)[:,1] > threshold)[1][1]
    FN = confusion_matrix(y_test_catboost, catboost_adasyn_clf.predict_proba(X_test_catboost)[:,1] > threshold)[1][0]
    FP = confusion_matrix(y_test_catboost, catboost_adasyn_clf.predict_proba(X_test_catboost)[:,1] > threshold)[0][1]

    num_test_users = len(X_test_catboost)

    # Total = Fraction of users for (TP, TN, FP, FN) * total users * Revenue per users

    Total = ((TP/num_test_users) * 3e6 * (Revenue_TP - Cost_TP) +
             (TN/num_test_users) * 3e6 * (Revenue_TN - Cost_TN) +
             (FP/num_test_users) * 3e6 * (Revenue_FP - Cost_FP) +
             (FN/num_test_users) * 3e6 * (Revenue_FN - Cost_FN))

    Total_revenue_cat.append(Total/1e6)
```

END OF SLIDES

Thank you!
Bentley Ou