Supervised Learning Project

Prediction Customer Churn with Video Streaming Data

EDA (Exploratory data analysis)

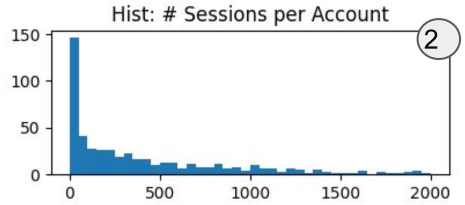
1. Summary - 116 days of data across ~4 months (2018-04-30). 325000 records. 1000 accounts.
2. Hist - Sessions per account (long tail)
3. Hist - Price distribution (2 humps at ~31$ & ~12$)
4. Hist - Highest mos score is 4.6. 3% = 0.
5. Hist - Avg. Session duration per user

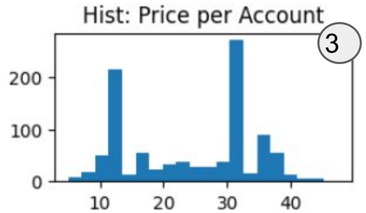
Summary - Need more context / meta data

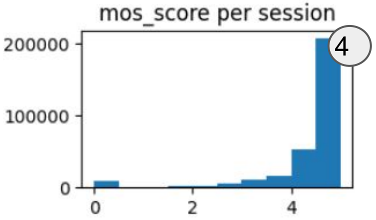
1

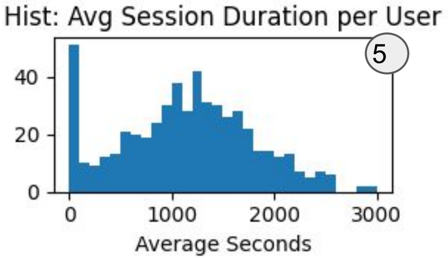
streams_df
account_number
start_timestamp
end_timestamp
mos_score

accounts_df
account_number
price
churned





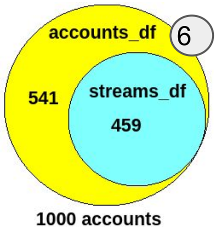


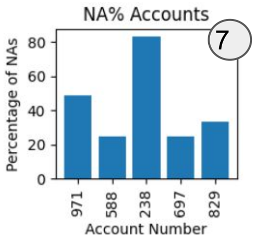


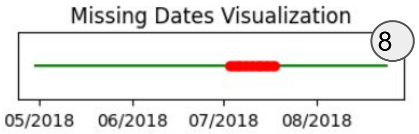
Data Integrity

6. Accounts with no streaming data (54%)
7. Nulls in 5% of rows (start_timestamp & mos_score)
8. Missing dates (7-3 to 7-18 i.e. 16 days)
9. Duplicate rows (24)
10. Stream duration outliers (3 rows) & negatives (5 rows)

Summary - Data mostly good!







10

Outliers in column 'duration_seconds':

	account_number	duration_seconds	Z Score
125493	573	78899.0	12.0
197377	952	79194.0	12.0
99053	573	86111.0	13.0
203567	242	639271.0	99.0
135117	213	1717080.0	268.0
306705	345	2925841.0	456.0

Feature Engineering

1. Adding features
    a. Features at stream level
        duration_seconds
        sinusoidal features
        dummy features
    b. Rolled up aggregation features
        aggregated last 5 sessions
        aggregated all sessions
    c. Merged streams_df & accounts_df (full outer)

2. Feature importance (random forest)
    a. Consistent viewing
    b. price
    c. many sessions
    d. mos_score & short sessions

3. Autoencoders / pca (didn't do this)

Summary - over-engineered for this use case

streams_df (1)

```
account_number
start_timestamp
end_timestamp
mos_score
```

streams_df + features (1a)

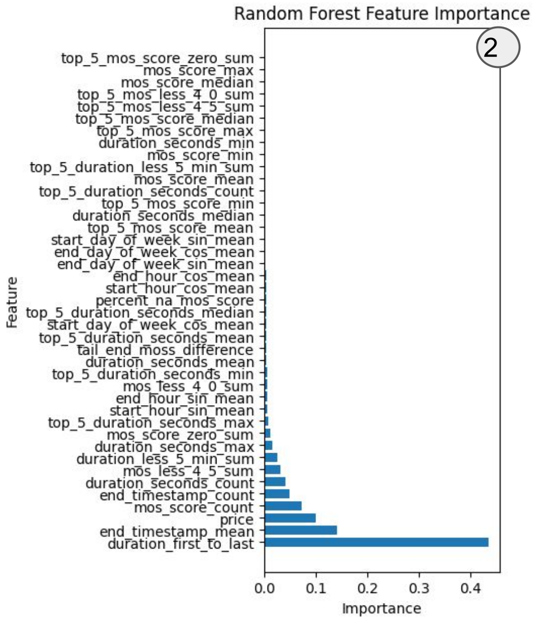
```
account_number
start_timestamp
end_timestamp
mos_score
duration_seconds
start_hour_sin
start_hour_cos
start_day_of_week_sin
start_day_of_week_cos
end_hour_sin
end_hour_cos
end_day_of_week_sin
end_day_of_week_cos
mos_score_zero
duration_less_5_min
duration_less_1_min
mos_less_4_5
mos_less_4_0
```

streams_df rolled up with agg features (1b)

```
account_number
start_timestamp_min
end_timestamp_max
end_timestamp_mean
end_timestamp_count
duration_seconds_max
duration_seconds_mean
duration_seconds_min
duration_seconds_median
duration_seconds_count
mos_score_count
mos_score_max
mos_score_mean
mos_score_min
mos_score_median
start_hour_sin_mean
end_hour_sin_mean
start_hour_cos_mean
end_hour_cos_mean
start_day_of_week_sin_mean
end_day_of_week_sin_mean
start_day_of_week_cos_mean
end_day_of_week_cos_mean
mos_score_zero_sum
duration_less_5_min_sum
mos_less_4_5_sum
mos_less_4_0_sum
top_5_duration_seconds_max
top_5_duration_seconds_mean
top_5_duration_seconds_min
top_5_duration_seconds_median
top_5_duration_seconds_count
top_5_mos_score_max
top_5_mos_score_mean
top_5_mos_score_min
top_5_mos_score_median
top_5_mos_score_zero_sum
top_5_duration_less_5_min_sum
top_5_mos_less_4_5_sum
top_5_mos_less_4_0_sum
```

streams_df merged with accounts_df (1c)

```
account_number
end_timestamp_mean
end_timestamp_count
duration_seconds_max
duration_seconds_mean
duration_seconds_min
duration_seconds_median
duration_seconds_count
mos_score_count
mos_score_max
mos_score_mean
mos_score_min
mos_score_median
start_hour_sin_mean
end_hour_sin_mean
start_hour_cos_mean
end_hour_cos_mean
start_day_of_week_sin_mean
end_day_of_week_sin_mean
start_day_of_week_cos_mean
end_day_of_week_cos_mean
mos_score_zero_sum
duration_less_5_min_sum
mos_less_4_5_sum
mos_less_4_0_sum
top_5_duration_seconds_max
top_5_duration_seconds_mean
top_5_duration_seconds_min
top_5_duration_seconds_median
top_5_duration_seconds_count
top_5_mos_score_max
top_5_mos_score_mean
top_5_mos_score_min
top_5_mos_score_median
top_5_mos_score_zero_sum
top_5_duration_less_5_min_sum
top_5_mos_less_4_5_sum
top_5_mos_less_4_0_sum
percent_na_mos_score
tail_end_moss_difference
price
churned
duration_first_to_last
```



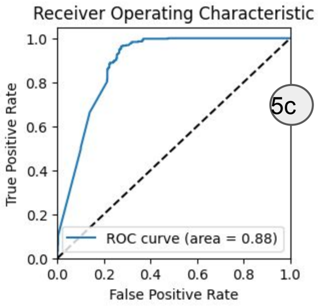
Supervised Learning

1. Model Design - window of time (ideally rolling window)
2. Model Comparison
    a. K-fold cross validation (3)
    b. confusion matrix
3. Model Selection (random forest) - Why?
    a. Intuitive interpretation
    b. avoid overfits well
4. Model Tuning
    a. take top 15 features only (reduce overfit)
    b. precision vs recall
    c. class imbalance (41% churned)
5. Model Evaluation
    a. learning curve (overfit vs underfit)
    b. ROC curve
6. Convert binary prediction to probability

Summary - need more data and better model design

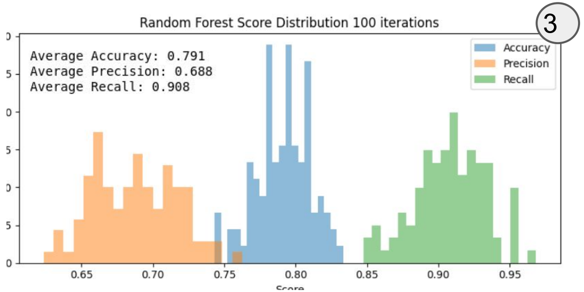
4

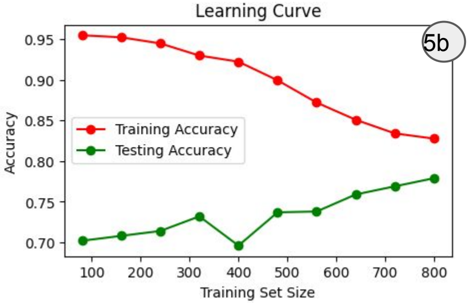
```
rf = RandomForestClassifier(
    n_estimators = 100,
    min_samples_split = 10,
    min_samples_leaf = 5,
    max_depth=4,
    criterion="entropy",
    class_weight={False:1,True:2},
    max_features=.5
```



2

```
0.801 Average Accuracy: Random Forest
Confusion Matrix:
[[421 166]
 [ 33 380]]
Precision: 0.696
Recall: 0.920

0.800 Average Accuracy: Gradient Boost
Confusion Matrix:
[[434 153]
 [ 47 366]]
Precision: 0.705
Recall: 0.886

0.775 Average Accuracy: XGBoost
Confusion Matrix:
[[441 146]
 [ 79 334]]
Precision: 0.696
Recall: 0.809

0.754 Average Accuracy: Logistic Regression
Confusion Matrix:
[[395 192]
 [ 54 359]]
Precision: 0.652
Recall: 0.869

0.753 Average Accuracy: SVM
Confusion Matrix:
[[374 213]
 [ 34 379]]
Precision: 0.641
Recall: 0.918

0.729 Average Accuracy: Naive Bayes
Confusion Matrix:
[[331 256]
 [ 15 398]]
Precision: 0.609
Recall: 0.964
```





6

account number	price	total sessions	prediction	churned	probability
31	33.57	2808	0	0	0%
34	17.3	828	0	0	1%
36	28.07	156	0	0	3%
28	30.56	413	0	0	4%
37	30.56	92	0	0	8%
26	30.01	34	0	0	15%
33	34.55	212	1	1	61%
32	36.9	2	1	0	64%
27	12.62	7	1	1	87%
38	12.62	139	1	1	92%

Conclusion, Learnings & Issues

Conclusion

- Learned many things
- Model got good performance

Learning

- Recall vs Precision Focus
- Accuracy is not a good metric

Issues

- Considerable Data Quality Issues
- Data Leakage Issue

Fin

Thank you for your time!