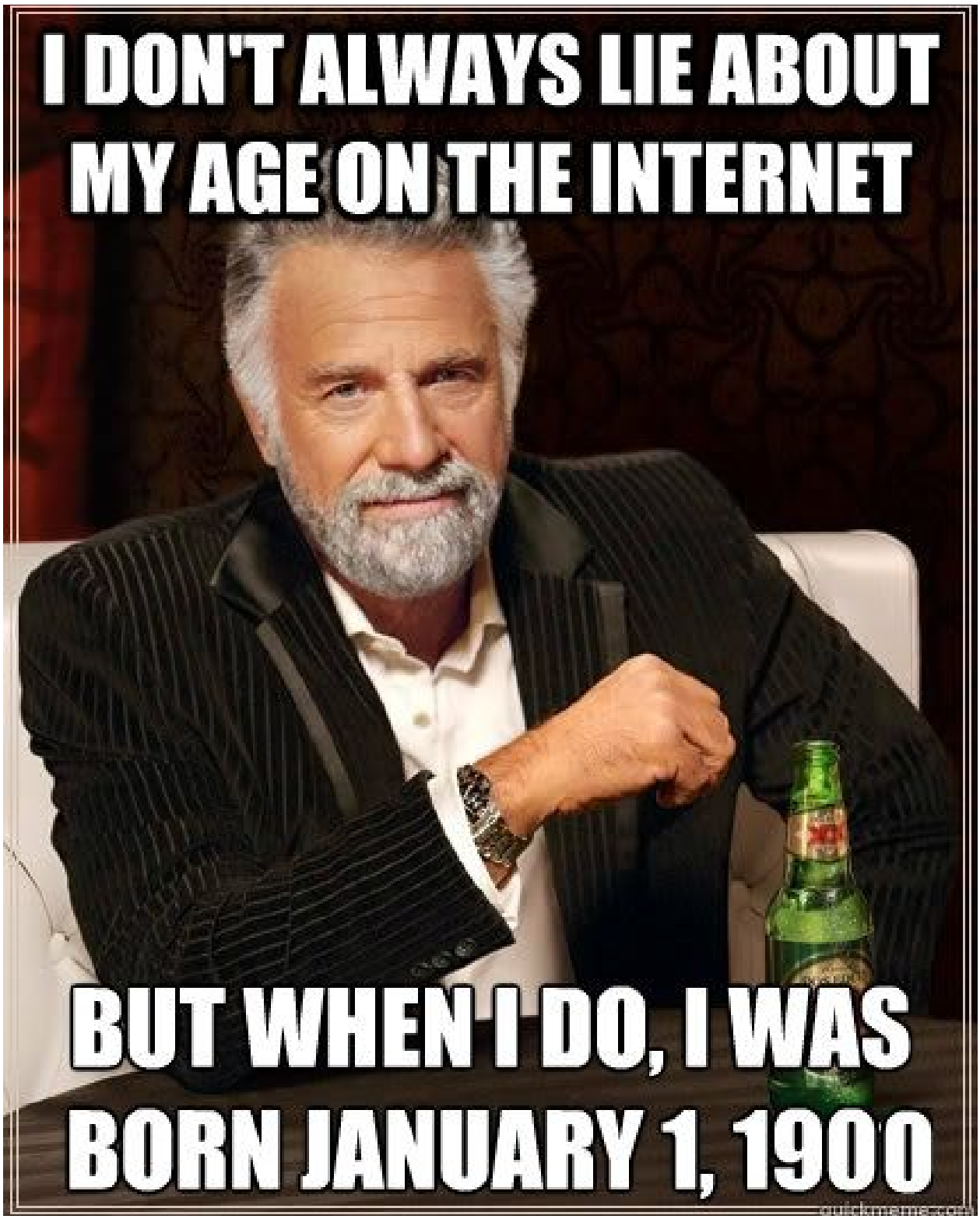
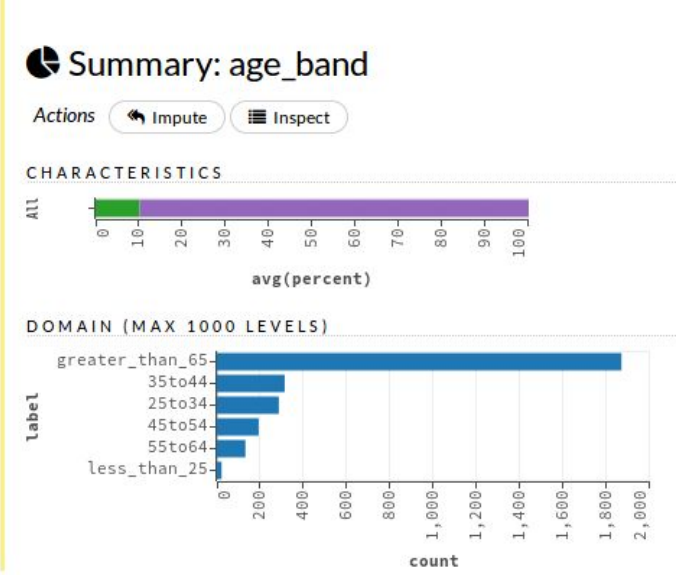

I DON'T ALWAYS LIE ABOUT MY AGE ON THE INTERNET
BUT WHEN I DO, I WAS BORN JANUARY 1, 1900
quickmeme.com

PROBLEM STATEMENT:

The objective of the exercise is to create a model to predict and assign age-bands to each of the individual households present in the dataset. Below is the rationale for the solution

The meme was a half joke; because it's actually true

With 66% of families having quoted their age as 1900-01-01 we have a heavily imbalanced dataset, where 1868 families (out of 2813) have mentioned or not mentioned and hence defaulted their age as 1900-01-01.



Assumptions and pitfalls of the dataset

Each of these families may have family members who might belong to different age groups, having different viewing habits but are all being grouped together to the age group of the person on whose name the connection is. Further the cardinality of the categoricals in the raw data is huge and one hot encoding the raw categoricals is going to induce the famous curse of dimensionality because of the feature explosion.

Clickstream data analysis is a tough problem and requires large amount of data however we just have one month of data at our disposal. In production at least one full year of data needs to be evaluated on.

Lets view what the aggregated data per family id says

PySpark 2.2.0 was used to aggregate the data set per household id and the feature generation was done based on the NIPS 2015 winning solution, resulting in 42 features after selective pruning(RFE) and PCA of the features set. The pyspark aggregation notebook is attached. After grouping the dataset per household id, aggregations were carried on to find out the following features per household id:

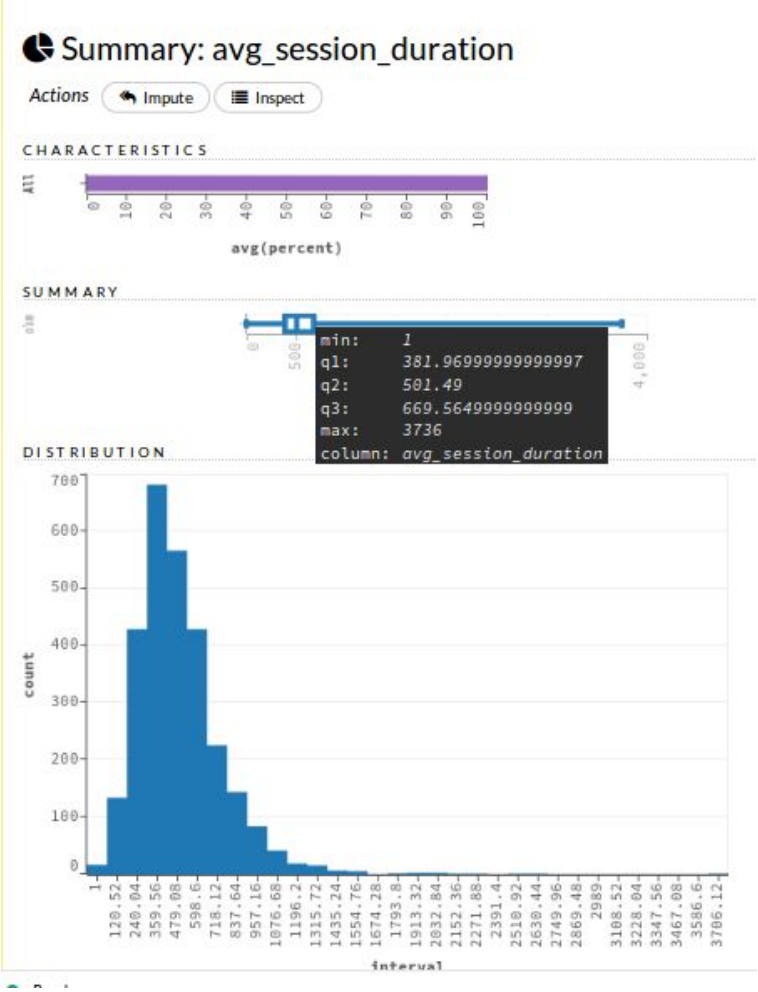
1. maximum number of sessions
2. Max, min, sd, skewness, mean, sum of session_durations
3. Number of overlaps between original broadcast time and actual viewing.
4. Average of the playback speed.
5. Gender
6. Top 5 channels watched by duration
7. Top 5 genre and sub genre watched by duration
8. Top 5 hours of the day ordered by duration of watch
9. Top 5 days of week
10. Top 5 session type and session subtype

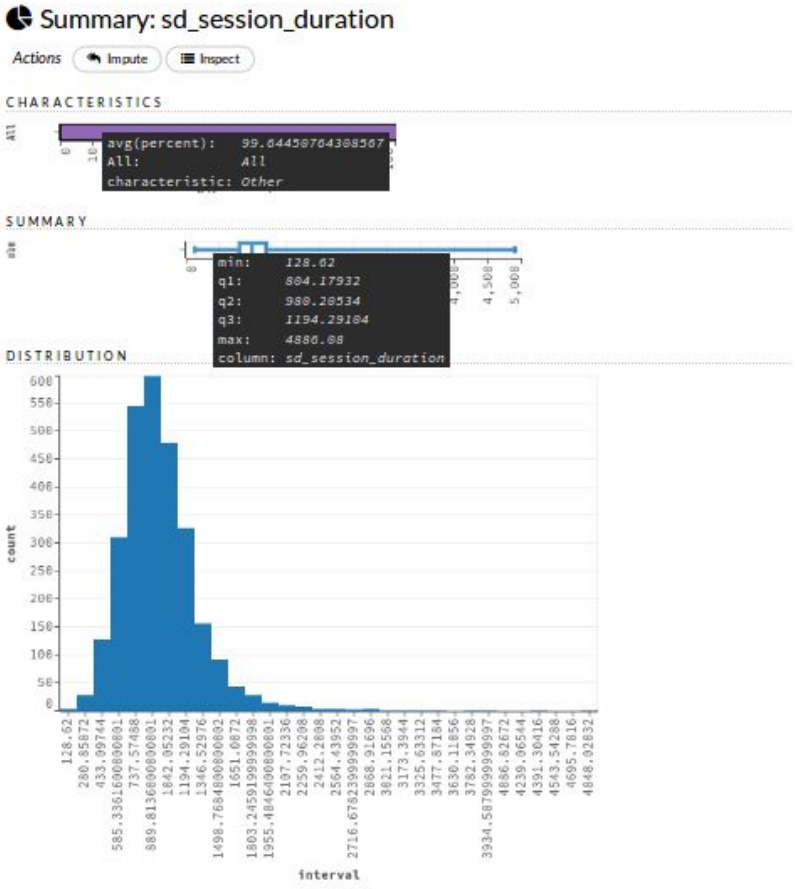
Tools used

Spark 2.2 for data cleaning and aggregation, H2O for analysis and model building

DWELL time

In clickstream/session data analysis "**dwell time**" is an important feature of models based on kaggle competitions. The periodicity/volatility of the dwell times give a strong signal to the final generated model. Below are some of them



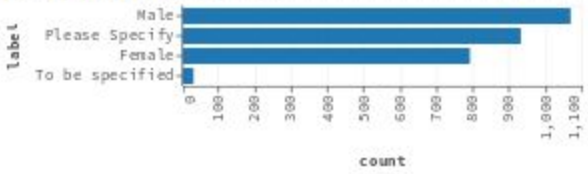

Summary: sd_session_duration
Actions
Impute
Inspect
CHARACTERISTICS
All
avg(percent): 99.64450764308567
All: All
characteristic: Other
SUMMARY
min: 128.62
q1: 804.17932
q2: 980.28534
q3: 1194.29104
max: 4886.08
column: sd_session_duration
DISTRIBUTION
count
interval

Summary: gender

Actions Impute Inspect

CHARACTERISTICS

DOMAIN (MAX 1000 LEVELS)

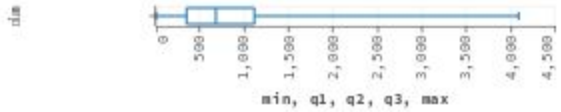


Summary: sum_overlaps

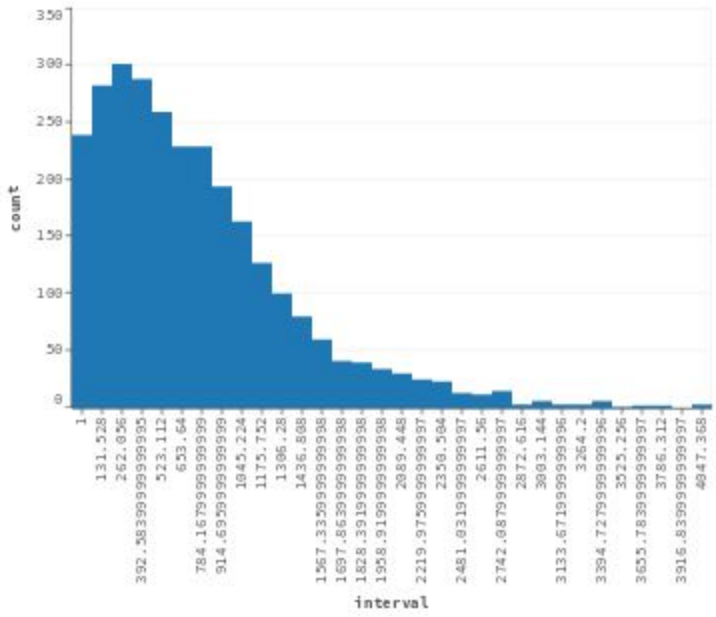
Actions Impute Inspect

CHARACTERISTICS

SUMMARY



DISTRIBUTION

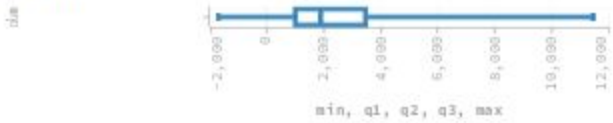


Summary: avg_playback_speed

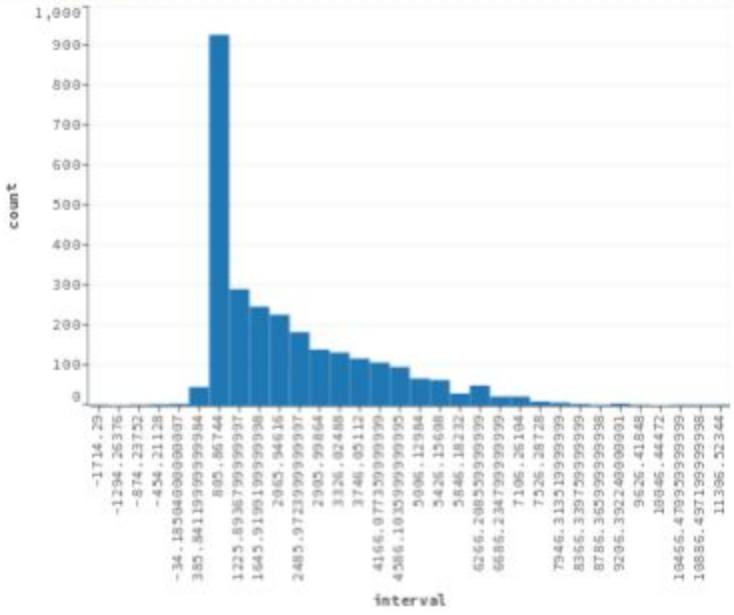
Actions Impute Inspect

CHARACTERISTICS

SUMMARY



DISTRIBUTION



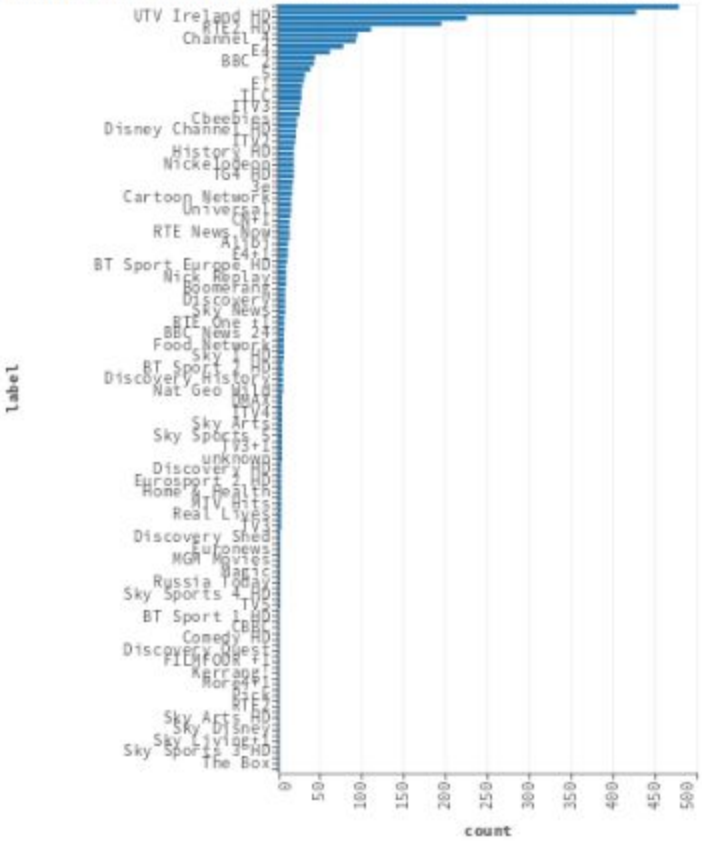
Ready

Summary: channel-2

Actions Impute Inspect

CHARACTERISTICS

DOMAIN (MAX 1000 LEVELS)

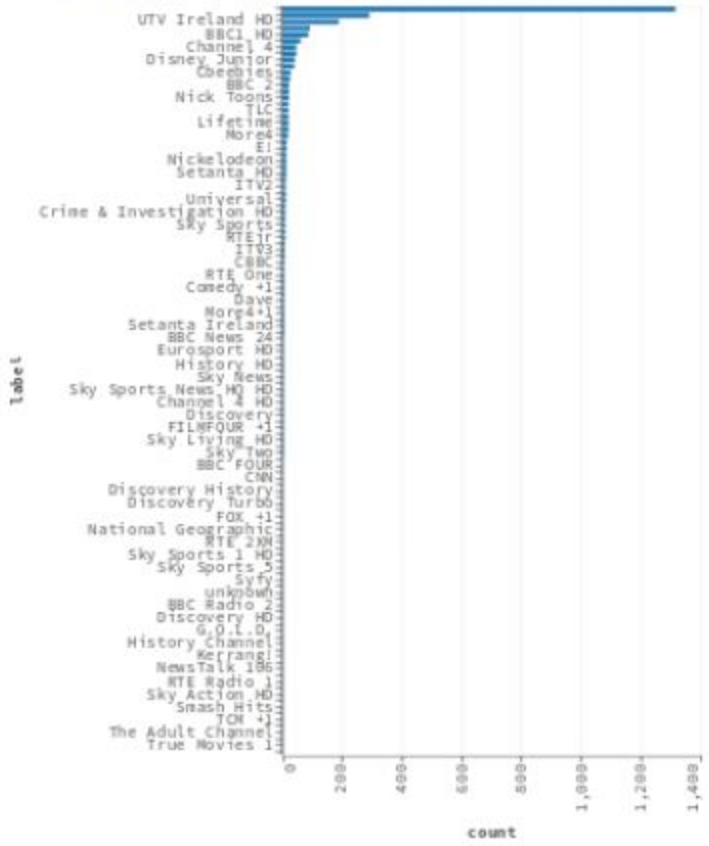


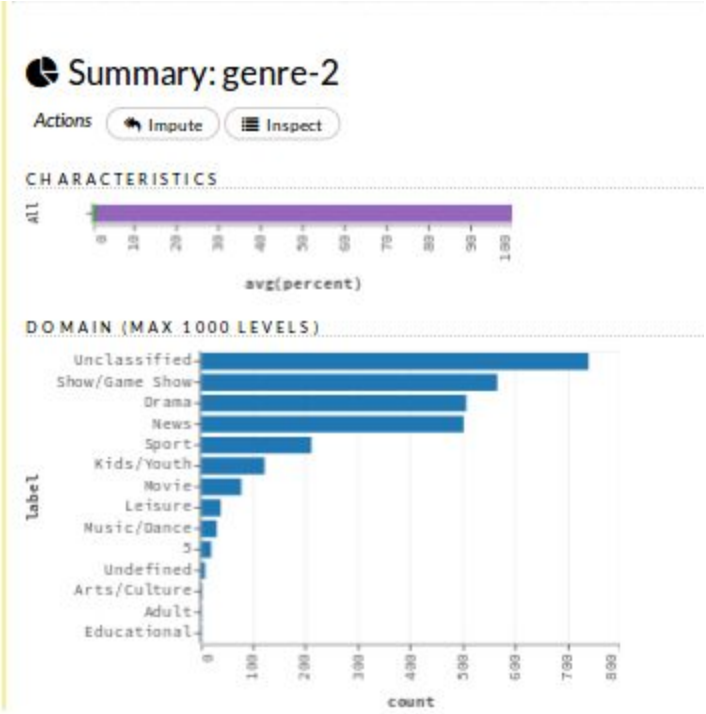
Summary: channel-1

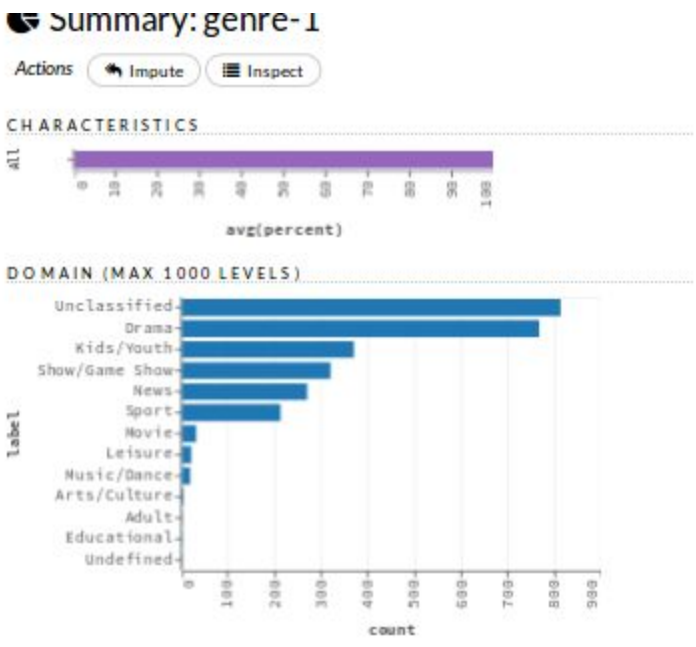
Actions Impute Inspect

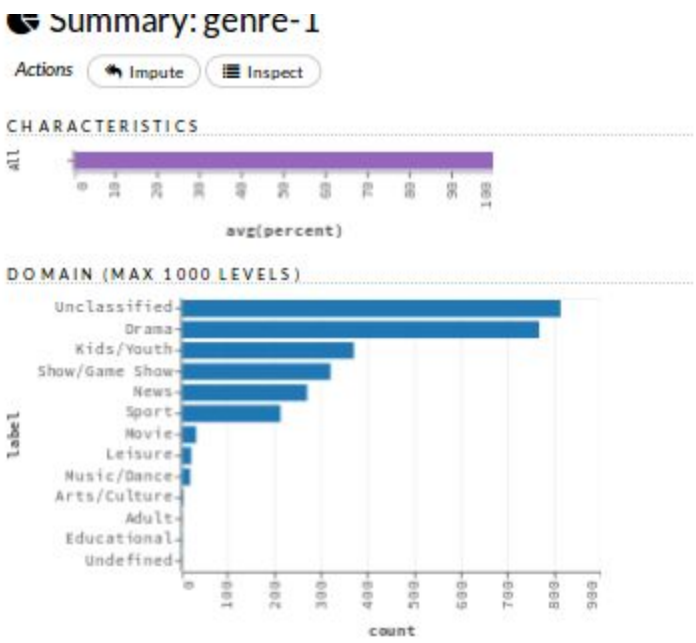
CHARACTERISTICS

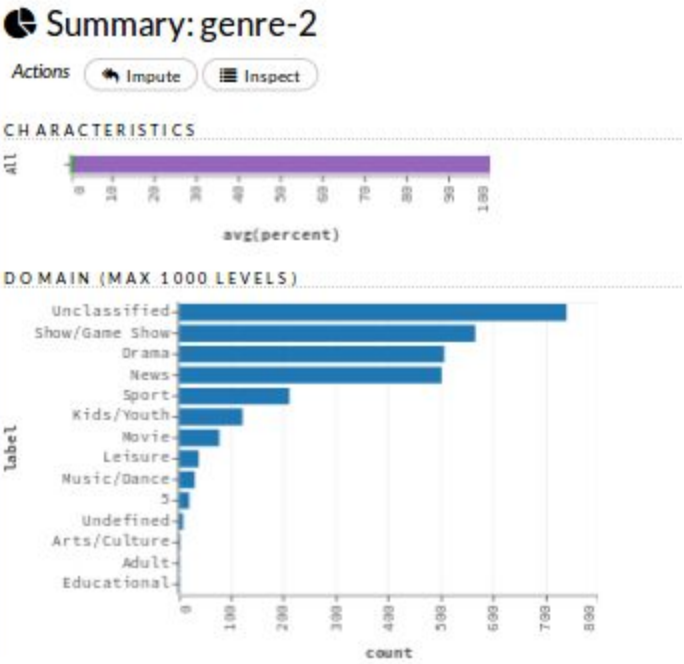
DOMAIN (MAX 1000 LEVELS)

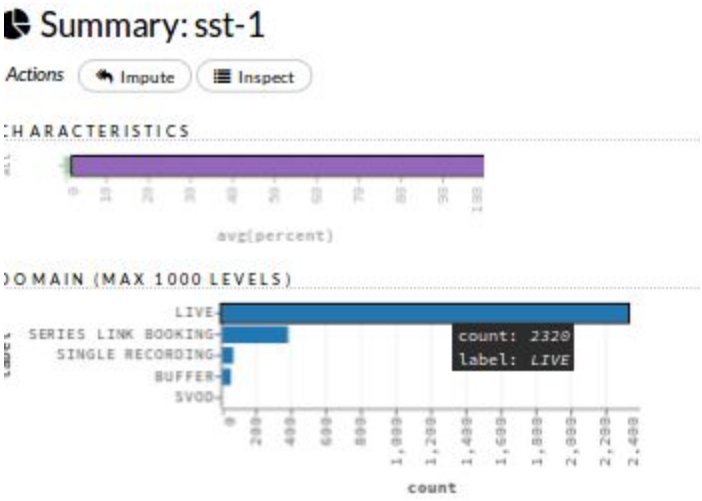


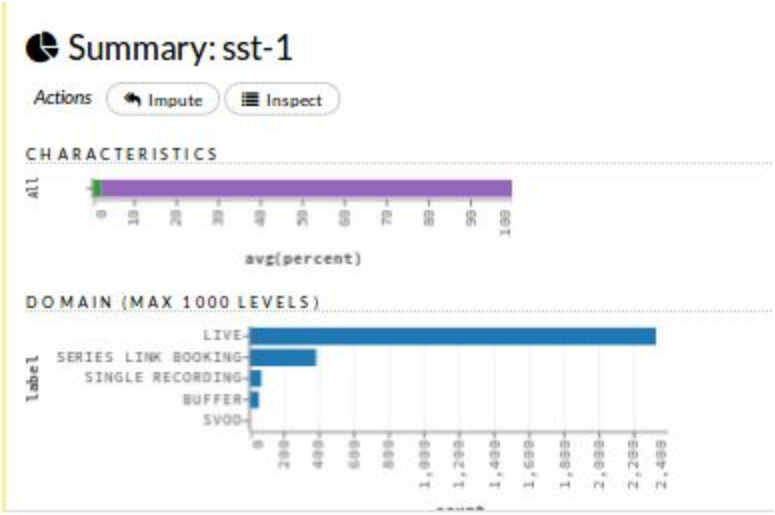

Summary: genre-2
Actions
Impute
Inspect
CHARACTERISTICS
All
0
10
20
30
40
50
60
70
80
90
100
avg(percent)
DOMAIN (MAX 1000 LEVELS)
Unclassified
Show/Game Show
Drama
News
Sport
Kids/Youth
Movie
Leisure
Music/Dance
5
Undefined
Arts/Culture
Adult
Educational
label
0
100
200
300
400
500
600
700
800
count


Summary: genre-1
Actions
Impute
Inspect
CHARACTERISTICS
All
0
10
20
30
40
50
60
70
80
90
100
avg(percent)
DOMAIN (MAX 1000 LEVELS)
Unclassified
Drama
Kids/Youth
Show/Game Show
News
Sport
Movie
Leisure
Music/Dance
Arts/Culture
Adult
Educational
Undefined
label
0
100
200
300
400
500
600
700
800
900
count


Summary: genre-1
Actions
Impute
Inspect
CHARACTERISTICS
All
0
10
20
30
40
50
60
70
80
90
100
avg(percent)
DOMAIN (MAX 1000 LEVELS)
label
Unclassified
Drama
Kids/Youth
Show/Game Show
News
Sport
Movie
Leisure
Music/Dance
Arts/Culture
Adult
Educational
Undefined
0
100
200
300
400
500
600
700
800
900
count


Summary: genre-2
Actions
Impute
Inspect
CHARACTERISTICS
All
0
10
20
30
40
50
60
70
80
90
100
avg(percent)
DOMAIN (MAX 1000 LEVELS)
label
Unclassified
Show/Game Show
Drama
News
Sport
Kids/Youth
Movie
Leisure
Music/Dance
5
Undefined
Arts/Culture
Adult
Educational
0
100
200
300
400
500
600
700
800
count


Summary: sst-1
Actions
Impute
Inspect
CHARACTERISTICS
avg(percent)
DOMAIN (MAX 1000 LEVELS)
LIVE
SERIES LINK BOOKING
SINGLE RECORDING
BUFFER
SVOD
count: 2320
label: LIVE
count

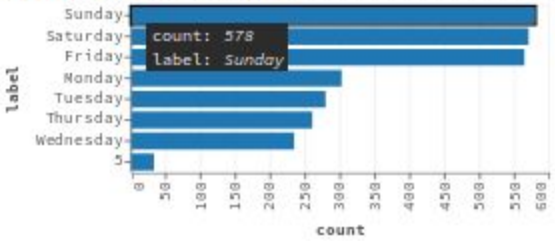

Summary: sst-1
Actions
Impute
Inspect
CHARACTERISTICS
All
avg(percent)
DOMAIN (MAX 1000 LEVELS)
label
LIVE
SERIES LINK BOOKING
SINGLE RECORDING
BUFFER
SVOD

Summary: dow-2

Actions Impute Inspect

CHARACTERISTICS

DOMAIN (MAX 1000 LEVELS)

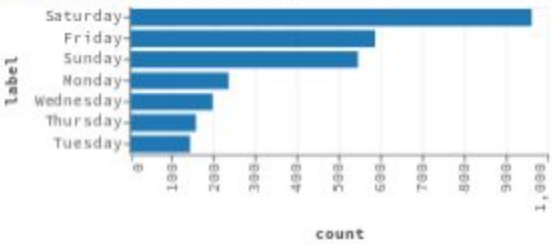


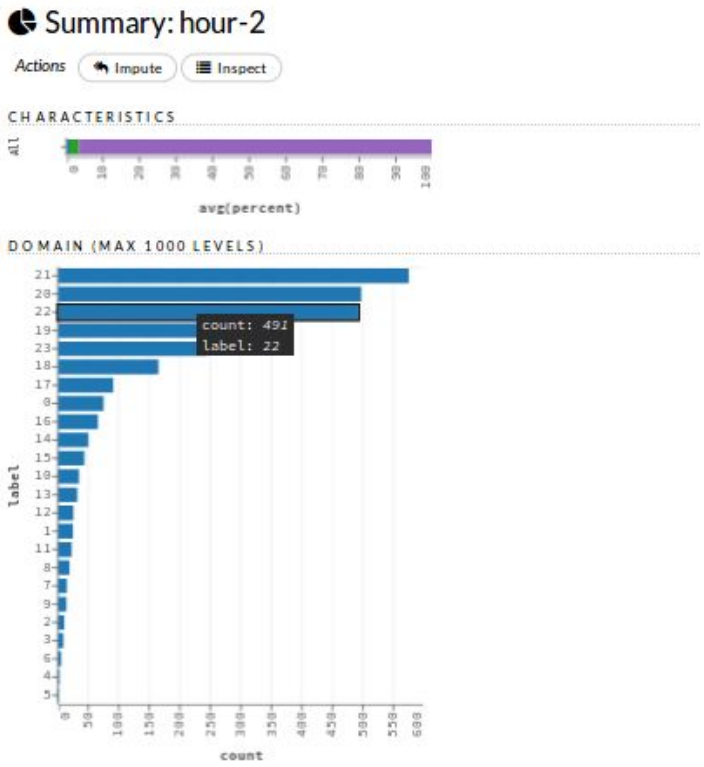
Summary: dow-1

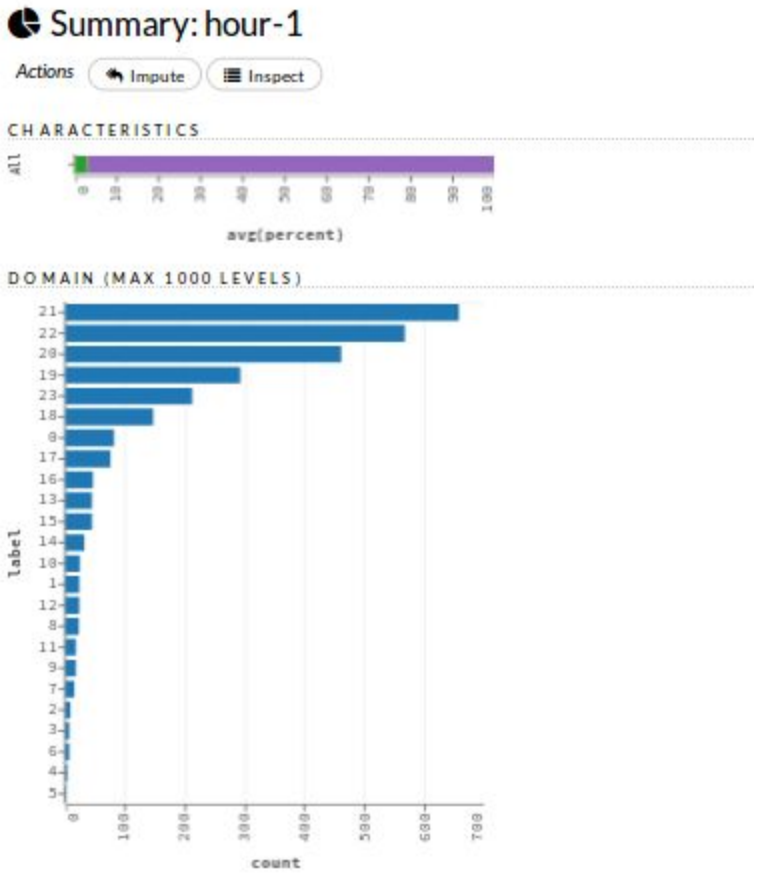
Actions Impute Inspect

CHARACTERISTICS

DOMAIN (MAX 1000 LEVELS)




Summary: hour-2
Actions
Impute
Inspect
CHARACTERISTICS
All
avg(percent)
DOMAIN (MAX 1000 LEVELS)
label
count
count: 491
label: 22


Summary: hour-1
Actions
Impute
Inspect
CHARACTERISTICS
All
0
10
20
30
40
50
60
70
80
90
100
avg(percent)
DOMAIN (MAX 1000 LEVELS)
label
21
22
20
19
23
18
0
17
16
13
15
14
10
1
12
8
11
9
7
2
3
6
4
5
0
100
200
300
400
500
600
700
count

Machine learning

Now that we have a good understanding of the data distributions lets see how ML can help.This **multiclass classification** problem is particularly tricky because it has a **class imbalance** issue (as previously seen) and 30+ features are **categoricals** and **not a lot of data** to draw inference from.

ML model selection

While GBT,DL,NN have worked with multiclass classifications, these require a larger data set and one-hot encoding the categoricals. One hot encoding categoricals lead to dimension explosion leading to the famous curse of dimensionality. Also not all algorithms can handle class imbalance issues without using upsampling or downsampling.

But one-hot encoding also presents two problems that are more particular to tree-based models:

1. The resulting sparsity virtually ensures that continuous variables are assigned higher feature importance.
2. A single level of a categorical variable must meet a very high bar in order to be selected for splitting early in the tree building. This can degrade predictive performance.

Why one-hot encoding is bad for tree based models

Predictive Performance

By one-hot encoding a categorical variable, we create many binary variables, and from the splitting algorithm's point of view, they're all independent. This means a categorical variable is already disadvantaged over continuous variables. But there's a further problem: these binary variables are *sparse*. Imagine our categorical variable has 100 levels, each appearing about as often as the others. The best the algorithm can expect to do by splitting on one of its one-hot encoded dummies is to reduce impurity by ≈1% since each of the dummies will be 'hot' for around 1% of the samples.

The result of all this is that, if we start by one-hot encoding a high-cardinality variable, the tree building algorithm is unlikely to select one of its dummies as the splitting variable near the root of the tree, instead choosing continuous variables. In datasets like the one we created here, that leads to inferior performance

Fewer levels, fewer problems

As we discussed, this problem is especially profound for high-cardinality categorical variables. If the categorical variables have few levels, then the induced sparsity is less severe and the one-hot encoded versions have a chance of competing with the continuous ones. Our aggregation method ensured that we **reduce the cardinality of the categoricals** to mitigate the issue.

Choice of ML model.

A host of models including GBT, DL, linear and stacked modeling using H2o was tried and the final model that gave good results and had best interpretability along with the possibilities of scaling was Distributed Random Forest model of H2O

Why choose random forest

1. Robust to overfitting - controlled by subsampling rate and feature subset strategy.
2. Robust to outliers due to inherent bagging.
3. By using one vs rest strategy, class weights/class thresholds one can handle imbalanced data
4. H2O distributed random forest doesn't use one-hot encoding for categoricals which is a huge advantage especially when dealing with our current skewed categorical feature set.
5. Random forest can be scaled and since viewership data falls under category of clickstream data, having a scalable machine learning model is a necessity.
6. Using stratified sampling and k fold validation to prevent overfitting and handling class imbalances.
7. Missing values are interpreted as containing information (i.e., missing for a reason), rather than missing at random. During tree building, split decisions for every node are found by minimizing the loss function and treating missing values as a separate category that can go either left or right. Missing values are not imputed with mean
8. By tweaking following hyperparameters balance_classes, class_sampling_factors and max_after_balance_size, sample_rate_per_class over/under-sampling can be controlled
9. Large numbers of categoricals are handled very efficiently - there is never any one-hot encoding.

Distributed Random Forest parameters

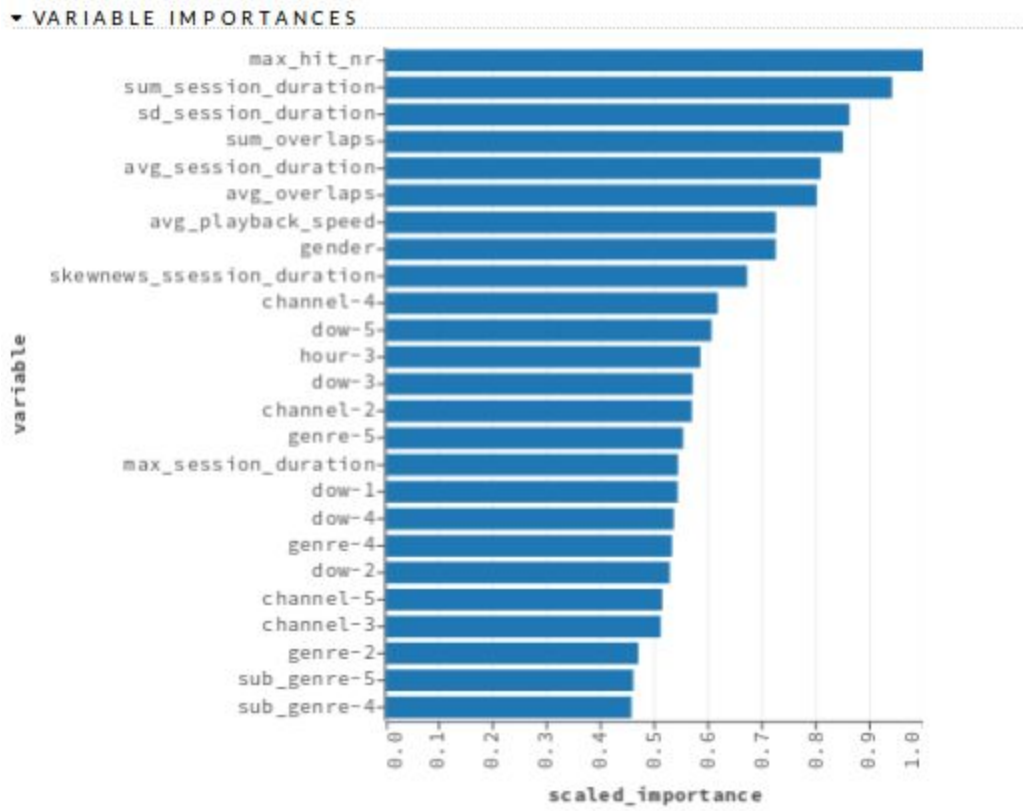
buildModel 'drf',


```
{
  "model_id":"drf-bae79158-70e5-4dda-b8b6-2d3725b05793",
  "training_frame":"frame_0.750",
  "validation_frame":"frame_0.250",
  "nfolds":"3",
  "response_column":"age_band",
  "ignored_columns":[
    "household_id",
    "hour-4",
    "hour-5"
  ],
  "ignore_const_cols":true,
  "ntrees":"500",
  "max_depth":"30",
  "min_rows":1,
  "nbins":20,
  "seed":-1,
  "mtries":-1,
  "sample_rate":0.6320000290870667,
  "score_each_iteration":false,
  "score_tree_interval":0,
  "fold_assignment":"Stratified",
  "balance_classes":true,
  "max_confusion_matrix_size":20,
  "max_hit_ratio_k":0,
  "nbins_top_level":1024,
  "nbins_cats":"6",
  "r2_stopping":1.7976931348623157e+308,
  "stopping_rounds":0,
  "stopping_metric":"AUTO",
  "stopping_tolerance":0.001,
  "max_runtime_secs":0,
```

```
  "checkpoint":"",
  "col_sample_rate_per_tree":1,
  "min_split_improvement":0.00001,
  "histogram_type":"AUTO",
  "categorical_encoding":"AUTO",
  "distribution":"AUTO",
  "custom_metric_func":"",
  "keep_cross_validation_predictions":false,
  "keep_cross_validation_fold_assignment":false,
  "class_sampling_factors":[

  ],
  "max_after_balance_size":5,
  "build_tree_one_node":false,
  "sample_rate_per_class":[

  ],
  "binomial_double_trees":false,
  "col_sample_rate_change_per_level":1,
  "calibrate_model":false
}
```


VARIABLE IMPORTANCES
max_hit_nr
sum_session_duration
sd_session_duration
sum_overlaps
avg_session_duration
avg_overlaps
avg_playback_speed
gender
skewnews_ssession_duration
channel-4
dow-5
hour-3
dow-3
channel-2
genre-5
max_session_duration
dow-1
dow-4
genre-4
dow-2
channel-5
channel-3
genre-2
sub_genre-5
sub_genre-4
variable
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
scaled_importance

▸ TRAINING METRICS - CONFUSION MATRIX ROW LABELS: ACTUAL CLASS; COLUMN LABELS: PREDICTED CLASS

	25to34	35to44	45to54	55to64	greater_than_65	less_than_25	Error	Rate
25to34	1382	0	0	0	6	0	0.0043	6 / 1,388
35to44	0	1391	0	0	0	0	0	0 / 1,391
45to54	0	0	1401	0	0	0	0	0 / 1,401
55to64	0	0	0	1382	0	0	0	0 / 1,382
greater_than_65	0	0	0	0	1388	0	0	0 / 1,388
less_than_25	0	0	0	0	0	1387	0	0 / 1,387
Total	1382	1391	1401	1382	1394	1387	0.0007	6 / 8,337

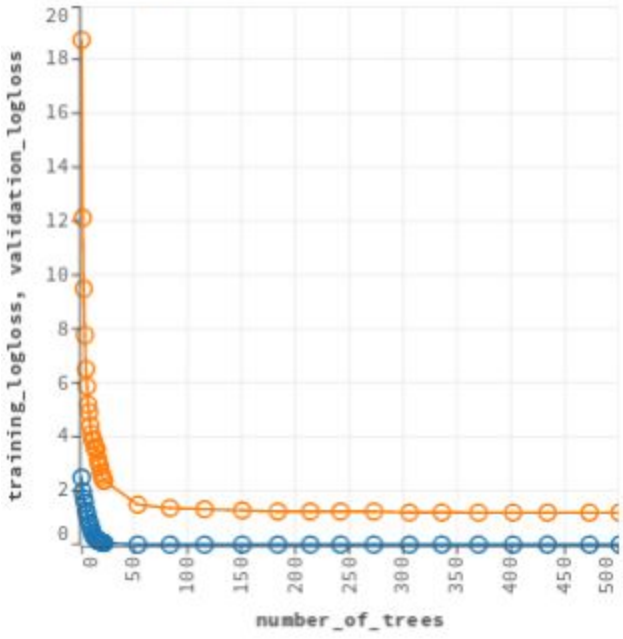
▸ VALIDATION METRICS - CONFUSION MATRIX ROW LABELS: ACTUAL CLASS; COLUMN LABELS: PREDICTED CLASS

	25to34	35to44	45to54	55to64	greater_than_65	less_than_25	Error	Rate
25to34	0	0	0	0	56	0	1.0	56 / 56
35to44	0	0	0	0	67	0	1.0	67 / 67
45to54	0	0	0	0	47	0	1.0	47 / 47
55to64	0	0	0	0	33	0	1.0	33 / 33
greater_than_65	0	0	0	0	480	0	0	0 / 480
less_than_25	0	0	0	0	7	0	1.0	7 / 7
Total	0	0	0	0	690	0	0.3043	210 / 690

▸ CROSS VALIDATION METRICS - CONFUSION MATRIX ROW LABELS: ACTUAL CLASS; COLUMN LABELS: PREDICTED CLASS

	25to34	35to44	45to54	55to64	greater_than_65	less_than_25	Error	Rate
25to34	0	0	0	0	230	0	1.0	230 / 230
35to44	0	0	0	0	246	0	1.0	246 / 246
45to54	0	0	0	0	146	0	1.0	146 / 146
55to64	0	0	0	0	98	0	1.0	98 / 98
greater_than_65	1	0	0	0	1387	0	0.0007	1 / 1,388
less_than_25	0	0	0	0	15	0	1.0	15 / 15
Total	1	0	0	0	2122	0	0.3467	736 / 2,123

▾ SCORING HISTORY - LOGLOSS



▾ OUTPUT - VARIABLE IMPORTANCES

variable	relative_importance	scaled_importance	percentage
max_hit_nr	108933.8672	1.0	0.0490
sum_session_duration	102373.4609	0.9398	0.0460
sd_session_duration	93737.8672	0.8605	0.0421
sum_overlaps	92429.0156	0.8485	0.0415
avg_session_duration	87954.9141	0.8074	0.0395
avg_overlaps	87158.9844	0.8001	0.0392
avg_playback_speed	78854.2500	0.7239	0.0354
gender	78820.0781	0.7236	0.0354
skewnews_ssession_duration	73049.5781	0.6706	0.0328
channel-4	67087.9609	0.6159	0.0302
dow-5	65821.7578	0.6042	0.0296
hour-3	63614.0234	0.5840	0.0286
dow-3	62050.2734	0.5696	0.0279
channel-2	61869.3477	0.5680	0.0278
genre-5	60104.4688	0.5518	0.0270
max_session_duration	59048.0273	0.5421	0.0265
dow-1	58988.8047	0.5415	0.0265
dow-4	58199.4297	0.5343	0.0262
genre-4	57834.2891	0.5309	0.0260
dow-2	57371.0742	0.5267	0.0258
channel-5	55874.4492	0.5129	0.0251
channel-3	55538.6914	0.5098	0.0250
genre-2	51034.3008	0.4685	0.0229
sub_genre-5	50010.8281	0.4591	0.0225
sub_genre-4	49650.1172	0.4558	0.0223
genre-1	48363.7617	0.4440	0.0217
genre-3	47161.3633	0.4329	0.0212
hour-1	46602.9063	0.4278	0.0209
sub_genre-1	44964.9961	0.4128	0.0202
hour-2	44692.7305	0.4103	0.0201
sub_genre-3	43923.8047	0.4032	0.0197
sst-2	43417.8242	0.3986	0.0195

Ready

▾ OUTPUT - CROSS-VALIDATION METRICS SUMMARY

	mean	sd	cv_1_valid	cv_2_valid	cv_3_valid
accuracy	0.6533153	0.01598757	0.68333334	0.62876713	0.64784545
err	0.3466847	0.01598757	0.31666666	0.37123287	0.35215452
err_count	245.33333	13.093679	228.0	271.0	237.0
logloss	1.3718519	0.10306299	1.1665109	1.4900868	1.458958
max_per_class_error	1.0	0.0	1.0	1.0	1.0
mean_per_class_accuracy	0.1665459	1.20772944E-4	0.16666667	0.16630435	0.16666667
mean_per_class_error	0.83345413	1.20772944E-4	0.8333333	0.83369565	0.8333333
mse	0.32556966	0.014462095	0.29825106	0.34745863	0.33099928
r2	0.85102576	0.0022052918	0.85533446	0.8496877	0.8480551
rmse	0.5703018	0.01275872	0.5461237	0.5894562	0.57532537

▾ OUTPUT - CROSS_VALIDATION_METRICS

model	drf-bae79158-70e5-4dda-b8b6-2d3725b05793
model_checksum	7365975645156546560
frame	frame_0.750
frame_checksum	6283406801818493952
description	3-fold cross-validation on training data (Metrics computed for combined holdout predictions)
model_category	Multinomial
scoring_time	1522696505423
predictions	·
MSE	0.325553
RMSE	0.570572
nobs	2123
custom_metric_name	·
custom_metric_value	0
r2	0.851180
logloss	1.370480
mean_per_class_error	0.833453

▾ MODEL PARAMETERS

Show all parameters

Parameter	Value	Description
model_id	drf-bae79158-70e5-4dda-b8b6-2d3725b05793	Destination id for this model; auto-generated if not specified.
training_frame	frame_0.750	Id of the training data frame.
validation_frame	frame_0.250	Id of the validation data frame.
nfolds	3	Number of folds for K-fold cross-validation (0 to disable or >= 2).
fold_assignment	Stratified	Cross-validation fold assignment scheme, if fold_column is not specified. The 'Stratified' option will stratify the folds based on the response variable, for classification problems.
response_column	age_band	Response variable column.
ignored_columns	household_id, hour-4, hour-5	Names of columns to ignore for training.
balance_classes	true	Balance training data class counts via over/under-sampling (for imbalanced data).
ntrees	500	Number of trees.
max_depth	30	Maximum tree depth.
nbins_cats	6	For categorical columns (factors), build a histogram of this many bins, then split at the best point. Higher values can lead to more overfitting.
seed	5711323659330028000	Seed for pseudo random number generator (if applicable)
distribution	multinomial	Distribution function
custom_metric_func		Reference to custom evaluation function, format: `language:keyName=funcName`

▾ SCORING HISTORY - LOGLOSS

What if we remove the dummy values of date of birth corresponding to 1900-01-01?

We get more predictions on other classes simply because the huge data skew is removed.

▾ TRAINING METRICS - CONFUSION MATRIX ROW LABELS: ACTUAL CLASS; COLUMN LABELS: PREDICTED CLASS

	25to34	35to44	45to54	55to64	greater_than_65	less_than_25	Error	Rate
25to34	149	86	2	0	0	0	0.3713	88 / 237
35to44	72	157	7	0	0	0	0.3347	79 / 236
45to54	22	31	176	0	0	0	0.2314	53 / 229
55to64	0	0	0	237	0	0	0	0 / 237
greater_than_65	0	0	0	0	230	0	0	0 / 230
less_than_25	0	0	0	0	0	238	0	0 / 238
Total	243	274	185	237	230	238	0.1564	220 / 1,407

▾ VALIDATION METRICS - CONFUSION MATRIX ROW LABELS: ACTUAL CLASS; COLUMN LABELS: PREDICTED CLASS

	25to34	35to44	45to54	55to64	greater_than_65	less_than_25	Error	Rate
25to34	34	38	0	0	1	0	0.5342	39 / 73
35to44	21	56	0	0	0	0	0.2727	21 / 77
45to54	18	31	3	0	0	0	0.9423	49 / 52
55to64	9	21	3	0	0	0	1.0	33 / 33
greater_than_65	4	16	2	0	0	0	1.0	22 / 22
less_than_25	6	1	0	0	0	0	1.0	7 / 7
Total	92	163	8	0	1	0	0.6477	171 / 264

▾ CROSS VALIDATION METRICS - CONFUSION MATRIX ROW LABELS: ACTUAL CLASS; COLUMN LABELS: PREDICTED CLASS

	25to34	35to44	45to54	55to64	greater_than_65	less_than_25	Error	Rate
25to34	103	109	1	0	0	0	0.5164	110 / 213
35to44	73	158	4	1	0	0	0.3305	78 / 236
45to54	46	88	7	0	0	0	0.9504	134 / 141
55to64	20	71	7	0	0	0	1.0	98 / 98
greater_than_65	18	36	10	1	0	0	1.0	65 / 65
less_than_25	10	4	1	0	0	0	1.0	15 / 15
Total	270	466	30	2	0	0	0.6510	500 / 768