Predicting YouTube Success

Jessica Ling, Edmund Doerksen, William Qi

Background:

YouTube is an ever-growing platform for:

- Influencing and advertising
- Artistic expression
- Social commentary
- Making money

With financial earnings and social influence directly tied towards viewership, there is a strong incentive to **build a strong subscriber base** on YouTube.

Our Objective:

To develop a predictive model of channel success based on actionable channel parameters.

Inputs:

- Engagement statistics
- Weekly upload schedule
- Optional personalizations
- Content type

Outputs:

- Predicted Subscriber Count

Our Dataset:

1.10 million channels

- Randomly sampled from ~50 million channels on YouTube
- Represents channels up to 244 million subscribers

Sourced from Kaggle, gathered in 2024

- 16 Channel Parameters

source

Parameters

Channel Intrinsics:

- channel_id (str)
- channel_link (url)
- join_date (date)

Engagement Stats:

- subscriber_count (int)
- total_views (float)
- total_videos (float)
- monthly stats:
  - mean_views (float)
  - median_views (float)
  - std_views (float)
- videos_per_week (float)

Personalizable data:

- channel_name (str)
- banner_link (url)
- description (str)
- keywords (str)
- avatar (url)
- country (str)

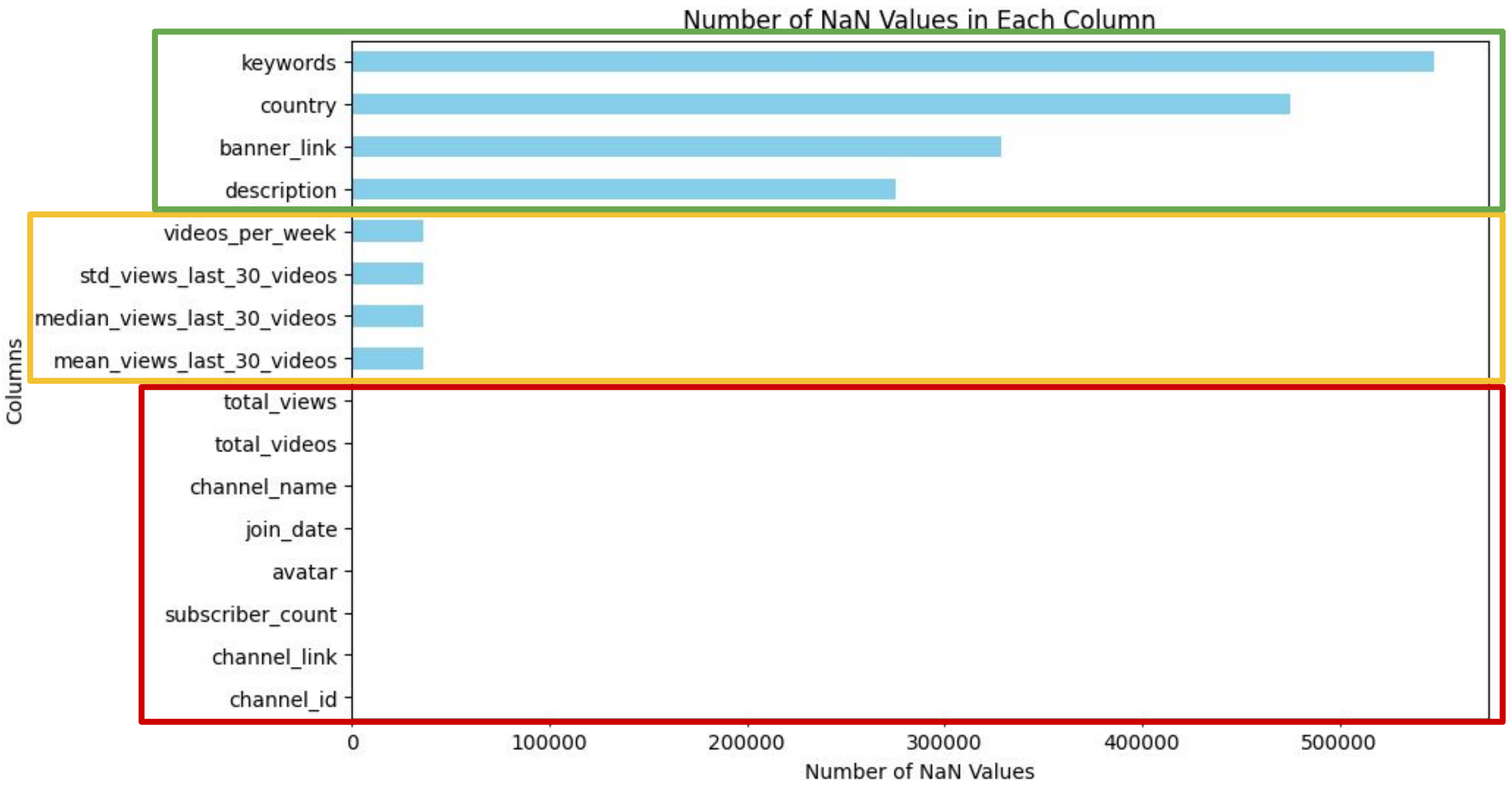
EDA

Important Parameters

- **subscriber_count**: data label

- **description, keywords, banner, avatar**: important customizations for channel appeal
- **engagement stats:** measure video quality and output rate

Handling NaNs

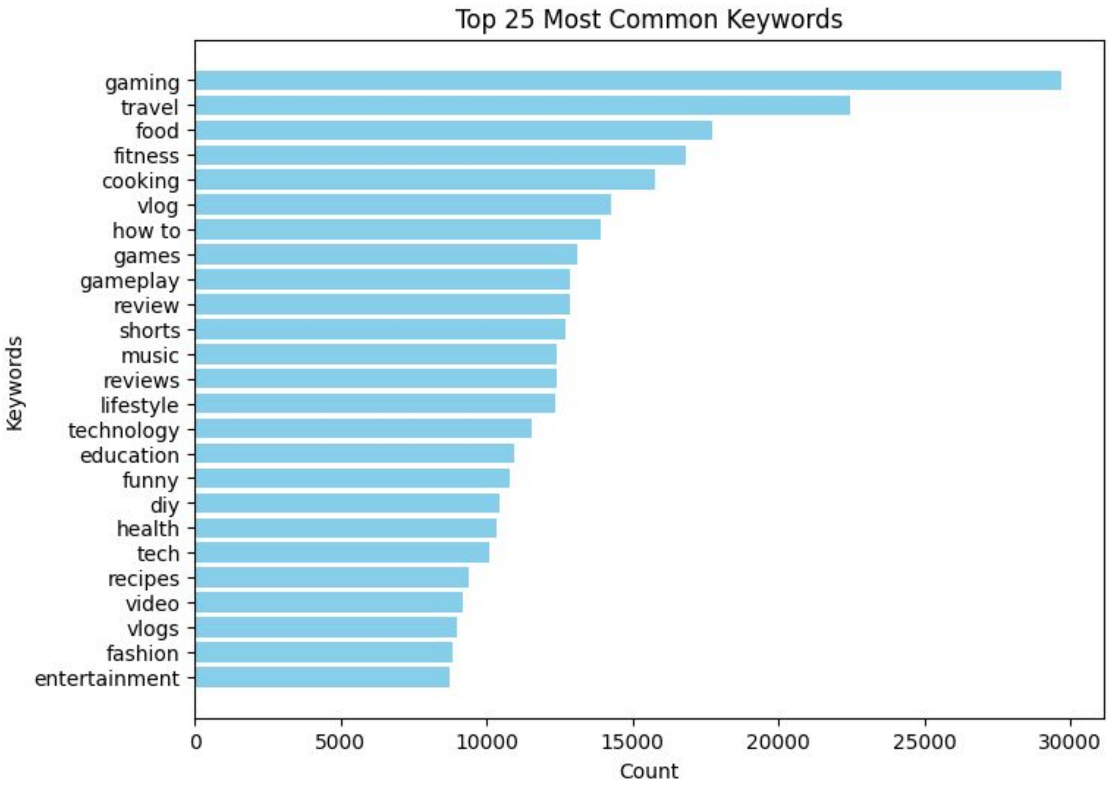


Binary Encoding

Fill NaNs with zeros

Drop NaNs

Feature Engineering



Gaming

Lifestyle

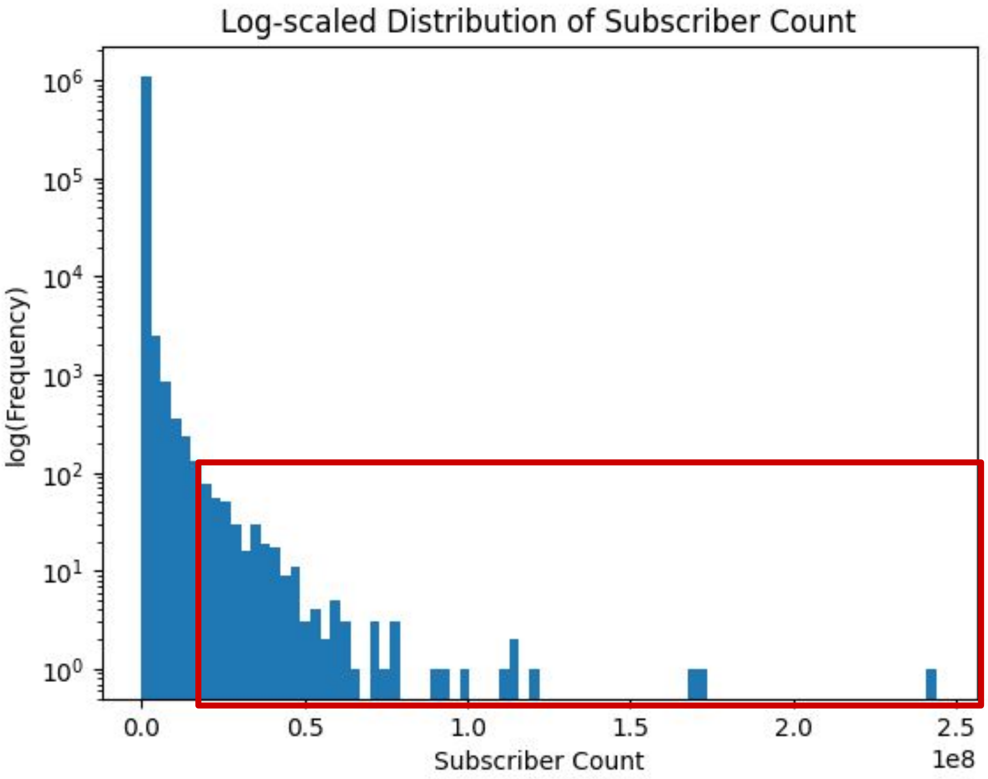
Guides & Tutorials

Technology

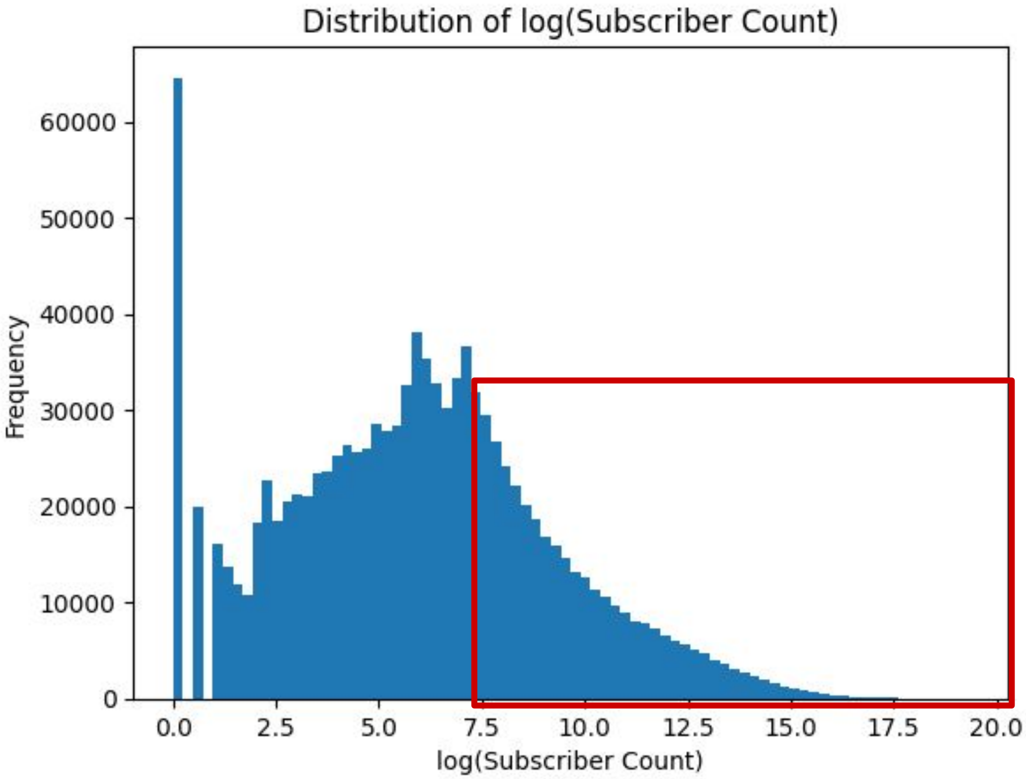
News & Finance

Entertainment

Anticipated Challenges



Anticipated Challenges



Correlation Matrix

Raw data

Binary Encoding

Computation

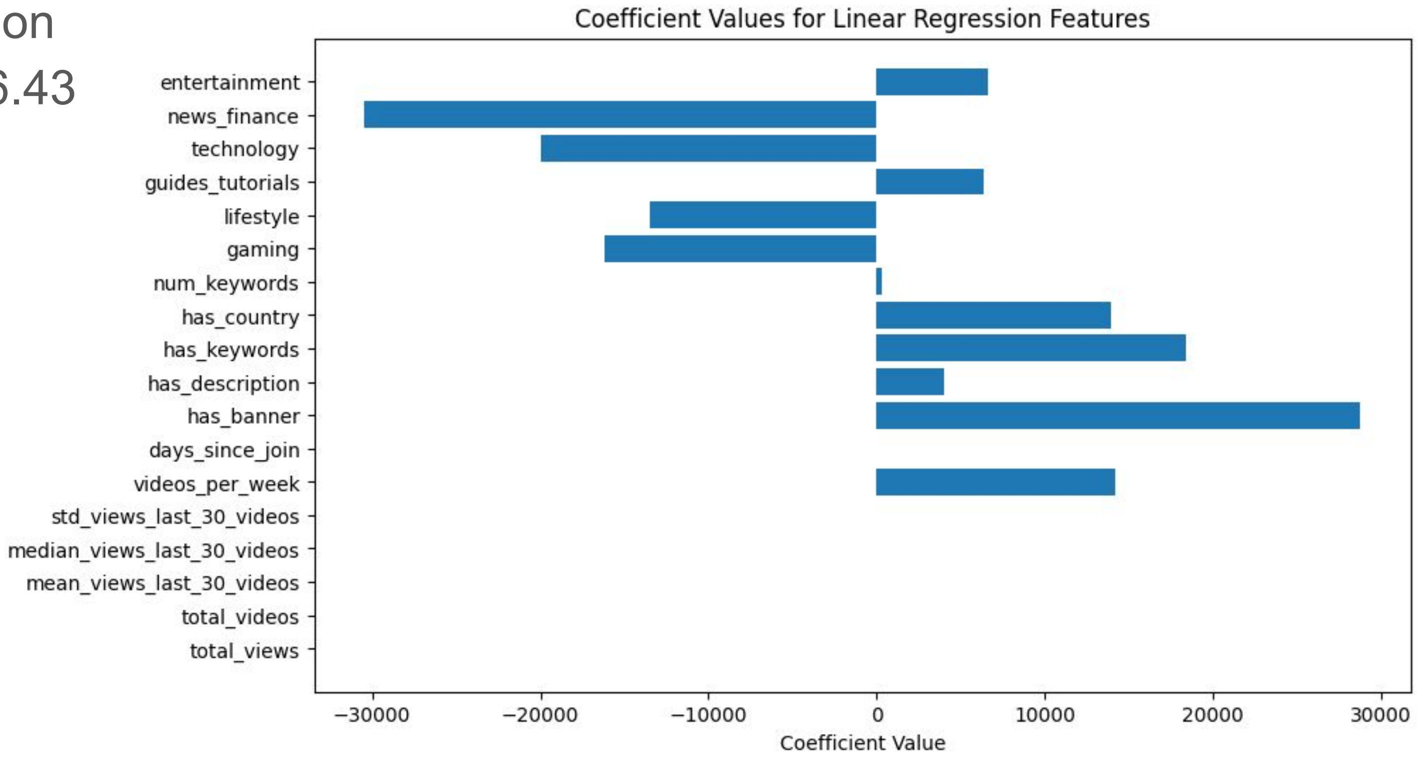
Keyword Category Encodings

	subscriber_count	total_views	total_videos	mean_views_last_30_videos	median_views_last_30_videos	std_views_last_30_videos	videos_per_week	has_banner	has_description	has_keywords	has_country	num_keywords	days_since_join	gaming	lifestyle	guides_tutorials	technology	news_finance	entertainment
subscriber_count	1.0	0.78	0.15	0.11	0.07	0.12	0.11	0.07	0.05	0.08	0.07	0.07	0.04	0.01	0.01	0.02	0.0	-0.0	0.04
total_views	0.78	1.0	0.21	0.08	0.06	0.09	0.1	0.04	0.04	0.05	0.05	0.05	0.03	0.01	0.0	0.01	-0.0	-0.0	0.04
total_videos	0.15	0.21	1.0	-0.0	-0.0	-0.0	0.19	0.03	0.03	0.05	0.04	0.05	0.04	0.02	-0.0	0.01	0.01	0.03	0.02
mean_views_last_30_videos	0.11	0.08	-0.0	1.0	0.78	0.74	-0.01	0.0	-0.0	0.01	0.0	0.0	0.01	-0.0	-0.0	-0.0	-0.0	-0.0	0.01
median_views_last_30_videos	0.07	0.06	-0.0	0.78	1.0	0.2	-0.0	0.0	-0.0	0.0	0.0	-0.0	0.01	-0.0	-0.0	-0.0	-0.0	-0.0	0.0
std_views_last_30_videos	0.12	0.09	-0.0	0.74	0.2	1.0	-0.01	0.01	0.0	0.01	0.0	0.0	0.02	-0.0	-0.0	-0.0	-0.0	-0.0	0.0
videos_per_week	0.11	0.1	0.19	-0.01	-0.0	-0.01	1.0	0.12	0.13	0.14	0.17	0.17	-0.07	0.06	-0.01	0.02	0.02	0.04	0.04
has_banner	0.07	0.04	0.03	0.0	0.0	0.01	0.12	1.0	0.42	0.33	0.37	0.25	-0.05	0.09	0.11	0.1	0.06	0.07	0.12
has_description	0.05	0.04	0.03	-0.0	-0.0	0.0	0.13	0.42	1.0	0.36	0.39	0.28	-0.12	0.1	0.12	0.11	0.07	0.08	0.13
has_keywords	0.08	0.05	0.05	0.01	0.0	0.01	0.14	0.33	0.36	1.0	0.57	0.67	0.04	0.22	0.31	0.25	0.18	0.21	0.33
has_country	0.07	0.05	0.04	0.0	0.0	0.0	0.17	0.37	0.39	0.57	1.0	0.45	-0.12	0.14	0.19	0.17	0.11	0.13	0.2
num_keywords	0.07	0.05	0.05	0.0	-0.0	0.0	0.17	0.25	0.28	0.67	0.45	1.0	-0.05	0.22	0.32	0.31	0.23	0.21	0.39
days_since_join	0.04	0.03	0.04	0.01	0.01	0.02	-0.07	-0.05	-0.12	0.04	-0.12	-0.05	1.0	0.04	-0.01	0.0	0.01	-0.02	0.02
gaming	0.01	0.01	0.02	-0.0	-0.0	-0.0	0.06	0.09	0.1	0.22	0.14	0.22	0.04	1.0	-0.03	0.13	0.15	-0.01	0.23
lifestyle	0.01	0.0	-0.0	-0.0	-0.0	-0.0	-0.01	0.11	0.12	0.31	0.19	0.32	-0.01	-0.03	1.0	0.23	0.01	0.02	0.2
guides_tutorials	0.02	0.01	0.01	-0.0	-0.0	-0.0	0.02	0.1	0.11	0.25	0.17	0.31	0.0	0.13	0.23	1.0	0.13	0.05	0.19
technology	0.0	-0.0	0.01	-0.0	-0.0	-0.0	0.02	0.06	0.07	0.18	0.11	0.23	0.01	0.15	0.01	0.13	1.0	0.05	0.17
news_finance	-0.0	-0.0	0.03	-0.0	-0.0	-0.0	0.04	0.07	0.08	0.21	0.13	0.21	-0.02	-0.01	0.02	0.05	0.05	1.0	0.06
entertainment	0.04	0.04	0.02	0.01	0.0	0.0	0.04	0.12	0.13	0.33	0.2	0.39	0.02	0.23	0.2	0.19	0.17	0.06	1.0

Modeling

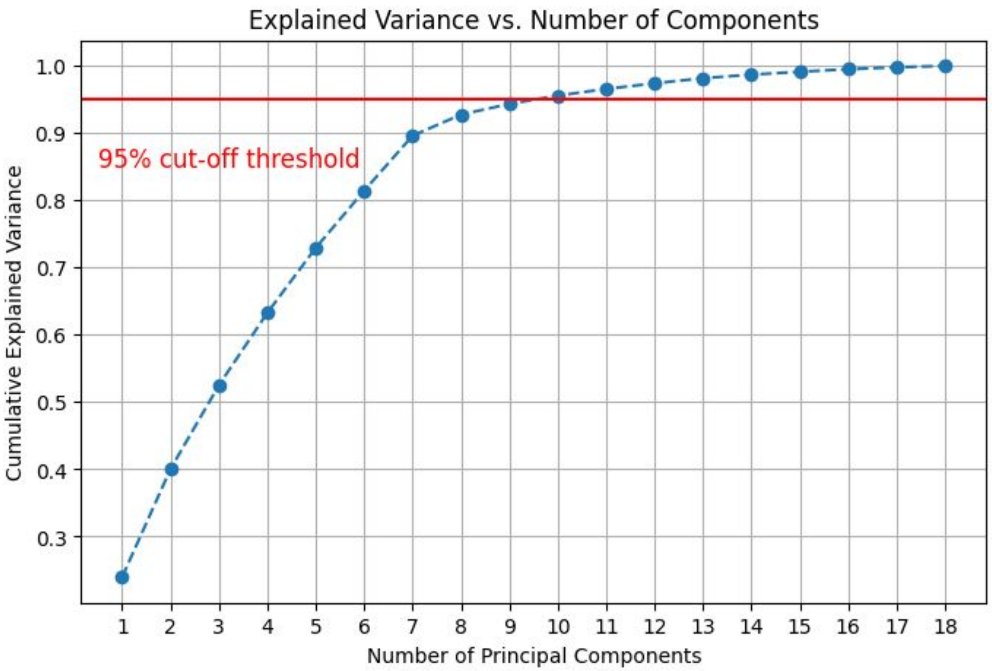
Baseline Model

- Linear regression
- RMSE: 455656.43
- R^2: 0.6731



PCA + Scaling

- PCA: Large number of features, with many highly correlated (e.g. mean_views_last_30_videos and median_views_last_30_videos)
- Scaling required for regularized regression and gradient boosting



Linear Regression Revisited

- Linear regression with reduced and normalized features

	Train R^2	Test R^2
Original labels	0.6904	0.6694
Log-scaled labels	0.3505	0.3517

Ridge Regression

- Varied the weight of the normalization term (α)
- Large α values required to significantly alter the regression model

α	Test Error (regular labels)	Test Error (log labels)
0.1	0.6694	**0.3517**
1	0.6694	0.3517
10	0.6694	0.3505
100	0.6694	0.3516
1000	0.6696	0.3516
10000	0.6716	0.3513
100000	**0.6816**	0.3371
1000000	0.5384	0.2193

Elastic Net Regression

- Combination of Lasso and Ridge
- Test R^2 generally increased as weight of Lasso regularization term increased relative to weight of Ridge regularization term
- Average Train R^2: 0.5931, Average Test R^2: 0.6065
- Table: effect of L1 ratio and α on Test R^2

	L1 ratio = 0.2	L1 ratio = 0.4	L1 ratio = 0.6	L1 ratio = 0.8
α=0.1	0.6800	0.6784	0.6761	0.6731
α=0.2	0.6816	**0.6817**	0.6800	0.6761
α=0.5	0.6588	0.6716	0.6800	0.6811
α=1.0	0.5936	0.6273	0.6588	0.6800
α=2.0	0.4750	0.5297	0.5935	0.6588
α=5.0	0.2866	0.3442	0.4292	0.5606

Random Forest

- Random Forest regression- grid search over hyperparameters for tree depth and number of trees
- Original features, original labels
- Average Train R^2: 0.8427, Average Test R^2: 0.7541
- Table: Effect of n_estimators and max_depth on Test R^2

	max_depth=3	max_depth=5	max_depth=8	max_depth=12
n_estimators=10	0.7347	0.7583	0.7484	0.7316
n_estimators=20	0.7391	0.7486	**0.7773**	0.7622
n_estimators=50	0.7421	0.7466	0.7724	0.7569
n_estimators=100	0.7461	0.7607	0.7673	0.7740

Gradient Boosting

- Used grid search over max depth of 4, 6, and 8, and learning rate of 0.01, 0.1, and 0.2.
- Performance on log-transformed data was significantly better than on original subscriber count, supporting our initial hypothesis.
- Train R^2 of 0.84 and test R^2 of 0.83 – no overfitting, no underfitting (!!)
- Best model out of the ones we tried – able to model nonlinearities, boosting was able to reduce the bias

	Train R^2	Test R^2
Original labels	0.56	0.44
Log-scaled labels	0.84	0.83

Performance With/out Outliers

Metric: Test R^2	With Outliers		Without Outliers	
	Original Labels	**Log(Labels)**	**Original Labels**	**Log(Labels)**
Linear Regression 2	0.6694	0.3517	0.6071	0.3626
Ridge Regression	0.6816	0.3517	0.6071	0.3626
Elastic Net Regression	0.6817	0.3400	0.6063	0.3509
Random Forest	0.7630		0.78	
Gradient Boosting	0.44	0.83	0.68	0.83

Implications/Insights

- The factors that influence subscriber count seem to be quite complex. If you are an aspiring YouTuber, there is no one-size-fits-all formula!
- Ensemble methods proved to be an important player, as we saw random forest and gradient boosting significantly outperform linear models.
- Big takeaway – don’t underestimate bagging and boosting! These methods can rival neural networks, especially for tabular data. They are also more interpretable and less computationally-expensive.
- The best model was gradient boosting, with a score of 0.83. This means that the proportion of the variation in the subscriber count that is predictable from the features is quite good.

Challenges/Limitations

- Dataset skew: subscriber counts range from 0 to 244 million, but 35% of channels have less than 100 subscribers and 94% have less than 100,000
- Data perhaps not granular enough. Additionally, data was very sparse.
- RMSE not as indicative of performance when models run using log of subscriber count because of extreme outliers
    - mean absolute error (MAE) a more useful metric instead due to punishing outliers less
- Processing limitations: runtime limits in Colab cutting long computations short

Future Work

Current model improvements

- Sentiment analysis of YouTube channel description
- More sophisticated balancing techniques
    - Webscraping top channels to augment high-subscription channels
    - SMOTE-NC
    - Further analysis on outlier effects
- Using deep neural networks to increase model complexity even more
- Joining on different datasets to get richer feature sets

Other Predictions

- Prediction of YouTube channel category given its description
- Prediction of channel age given its description

Thank you!