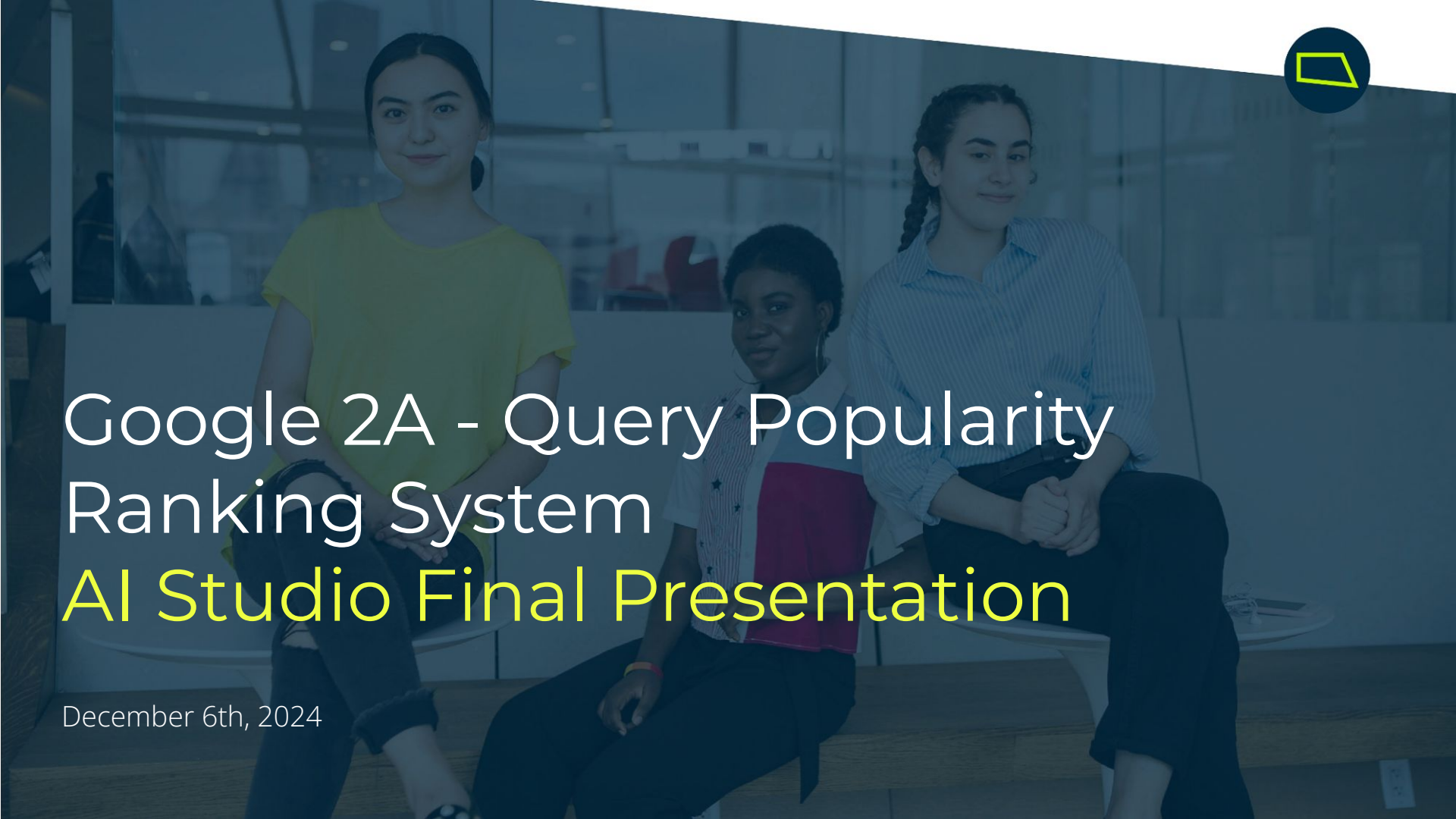


Google 2A - Query Popularity Ranking System

AI Studio Final Presentation

December 6th, 2024

Introductions

Meet Our Team!

Aimen Moten
DePauw University

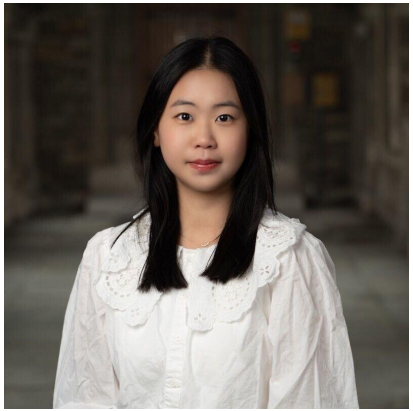
Ananya Jakilati
UC Santa Cruz

Nishan Lama
UC Santa Cruz

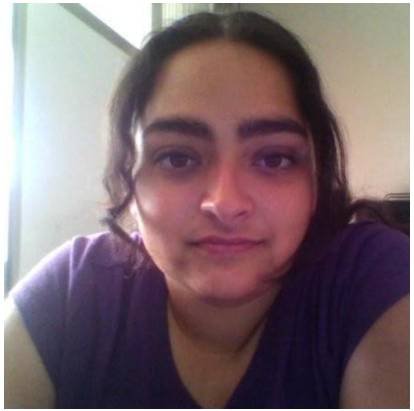
Diego Carrillo
New Mexico State University

Roma Solapurkar
UC Santa Cruz

Our AI Studio TA and Challenge Advisors



Esther Li
AI Studio TA



Ashley Prasad
Challenge Advisor

Kaung Khin
Challenge Advisor

Presentation Agenda

01 Problem/Goal

02 Resources

03 Our Approaches

04 Data Understanding

05 Data Visualization

06 Building Model

Our Problem

The Problem

Users struggle to find information and trending topics quickly. Predictive models improve search relevance, enabling faster discovery and personalized experiences.

Our Goals

- Understand machine learning techniques to predict search trends.
- Build and deploy a predictive model for Google to anticipate trends.
- Optimize model accuracy to enhance user experience and engagement.
- Use predictions to improve content discovery and satisfaction.

Business Impact

Predicting search trends can help with both infrastructure costs and quality increases. As an example:

- Infrastructure: Cache top search query results
  - **Proactive Caching**: Cache results for top anticipated queries to improve response times.
  - **Reduced Server Demand**: Minimizes processing load during peak search periods.
  - **Optimized Resource Allocation**: Frees up resources to handle additional queries, enhancing system efficiency.
  - **Cost Savings**: Lowers infrastructure costs by decreasing redundant processing for high-frequency searches.
  - **Improved User Experience**: Ensures stable and fast search performance, even during surges in query volume.

Resources We Leveraged

01 Machine Learning Foundations Course

02 Python Libraries & documentation

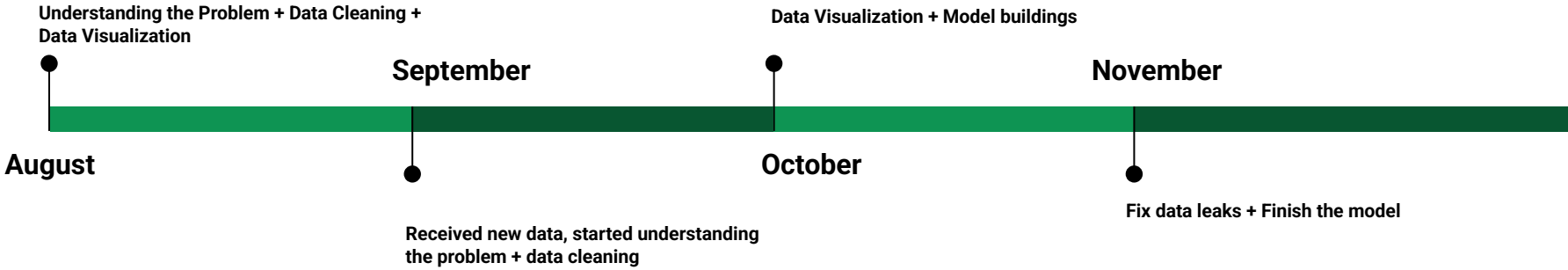
03 Chat GPT

04 Youtube

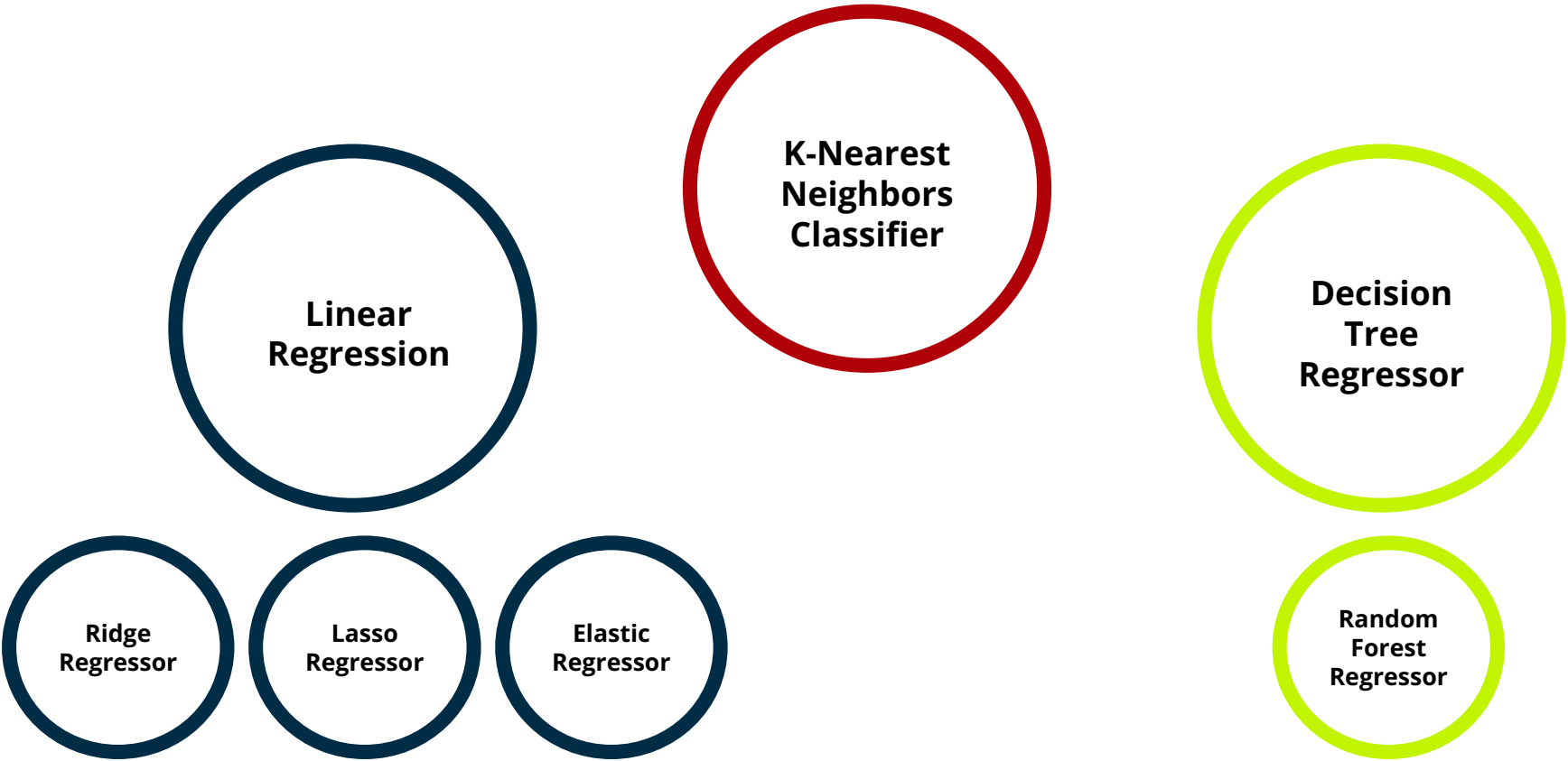
05 Challenge Advisors And TAs

06 Google

Our Approach



Our Approach



Data Understanding & Data Preparation

Understanding the Data

01 Term

String

the human readable identifier for a term, i.e. “Acme Inc”

02 Week

Date

first day of the week for the current row’s position in the time series for combination of term and score

03 Score

Int

index from 0-100 that denotes how popular this term was for a DMA during the current date, relative to the other dates in the same time series for this term (260 weeks = 52 weeks * 5 years)

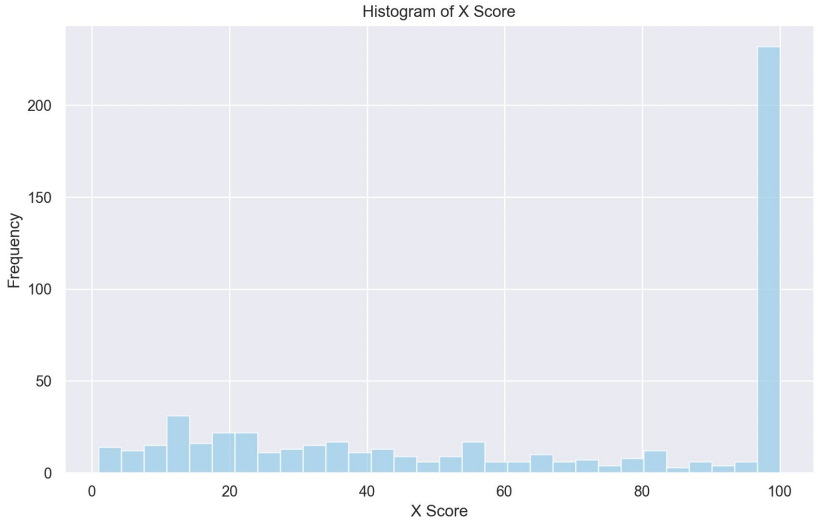
04 Rank

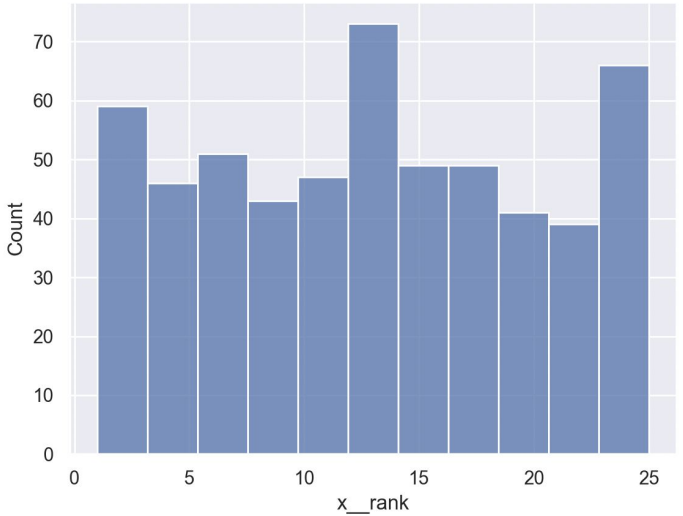
Int

numeric representation of where the term falls in comparison to the other top terms for the day across the US market (e.g. rank of 1-25)

Data Visualization

- Previous dataset:
  - x_score represents the number of times a query was searched
  - x_rank represents the ranking of each query
- Issues with the dataset: an aggregate of the search queries, so we are unable to predict accurately





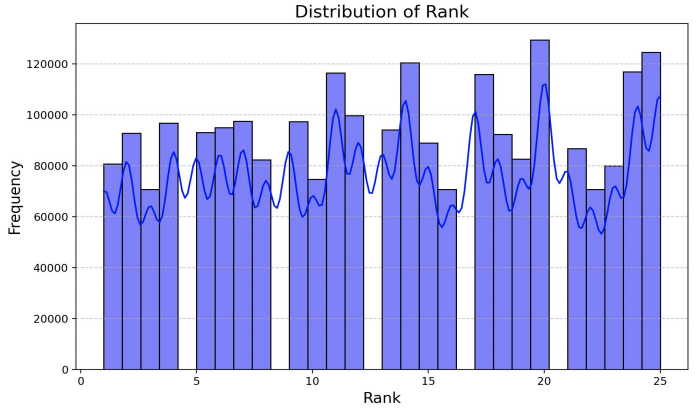
Data Visualization Pt. II - New Data

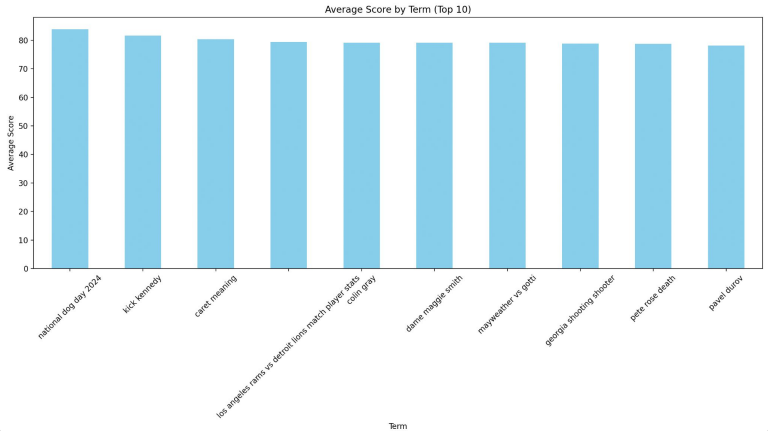
Data is too big resulting in running out of memory

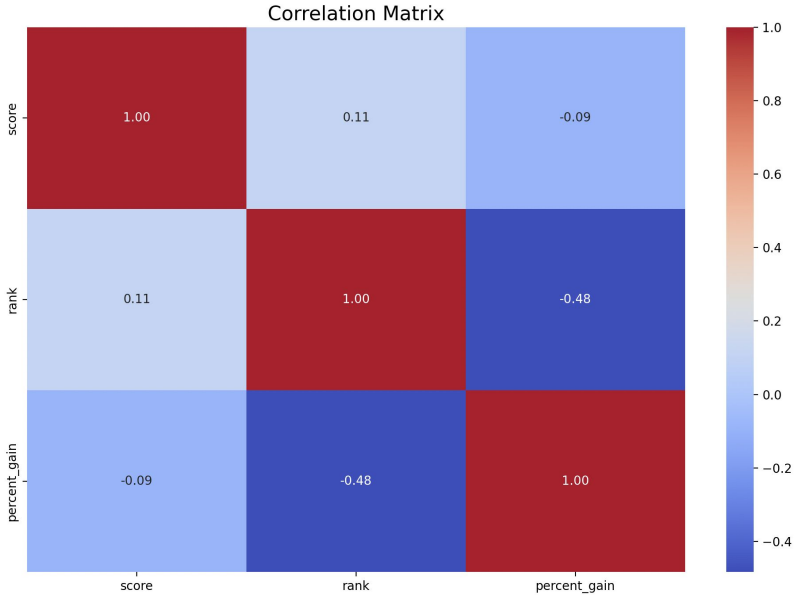
Smaller chunks: Changing the chunks into months

Use Dask: Works similarly to pandas but handles larger-than-memory datasets by parallelizing operations

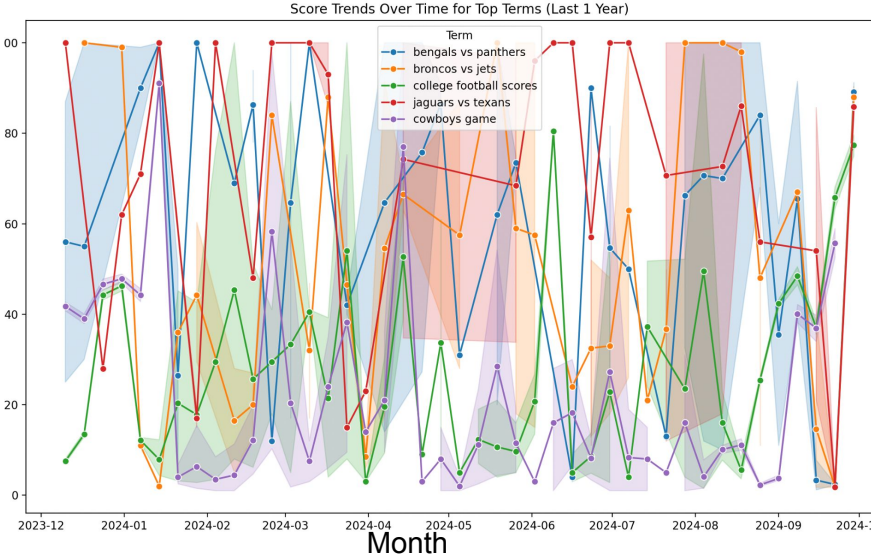
Data Visualization Pt. II - New Data

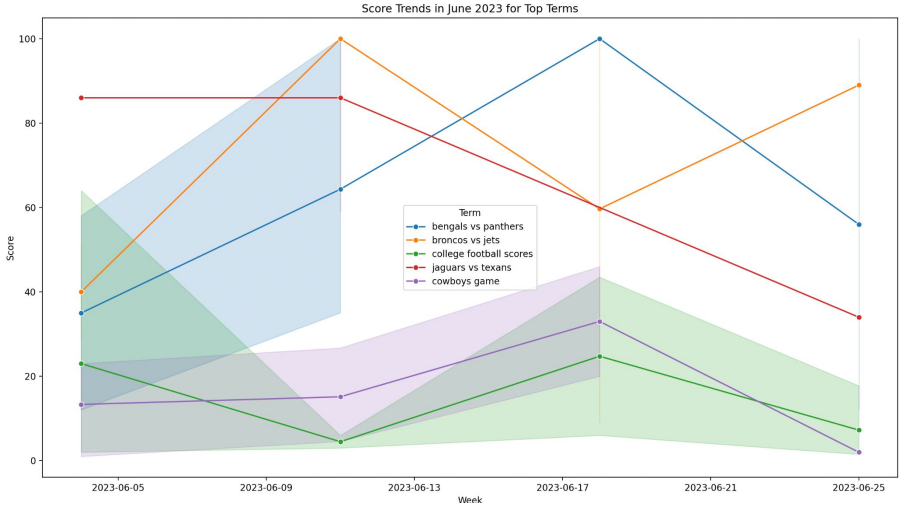






Data Visualization Pt. II - New Data





Data Cleaning - Part 1

01
Fill in missing data values
Fill in missing data values with -1

02
Transform
Transform date values to a numerical scale usable by a model.

03
Visual correlation plot and score
Use a visual correlation plot and score to perform feature selection (which features correlate with the label column).

04
Result
Little to no correlation was found (data set was far too small)...

Data Cleaning - Part 2

Steps Added...

01 Make data usable by our machines

02 Use a visual correlation plot and score to perform feature selection

03 Drop unnecessary columns (do not contribute valuable insights)

04 Fill in missing data values

05 Transform date values

06 One-hot encode categorical “search term” feature

07 Convert boolean-type

08 Standardize feature columns

Problem - Data

Building Models

Comparison of Models

- Metric for success: Accuracy
  - Lower cost for company
  - Efficiency in predictions
    - Reliability
  - Opportunity cost is less

Linear Regression - Data Leaks

Initial R-squared accuracy value of **>0.90.**

To test our data, we initially trained a basic linear regression model using Scikit-Learn's basic LinearRegression function, with ***no*** **tuned hyperparameters**.

However, this high accuracy was likely due to data leakage…

Data contains information about the same search terms, over several weeks

Predicted values

Because we split the training and testing data randomly, the model was likely using data from future dates to predict values for dates that occurred before them

.

Linear Regression - Data Leaks

- **Initial: Linear Regression model**
  - *No* hypertuned parameters
  - R-squared accuracy value: > 0.90
  - *Data Leakage*
- **Data**
  - Search terms over several weeks
- **Predicted Values**
  - Model was using data from future data for rank prediction

Data Leaks Solution

Identified Data Leaks

- **Train Test Split**
  - Model exposed to prediction week's data
- **Solution**:
  - Lower data volume
    - Training data set: 70%
    - Testing data set: 30%

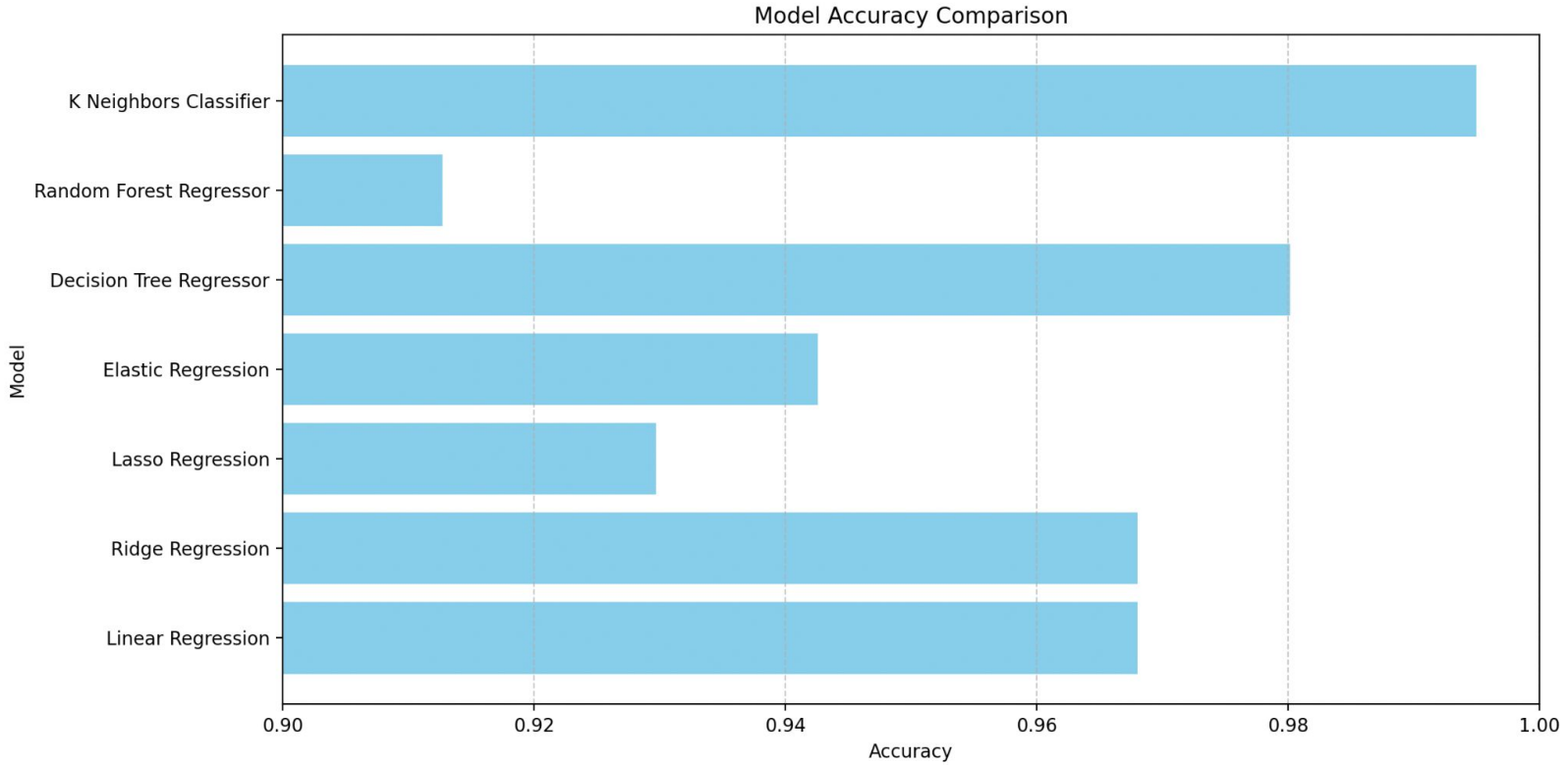
Linear Regression

- Base model for comparison
- **High accuracy**: 0.96

Model Comparison

Model Name	Results
Linear Regression	0.9680364947609563
Ridge Regression	0.9680364950517649
Lasso Regression	0.9297249063596292
Elastic Regression	0.9425945464028966
Decision Tree Regressor	0.9801675232906605
Random Forest Regressor	0.9126839606623705
K Neighbors Classifier	0.9949837258280682

Model Visualization Pt. III



Decision Tree Regressor

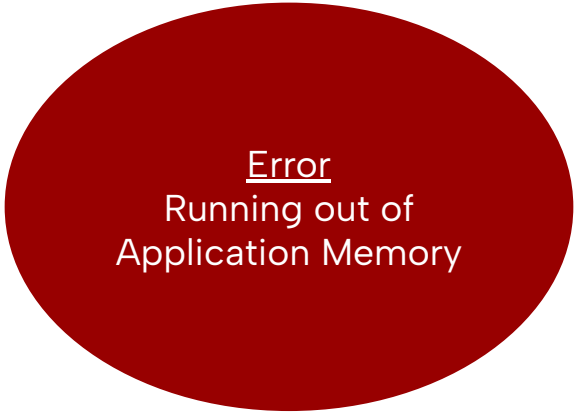
Parameters...

- **Criterion**
  - ‘Friedman mse’
    - Tests quality of data splitting
    - Accurate mean squared error testing
- **Maximum depth of the tree**
  - 5, 10, 15, 20
    - Values limit tree depth
    - Prevent overfitting
    - Less model complexity
- **Minimum number of samples at leaf node**
  - Tested: 1, 2, 5, 10
    - Higher values help figure out overfitting

Random Forest Regressor

Parameters...

- **Max_depth**
  - 3
- **N_Estimators**
  - 10
- Highest Complexity



A new thought...

K Neighbors Classifier

Parameters...

- **N_Neighbors**
  - 1, 2, 3, 4, 5
    - Best results: 3
    - Limitations: couldn't test beyond this complexity
- **Cross Validation Score**
  - Considered
    - Computationally expensive
    - Result data leakage

Next Steps (K Neighbors Classifier)

Hyperparameter tuning for optimization → Memory Space

More resources → Higher quality model, higher accuracy rates

Next Steps

Data preparation for new weeks

Automated data entry → Model predicts rank of new data

Hyperparameter tuning for optimization → Memory Space

Maintenance of models → Ensure quality of model as data progresses


Questions?