Efficient Visualization Recommendation under Updates

Todd Yu, Dixin Tang, advised by Prof. Aditya Parameswaran

Berkeley UNIVERSITY OF CALIFORNIA

EPIC DATA lab

Goal: Reduce latency when recomputing Visualization Recommendations during Data Analysis

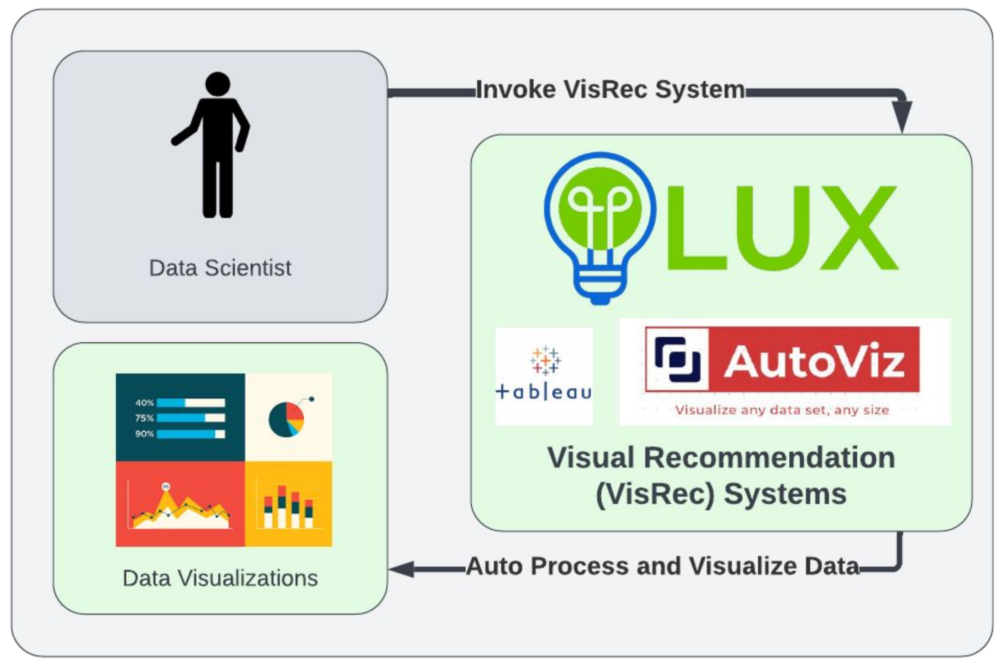
Background

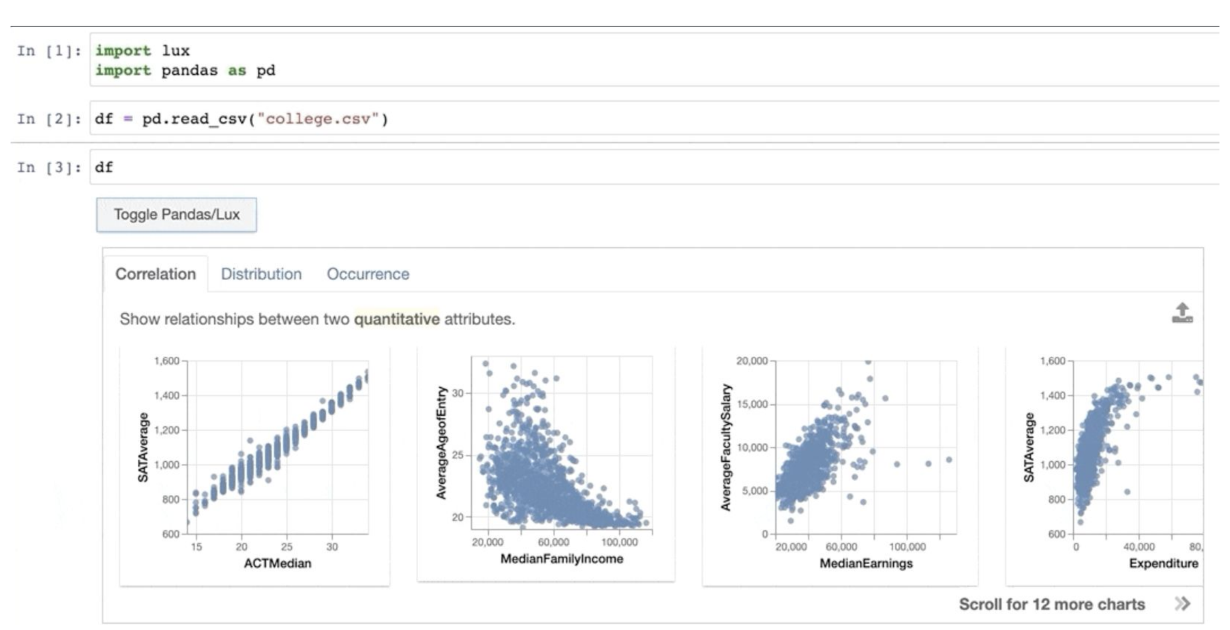
VisRec System – emerging system that automatically profiles data and recommends/generates visualizations

Ranking Scores – statistical, data-dependent scores for ranking possible visualizations

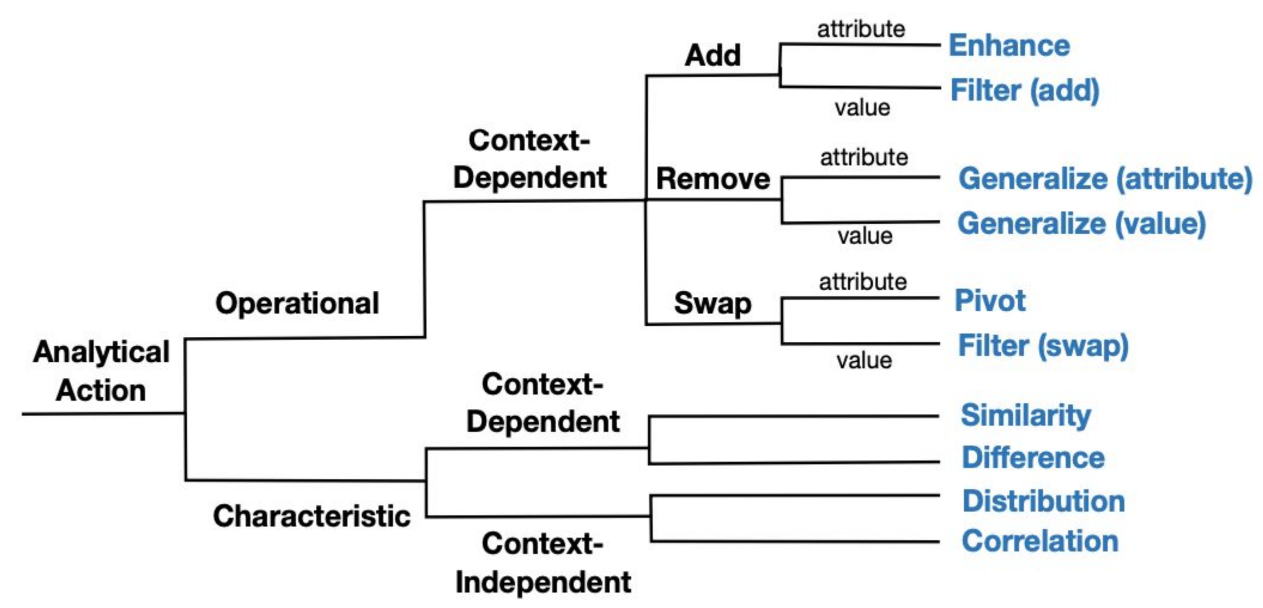
Problem – Computing ranking score statistics is **expensive**, creating high user latency

Our Solution – **Compute and Maintain** common VisRec system ranking scores with respect to common data-based Dataframe updates that map to real-world workflows





Decomposing VisRec System Ranking Scores



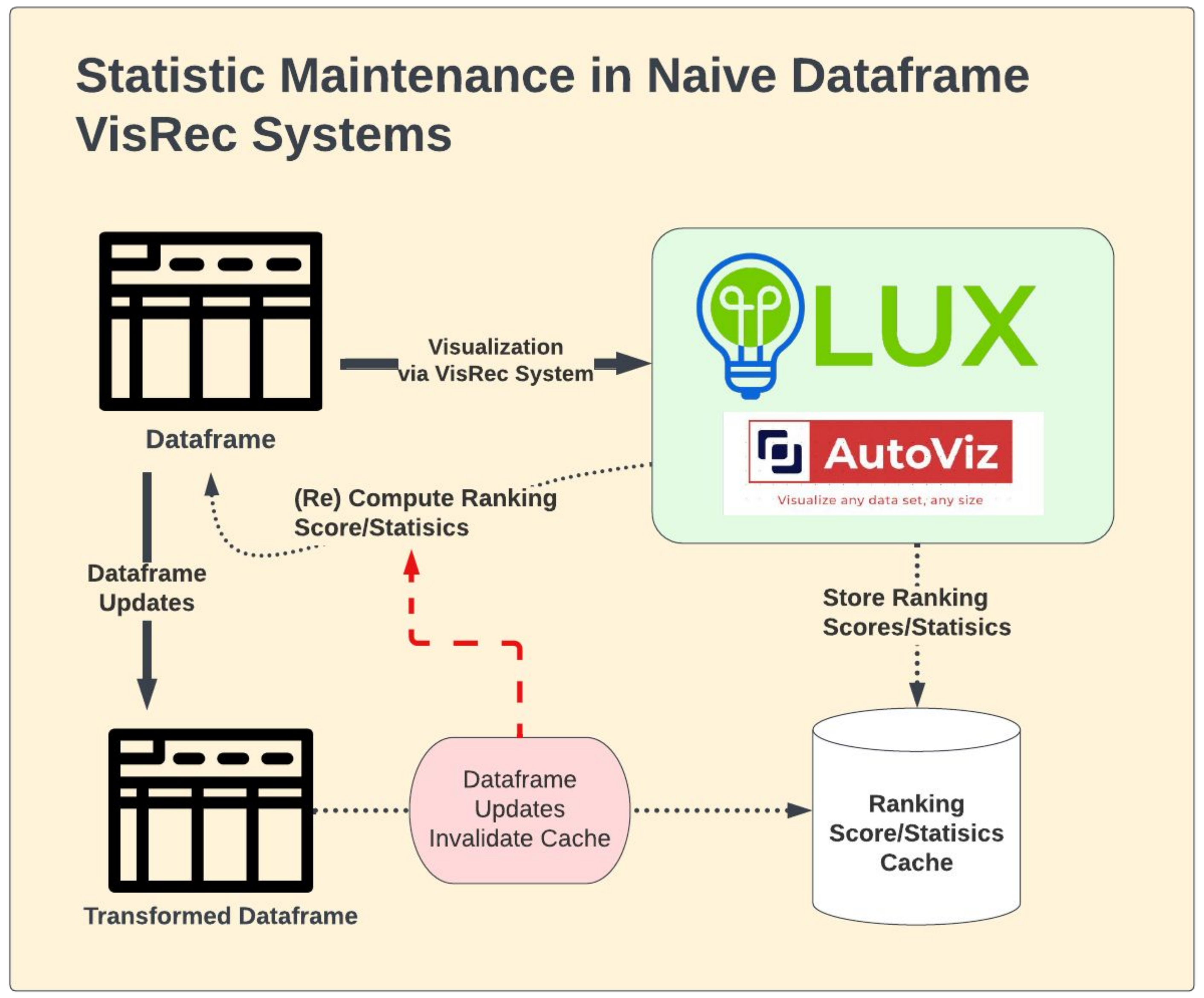
VisRec System Taxonomy – Taxonomy of common analytical actions for VisRec system (left) created by Lee et al. It presents VisRec system visualization categories (analytical actions) and associated ranking scores, which we **decompose into the table below**

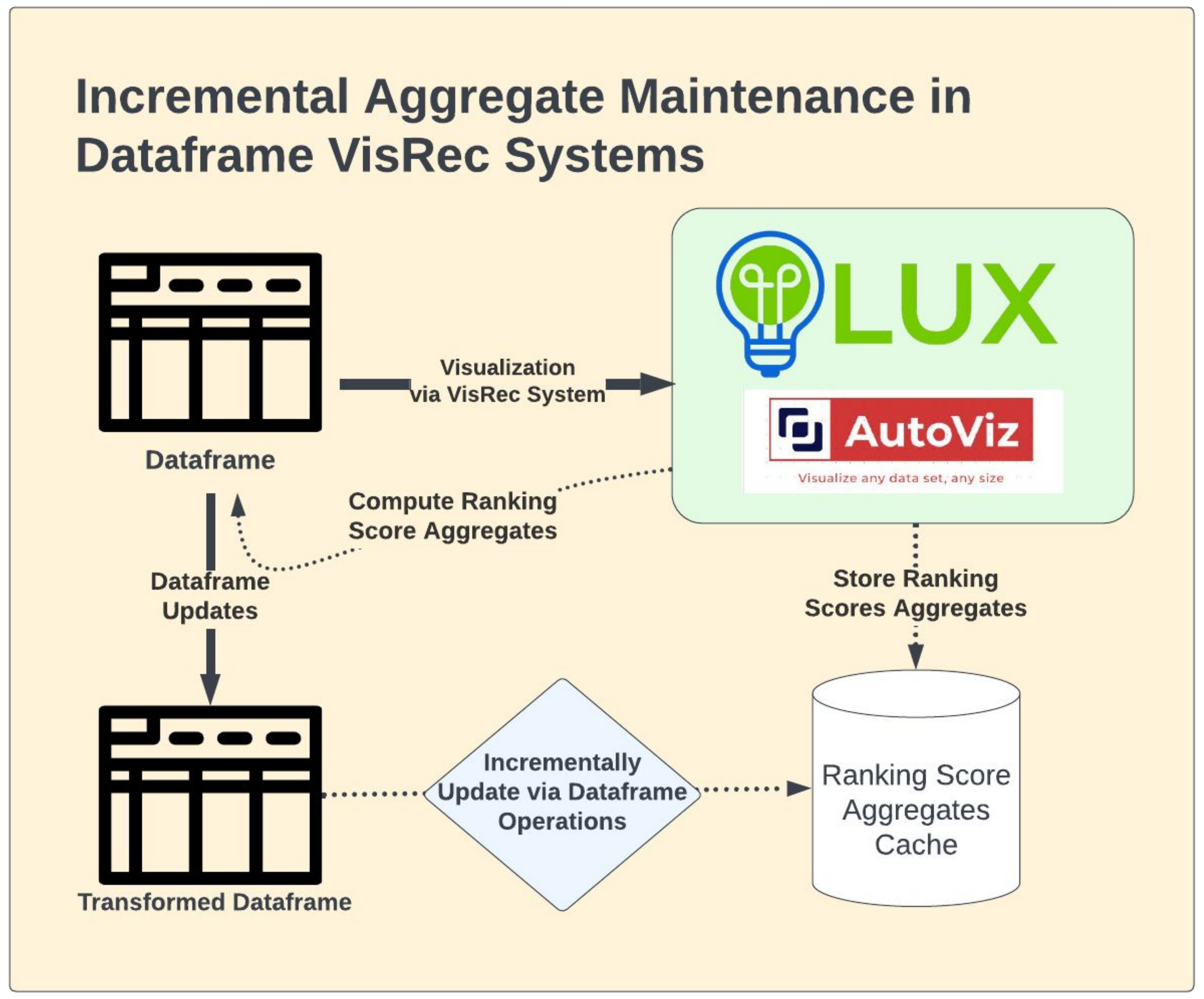
Ranking Score Decomposition – We decompose ranking scores into aggregates (right). We see that we have **five core aggregates** to maintain per column: sum of elements/elements^2, cardinality, pairwise sum, and downstream columns (e.g. filtered cols)

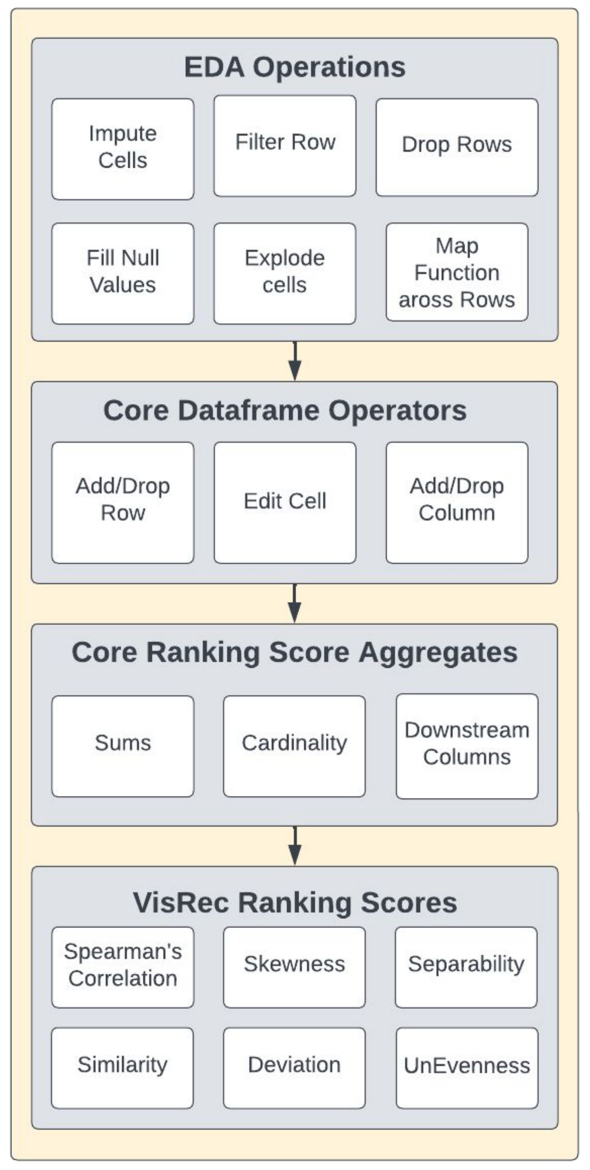
Ranking Score for Column X	$\sum_i X_i$	$\sum_i Y_i$	$\sum_i X_i^2$	$\sum_i Y_i^2$	$\sum_i X_i \cdot Y_i$	$\lvert X\rvert$
Correlation: Spearman$(X,Y)^2$	✓	✓[1]	✓	✓	✓	
Skewness: μ_X^3/σ_X^3	✓		✓			
Monotonicity: Spearman(X,Y)	✓	✓	✓	✓	✓	
Separability: Class Mean/Variance	✓		✓			
Similarity: $L_2(X, C_v)$			✓	✓	✓	
Deviation: $L_2(X, X_F)$			✓	✓[2]	✓	
Unevenness: $L_2(V_X, V_{flat})$, $V_X = \gamma_X$	✓[3]		✓[4]			✓

Table 2.1: Decomposing ranking scores for Data Variable (Column) X and (optional) external column Y into their respective aggregates. The columns of the table are listed as follows: Sum of X, Sum of Y, Sum of X's squared elements, Sum of Y's squared elements, Inner product of X and Y, Cardinality of X. V_X is an aggregate over X, X_F represents filtered values, and C_v represents current view.

Maintaining Ranking Scores

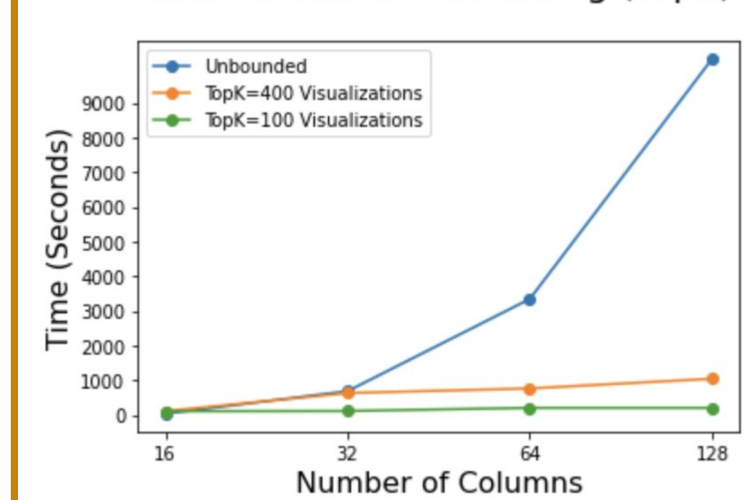




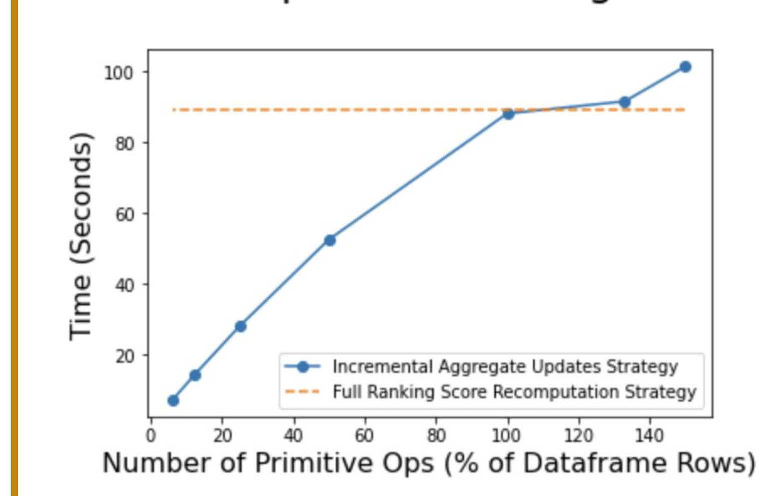


Evaluation

Latency of Computing Ranking Scores for 1000 Operations in Crime Data with Visualization Pruning (TopK)



Comparing Latency of Ranking Score Computation Strategies



Evaluation – We implement/evaluate our system in Lux, a popular Dataframe VisRec system that covers all analytical actions listed

Findings – **Maintaining aggregates is always faster** when number of Dataframe row updates cost is less than cost for computing equivalent updates based on number of existing rows, based on our cost model