ExplainED: Explanations for EDA Notebooks

Daniel Deutch, Amir Gilad[†], Tova Milo, and Amit Somech[‡]
Tel Aviv University
danielde@post.tau.ac.il, {amirgilad[†],amitsome[‡]}@mail.tau.ac.il, milo@cs.tau.ac.il

ABSTRACT

Exploratory Data Analysis (EDA) is an essential yet highly demanding task. To get a head start before exploring a new dataset, data scientists often prefer to view existing *EDA notebooks* – illustrative exploratory sessions that were created by fellow data scientists who examined the same dataset and shared their notebooks via online platforms. Unfortunately, creating an illustrative, well-documented notebook is cumbersome and time-consuming, therefore users sometimes share their notebook without explaining their exploratory steps and their results. Such notebooks are difficult to follow and to understand.

To address this, we present `ExplainED`, a system that automatically attaches explanations to views in EDA notebooks. `ExplainED` analyzes each view in order to detect what elements thereof are particularly interesting, and produces a corresponding textual explanation. The explanations are generated by first evaluating the interestingness of the given view using several measures capturing different interestingness facets, then computing the Shapely values of the elements in the view, w.r.t. the interestingness measure yielding the highest score. These Shapely values are then used to guide the generation of the textual explanation.

We demonstrate the usefulness of the explanations generated by `ExplainED` on real-life, undocumented EDA notebooks.

PVLDB Reference Format:
Daniel Deutch, Amir Gilad, Tova Milo, and Amit Somech. `ExplainED`: Explanations for EDA Notebooks. *PVLDB*, 13(12): 2917-2920, 2020.
DOI: https://doi.org/10.14778/3415478.3415508

1. INTRODUCTION

Exploratory Data Analysis (EDA) is an important step in any data scientific (DS) pipeline. Typically, EDA is done by applying a series of data-analysis operations (such as filtering, aggregation, and visualization) on an input dataset, with the goal of better understanding its nature and characteristics. The information accumulated thereby is often useful for subsequent parts of the pipeline such as feature engineering, model selection, etc.

Since EDA is known to be a difficult process, data scientists often examine *EDA notebooks* prepared and shared by others [8]. An EDA notebook contain a curated summary of an EDA process, presented through a *notebook interface* – a literate programming environment that allows users to easily document a sequence of programmatic operations, their results, as well as to add free-text explanations (see Figure 1 for an illustration).

However, creating illustrative, well-documented notebooks requires time and effort, and thus users sometimes refrain from adding explanations to their exploratory steps [8]. Notebooks without explanations are much harder to follow and to understand what exactly is interesting and important in each exploratory step.

To that end, we showcase `ExplainED`, a system for automatically explaining views in EDA notebooks. The explanations are presented in Natural Language (NL) and describe the particular elements of the view that are the most interesting. `ExplainED` analyzes the interestingness of each view, and computes the Shapley values [14] of its elements w.r.t. the interestingness score of the entire view. Shapley values were originally proposed in the context of Game Theory as a means of quantifying the contribution of each player to the achieved game value. Here, `ExplainED` uses Shapely to measure the contribution of each tuple to the interestingness score of the view. `ExplainED` then presents the view elements with the highest Shapley values; for clarity, these elements are presented in NL, using pre-defined textual templates. To illustrate, consider the following example:

EXAMPLE 1.1. *Consider the EDA notebook illustrated in Figure 1 exploring the 'flights.csv' dataset (publicly available on Kaggle[1]), containing details on delays and cancellations of domestic US flights. To illustrate the benefit of* ***`ExplainED`****, we demonstrate the explanations that were automatically generated for two views out of the four depicted in Figure 1 (the explanation for the third view is omitted for brevity). The first operation in the notebook is a group-by (denoted by the number 1), shows the average flight delay per month of the year. To highlight the interesting parts in the view it generates,* ***`ExplainED`*** *first analyzes the interstingness of the view, and determines that it is interesting since it contains diverse values (we explain how this is done in Section 2). It then derives the tuples most influencing the diversity of the view, and outputs a corresponding textual explanation, as appears in the red frame. This explanation allows*

[1]See `https://www.kaggle.com/usdot/flight-delays`.

Proceedings of the VLDB Endowment, Vol. 13, No. 12
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3415478.3415508

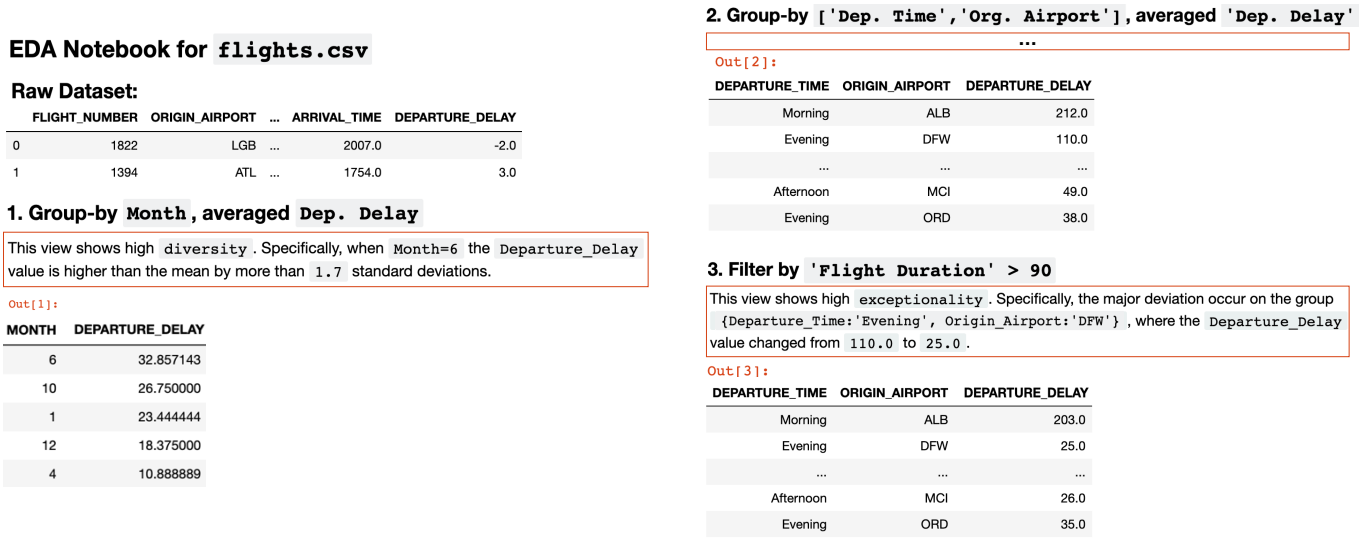

Figure 1: EDA Notebook with Explanations Produced by ExplainED (in the red frames)

users to focus on the fact that delays are considerably longer in June, compared to the other months. Then, for the view created by operation 3, ExplainED has determined that there is a group whose aggregated value significantly changed due to the filter operation, and therefore highlighted this group in the explanation. In both views, ExplainED assists users in finding the interesting parts of the view. This feature may be useful for users who analyze undocumented notebooks, as well as for users who create notebooks themselves and wish to quickly understand the gist of the view and document their EDA process.

In a nutshell, ExplainED takes as input a view from a given EDA notebook, and generates a textual explanation as follows: First, the *interestingness* of the view is evaluated using several measures, each corresponding to a different interestingness facet (e.g., diversity, conciseness, exceptionality, etc.). Next, focusing on the measure that yielded the highest score, ExplainED computes the Shapley values of the top-k elements in the view (i.e., groups/tuples) w.r.t. the interestingness measure. Last, the top-k elements with the highest Shapely values are used to compose an explanation sentence based on a prefabricated textual pattern, chosen according to the interestingness measure we focus on.

ExplainED can assist users in two ways: (1) explaining undocumented notebooks created by other users (in particular, *automatically-generated* EDA notebooks [1]), and (2) documenting their own EDA notebooks.

In our demonstration, We will first present the audience with an undocumented EDA notebook, then reveal the explanations generated by ExplainED for each exploratory step. Finally, we will present the manner in which ExplainED chooses the most interesting tuples and generates explanations that incorporate them.

Related Work. Various methods of explaining query results have been proposed in the literature. Prominently, explanations using provenance [7, 2], interventions [13], influence [17], Shapley values [9], or using NL [4] among others. The main difference between these works and ours is that these works explain which input tuples affected the output of a query, while we try to find the input tuples that make the view *interesting* (i.e., that most affect the view's interestingness score). There are also other tools for assisting users in composing EDA steps. For example, recommendations of EDA next-steps (e.g., [12]), and highlighting promising features to explore (e.g., [6]). However, such tools do not explain why are the generated views considered interesting.

2. TECHNICAL OVERVIEW

We next define our data model for EDA notebooks and the considered interestingness measures. In Section 2.2, we describe how ExplainED generates an explanation for a view.

2.1 Model & Background

We first describe our data model for EDA notebooks.

Data Model for EDA Notebooks. An EDA notebook created w.r.t. a given input dataset is defined as a sequence of views $V_0, \ldots, V_n$, where V_0 is the initial dataset. Each view is modeled as a set of elements $V_i = \{e_1, \ldots, e_k\}$, where an element e_i is either a tuple or a group (in case v_i is grouped-by). Each element e_i is a vector of values $(v_1, \ldots, v_t)$ over an attribute vector $Attr$. To get V_i from V_{i-1} we use an EDA operations q_i.

Following common data manipulation routines in notebooks (such as Python and R) the sequence is constructed in a parametric, incremental manner: each operation q_i is applied on one of the previous views and takes additional parameters. For example, a filter operation is defined by $\texttt{FILTER}(V, attr, op, term)$, and is used to select data tuples from view V that match a criteria. It takes an attribute, a comparison operator (e.g. $=, \geq, contains$) and a numeric/textual term, and results in a new view representing the corresponding data subset. Another operator is $\texttt{GROUP}(g\_attr, agg\_func, agg\_attr)$ which groups and aggregates the data. It takes a single[2] attribute to be grouped by, an aggregation function (e.g. SUM, MAX) and another attribute to employ the aggregation function on. Additional

[2]Complex group-by operations, are possible by employing consecutive group-by operations.

data manipulation such as aggregate, join and project are similarly represented.

EXAMPLE 2.1. *Reconsider the EDA notebook in Figure 1. The first EDA operation is* `GROUP`*(MONTH, AVG, DEP_DELAY), performed on the raw dataset (i.e., V_0). The resulted view is denoted by V_1. The second view, V_2, was generated from V_0 using* `GROUP`*(DEP_TIME, AVG, DEP_DELAY) and then using* `GROUP`*(ORIGIN_AIRPORT, AVG, DEP_DELAY). Last, View V_3 was generated by employing* `FILTER`*(FLIGHT_ DURATION,>,90) on V_2.*

Given a view V_i, `ExplainED` generates an explanatory text E_i, which highlights the elements that are particularly interesting in V_i. For example, see the generated explanations in the red frames in Figure 1.

Interestingness Measure for EDA Views. As explained in the sequel, `ExplainED` analyzes the interestingness of each EDA operation q_i before producing an explanation that describes what exactly is interesting in the resulting view V_i. An *interestingness measure* I is a function mapping each view to a real number (higher score indicates a more interesting view).

There are multiple different interestingness measures defined in the literature [5]. In our implementation, we focus on three common measures from the literature that we adapted to the EDA use case, as follows:

Conciseness Evaluation. Intuitively, a "concise" group-by view, that covers many tuples using a small number of groups, is both informative and easy to understand [5]. Correspondingly, the conciseness measure that we use evaluates group-by operations, following similar lines as existing measures [5, 3]. In case the view is grouped, the measure takes into account the number of groups, the number of attributes that are currently grouped-by, and the number of the underlying tuples, denoted g,a, and r (respectively). The calculation is given by:

$$\frac{h(g \cdot a)}{h(r)} \tag{1}$$

where $h(\cdot)$ is a normalized sigmoid function with a predefined width and center. In case V_i is not grouped, the measure is defined by $h(\frac{1}{r})$, giving a higher score to views with a small number of tuples.

Exceptionality Evaluation. We follow common measures in the literature that assess the *exceptionality* of an EDA operation. Following [16], our measure favors operations whose result view V_i deviates significantly from the previous one V_{i-1}. To quantify such deviation, we use the KL divergence measure, which determining the difference between probability distributions (Other measure, e.g., Earth Mover's Distance, are also viable.) The calculation depends on whether the view is grouped-by or not: In case V_i is not grouped, we define the value probability distribution P_A^i of an attribute $A \in Attr$ to be the relative frequency of its values (i.e., for each value v of attribute A in V_i, $p(v)$ is the probability to randomly choose v). The exceptionality score is defined by:

$$\max_{A \in Attr} KL(P_A^{i-1}, P_A^i) \tag{2}$$

Namely, the maximal KL divergence score obtained by the attribute that demonstrates the most significant "change" in its distribution due to the last filter operation. In case V_i is grouped, the KL divergence is compared only w.r.t. currently aggregated attributes (rather than on all attributes).

Diversity Evaluation. We use the *Variance* [5] diversity measure that rank higher views whose elements demonstrate notable differences in values. The measure is defined by:

$$h(\max_{A \in Attr} \mathrm{Var}(P_A)) \tag{3}$$

where P_A is the value distribution of attribute A as above.

EXAMPLE 2.2. *Consider the views in Figure 1. V_3 is generated from Operation 3 (filter). Intuitively (full calculation omitted for brevity), V_3 has high exceptionality score, as the aggregated value changes significantly from the one in V_2 due to the filter operation. Conversely, the number of groups is quite large, therefore the conciseness score of V_2 and V_3 is low. Also, notice that V_1 in Figure 1, generated by Operation 1, has high diversity score, as the aggregated values demonstrate high variance. Its conciseness score is also rather high, as its total number of groups is only* 12.

2.2 Explaining EDA Operations

Given a view V_i in an EDA notebook, we first assess its interstingness w.r.t. the measures defined above, then derive which specific elements in the view have the highest impact on the interestingness score of the view, and present them in an illustrative, NL template.

Choosing the Relevant Interestingness Measure. Each of the measures presented above captures a different aspect of interestingness. Therefore, views showing high score w.r.t one measure may show a lower score w.r.t others. We therefore wish to find the measure most relevant to V_i.

Since the measures have different value ranges and distributions, we first employ the interestingness comparison technique in [11], which derives an unbiased, *relative* interestingness score, for each measure, w.r.t. the scores of alternative views (that were generated by EDA operations of the same type). We then focus our explanations, as described next, on the measure producing the highest relative interestingness for V_i

Finding Tuples to Include in the Explanation. Next, we describe how to locate the elements in the view V_i that we wish to showcase in the explanation, i.e., the elements that make V_i interesting w.r.t. the chosen measure. This problem can be considered as a game theoretic problem, where tuples are 'players' and each tuples has a certain value to the game (the interestingness score, in our case). We are interested in those tuples that provide the maximum value to the interestingness score of V_i. Thus, a natural approach is to look at the Shapley value of each element in the view w.r.t. the interestingness score. This approach is common practice for ML models [15, 10] to evaluate the contribution of different features. We formalize the definition of a Shapley value of an element in a view w.r.t. an interestingness measure as follows.

DEFINITION 2.3. *Given an EDA view V_i generated from V_{i-1} and an interestingness measure I, the Shapley value of the element $e \in V_i$ is defined as:*

$$Shap(V_i, I, e) = \sum_{S \subseteq V_i \setminus \{e\}} \frac{|S|!(|V_i| - |S| - 1)!}{|V_i|!} \cdot (I(V_i|_{S \cup \{e\}}) - I(V_i|_S))$$

Note that V_i is represented as a set of elements.

The above definition intuitively says that the contribution of each tuple to the interestingness of the view is given by its Shapley value. We compute the Shapley value for every element by flattening the table into a vector (s.t. each eleement is associated with a coordinate) and then use SHAP [10] w.r.t. the chosen interestingness measure. Finally, the elements associated with the top-k Shapley values are used for generating the explanation.

EXAMPLE 2.4. *Reconsider view V_3 in Figure 1 (created by Operation 3), where the relevant interestingness measure is Exceptionality. The group we present in the explanation is composed of the tuples associated with evening flights that had a delay and left from the DFW airport, as the Shapley value of this group is the highest among the groups in V_3. We show the computation of the Shapley value for the group ('Evening', 'DFW', 25) (denoted D for brevity), considering only the four groups shown in V_3. Based on Definition 2.3, we consider the 8 subsets: $\emptyset$, $\{A\}$, $\{M\}$, $\{O\}$, $\{A, M\}$, $\{A, O\}$, $\{M, O\}$, $\{A, M, O\}$, and for each one, we compute the coefficient (e.g., for $\{A\}$ and for $\{A, M\}$ it is $\frac{1}{12}$) and the difference between the KL-divergence function of the subset with D and without it (e.g., the difference for KL-divergence of both $\{A\}$ and $\{A, M\}$ is 162.9). Overall, the values we get are: $Shap(A) = 9.2$, $Shap(D) = 162.9$, $Shap(M) = 31$, $Shap(O) = 3.1$.*

Generating NL Explanations. We employ NL patterns, one per supported interestingness measure, that are aimed to accommodate a textual description w.r.t. chosen measure. As depicted in Figure 1, each explanation begins with stating the chosen measure: "This view shows high [`chosen_measure`]". Then, the description of the top-k elements is generated via the template patterns shown in Figure 2 (for $k = 1$, group-by views). Every expression in square brackets is filled-in with the relevant attribute or value according to the top-k tuples. In general, more templates can be devised to provide tailored explanations.

```
Specifically,when [&attribute=
value] the [&attribute] value is
[&higher/lower] than the mean in [&x]
standard deviations
```

(a) Diversity Pattern

```
Specifically, the major deviation occurs
on the group [top-1 group_name] where the
[attribute] value changed from [value in V_{i-1}]
to [value in V_i]
```

(b) Exceptionality Pattern

```
Specifically, the Group
[&top-1 groups in V_i] covers more than
[&coverage] of the underlying tuples
```

(c) Conciseness Pattern

Figure 2: Textual Patterns for View Explanations

3. DEMONSTRATION

We will demonstrate the explanations that `ExplainED` generates for EDA notebooks and their usefulness over the Kaggle Flights dataset (as described in Example 1.1).

Data Exploration with `ExplainED`: Using the flight dataset, we will perform a live, interactive EDA process, using a notebook interface, in pursuit of an exploration goal: characterize delays in the datset. We will employ `ExplainED` to dynamically generate an explanation for each view in the notebook, demonstrating the value of the explanations to the data analysis process.

Technical Details: We will let participants look under the hood of `ExplainED` by showing the manner in which it selects the most relevant interestingness measures and finds the interesting tuples or groups in views based on their Shapley values. Finally, we will explain how the system composes the textual explanation.

Acknowledgements. This research has been funded by the Israeli Science Foundation (ISF), the Binational US-Israel Science Foundation, the Tel Aviv University Data Science center, the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 804302), and the Google Ph.D. Fellowship.

4. REFERENCES

[1] O. Bar El, T. Milo, and A. Somech. Automatically generating data exploration sessions using deep reinforcement learning. In *SIGMOD*, 2020.
[2] P. Buneman, S. Khanna, and W. Tan. Why and where: A characterization of data provenance. In *ICDT*, pages 316–330, 2001.
[3] V. Chandola and V. Kumar. Summarization - compressing data into an informative representation. *KAIS*, 12(3), 2007.
[4] D. Deutch, N. Frost, and A. Gilad. Provenance for natural language queries. *PVLDB*, 10(5):577–588, 2017.
[5] L. Geng and H. J. Hamilton. Interestingness measures for data mining: A survey. *CSUR*, 2006.
[6] A. Giuzio, G. Mecca, E. Quintarelli, M. Roveri, D. Santoro, and L. Tanca. Indiana: An interactive system for assisting database exploration. *Information Systems*, 83:40–56, 2019.
[7] T. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *PODS*, pages 31–40, 2007.
[8] M. B. Kery, M. Radensky, M. Arya, B. E. John, and B. A. Myers. The story in the notebook: Exploratory data science using a literate programming tool. In *CHI*, 2018.
[9] E. Livshits, L. E. Bertossi, B. Kimelfeld, and M. Sebag. The shapley value of tuples in query answering. In *ICDT*, pages 20:1–20:19, 2020.
[10] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *NIPS*. 2017.
[11] T. Milo, C. Ozeri, and A. Somech. Predicting "what is interesting" by mining interactive-data-analysis session logs. In *EDBT*, 2019.
[12] T. Milo and A. Somech. Next-step suggestions for modern interactive data analysis platforms. In *KDD*, 2018.
[13] S. Roy and D. Suciu. A formal approach to finding explanations for database queries. In *SIGMOD*, pages 1579–1590, 2014.
[14] L. SHAPLEY. A value for n-person games. *Contributions to the Theory of Games*, (28):307–317, 1953.
[15] E. Strumbelj and I. Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.*, 41(3):647–665, 2014.
[16] M. van Leeuwen. Maximal exceptions with minimal descriptions. *DMKD*, 21(2):259–276, 2010.
[17] E. Wu and S. Madden. Scorpion: Explaining away outliers in aggregate queries. *PVLDB*, 6(8):553–564, 2013.