

Ava: From data to insights through conversations

A review by
Apaar Shanker

DATA ANALYTICS
USING DEEP LEARNING
GT CS 8803 // FALL 2018 //

Paper under review

Ava: From Data to Insights Through Conversation

Authors: Rogers Jeffrey Leo John¹, Navneet Potti¹, Jignesh M. Patel¹

Computer Sciences Department, ¹University of Wisconsin-Madison

Publication: CIDR '17

doi:<http://pages.cs.wisc.edu/~jignesh/publ/Ava.pdf>

Motivation

Why do my customers churn?



The current paradigm of data driven decision making



Data Scientist



A legal pyramid scheme (org chart)

Issues with the current model

1. Lost In translation
2. Long turnaround time
3. Correctness
4. Reproducibility
5. A cognitive overload due to surfeit of models and libraries

Proposed Solution

Key Observations:

- Controlled natural language methods are now practically implemented as interfaces to software toolboxes
- The data science workflow can be templated

We can use a chat-bot as a natural language UI to set up a data science pipeline by drawing on templates stored in a library.

Data Science Pipeline



Data Loading

- from a csv file



Data Cleaning

- fill missing values



Feature Engineering

- pick/create appropriate features



Model Selection and Training

- pick an ML model based on the input and task at hand



Parameter Tuning

- hyperparameter optimization



Save model for deployment

- As a UDF, PMML file, ...

Typical Data Science Workflow

The workflow is a (often cyclic) graph.

The actual pipeline is a subgraph of the workflow graph.

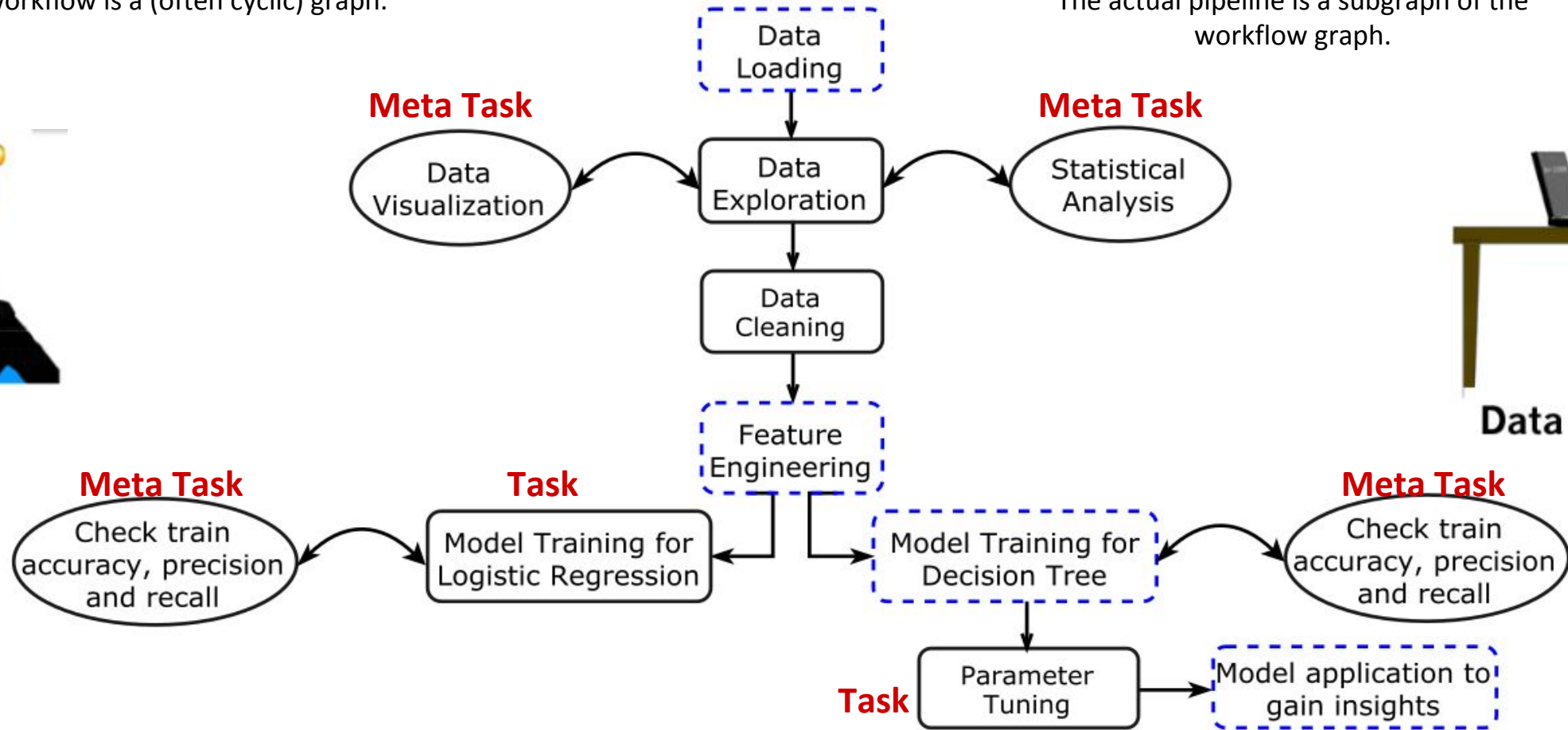


Figure 1: The Data Science workflow for Example 1. Tasks are shown in rectangles, and meta-tasks in ovals. The pipeline is highlighted in dotted blue boxes.

Once a workflow has been finalized - only the pipeline(constituted of dotted blue boxes) needs to be preserved.

Data Science Workflow can be Templated

```
from sklearn import tree
```

```
model = DecisionTreeRegressor(criterion= 'mse', splitter= 'best',  
max_depth=None)
```

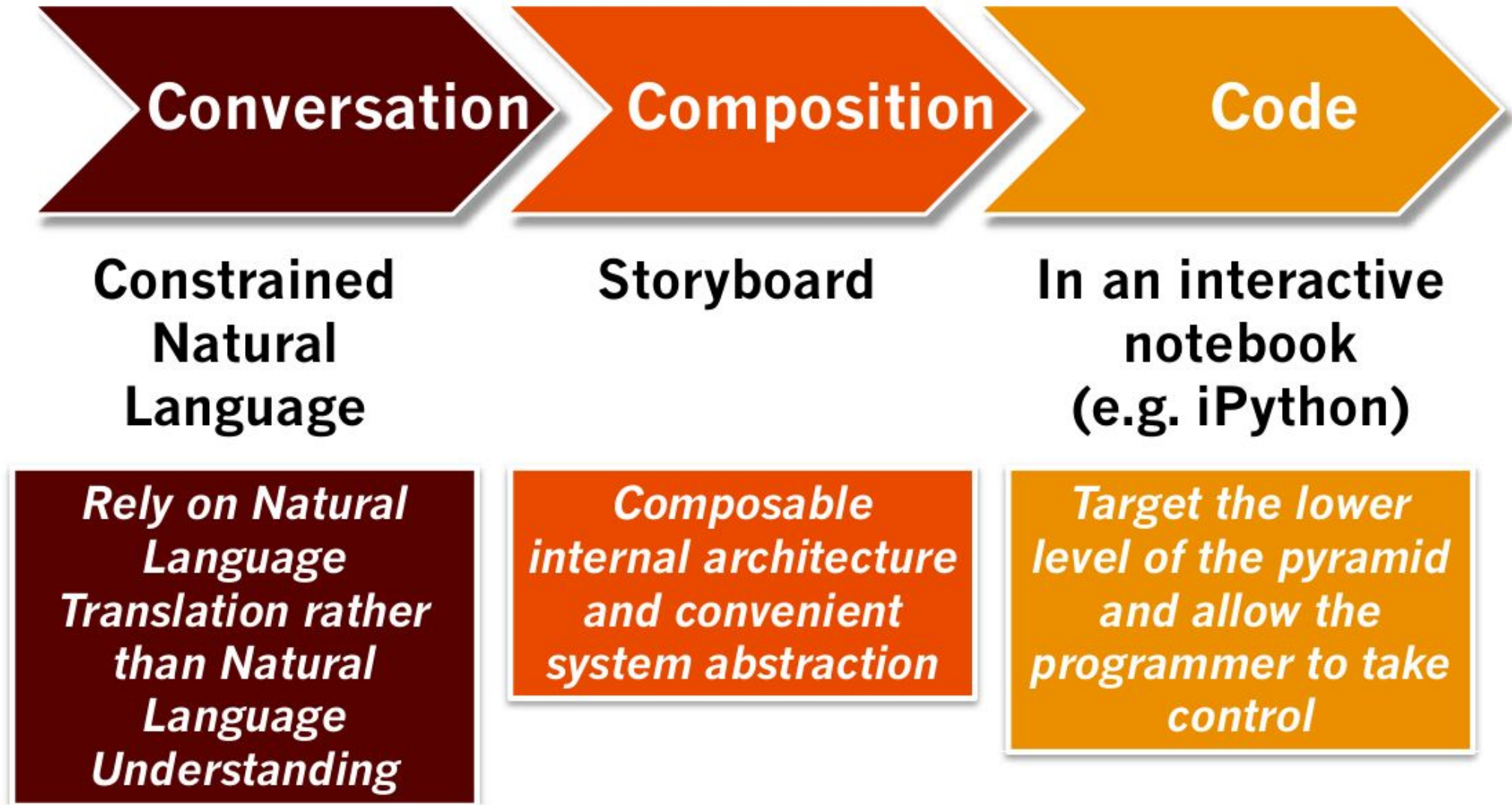
```
model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)
```

There is a clean separation of **specification** (parameter values) and **template**, such that task can be composed by simply substituting parameters into a pre-defined code template.

Insights

- Often the task is ...



Introducing AVA

The screenshot displays a Jupyter Notebook interface with a code cell and its output. The code cell contains the following Python code:

```
dot_data = StringIO()
tree.export_graphviz(dt_clf_gini, out_file=dot_data,
                    filled=True, rounded=True,
                    feature_names = [c.encode("utf-8") for c in column_names],
                    class_names = [str(v) for v in loaded_data["Survival"]],
                    special_characters=True)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

The output of the code cell is a decision tree visualization. The root node is an orange box with the following text: "Age ≤ 6.5", "gini = 0.4694", "samples = 534", "value = [333, 201]", "class = 0". The tree branches into two paths: "True" and "False".

- The "True" branch leads to a blue box: "Sex_male ≤ 0.5", "gini = 0.4297", "samples = 32", "value = [10, 22]", "class = 1". This node further branches into two blue boxes, both with "gini = 0.4297", "samples = 16", "value = [5, 11]", "class = 1".
- The "False" branch leads to an orange box: "Sex_female ≤ 0.5", "gini = 0.4589", "samples = 502", "value = [323, 179]", "class = 0". This node branches into two orange boxes: "gini = 0.2571", "samples = 330", "value = [280, 50]", "class = 0" and "gini = 0.375", "samples = 172", "value = [43, 129]", "class = 1".

On the right side of the notebook, there is a chat interface with AVA. The chat history includes:

- Ava:** Your Data Science assistant you have are:
 - An integer number, which will set the max_features features at each split
 - A decimal number, in this case the max_features is a percentage and int(max_features * n_features) features are considered at each split
 - If auto, then max_features=sqrt(n_features)
 - If None, then max_features=number of features
- You:** 1
- Ava:** The accuracy after cross validation is 0.805. Do you want to run your model on the test data?
- You:** yes
- Ava:** The testing accuracy is 0.779. What do you want to do next?
- Hint:** Type "show ROC plot" to see a ROC of the classifier you chose. You can also type "pick Column1, Column2, Column3 as features" to select a new set of features. If your current choice of classifier is decision trees, type "show the decision tree" to visualize the tree.
- You:** show decision tree
- Ava:** Decision Tree generated. Do you want to view any other plots?
- Hint:** Type "show ROC plot" to see a ROC of the classifier you chose. If your current choice of classifier is decision trees, type

AVA in action

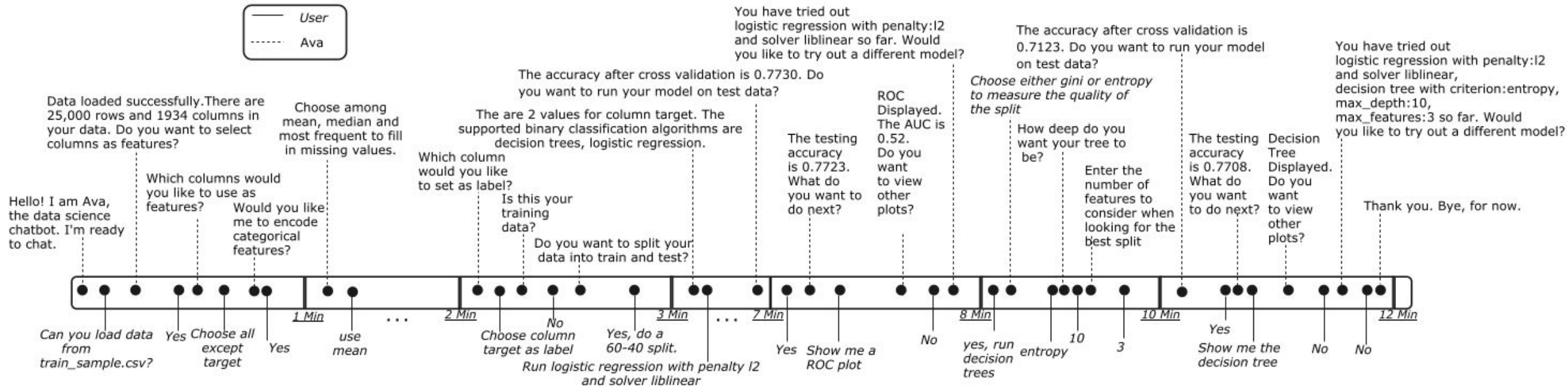
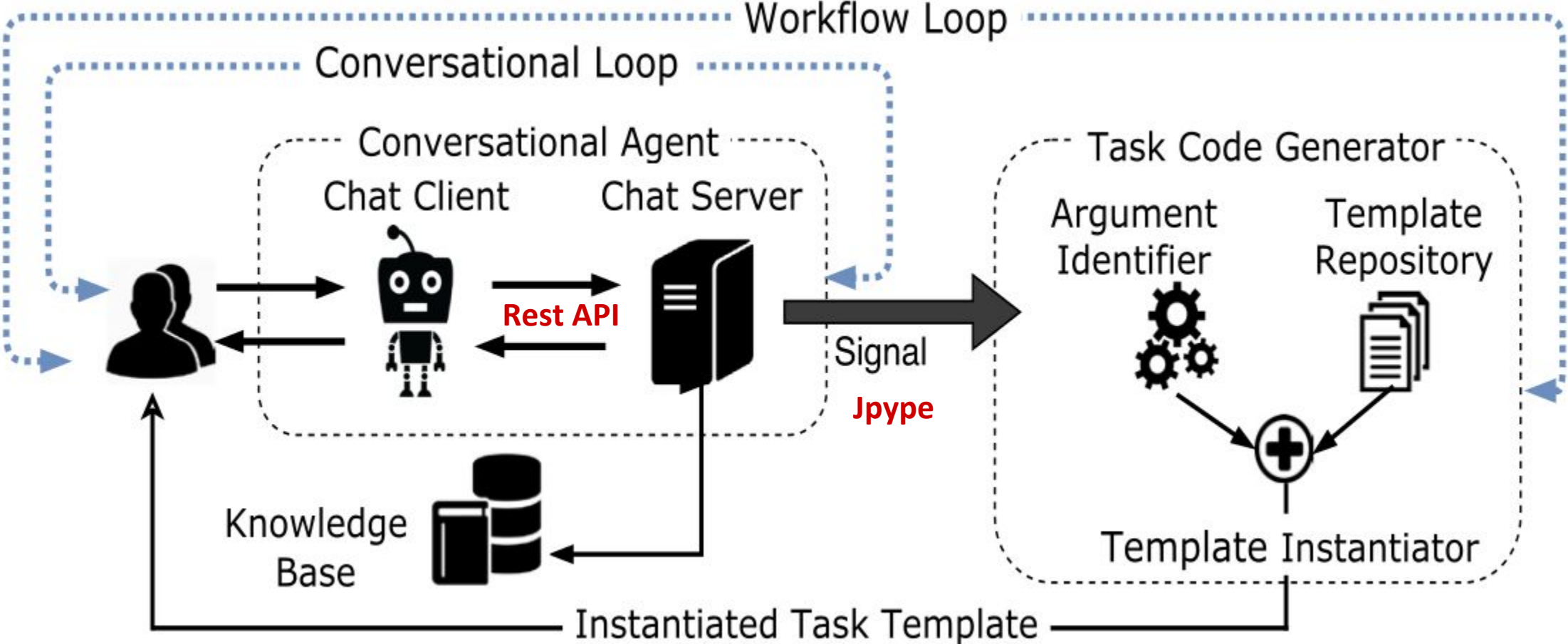
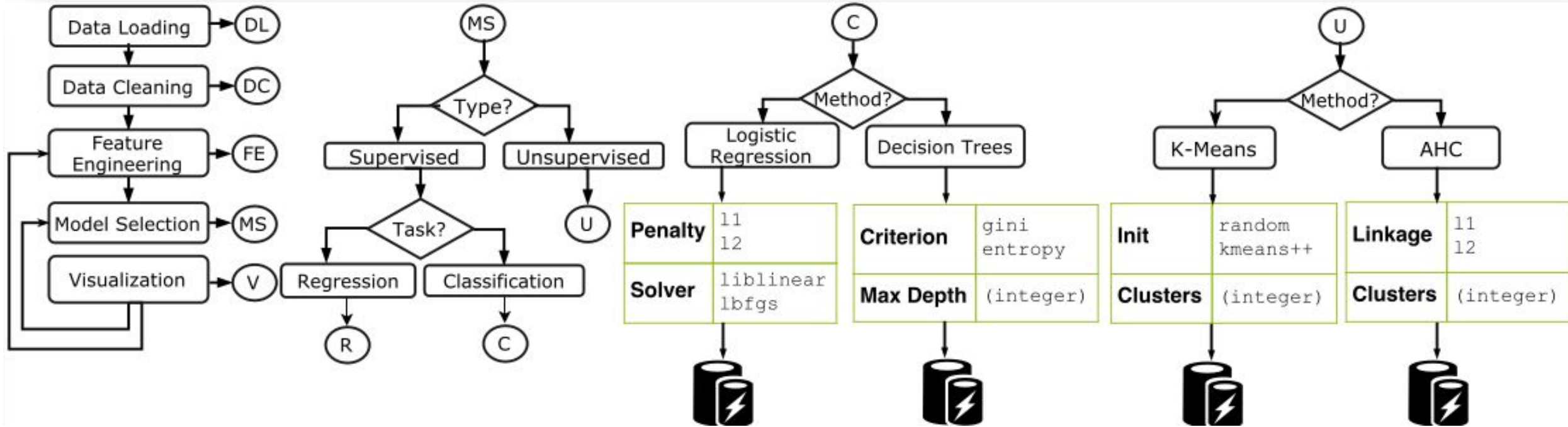


Figure 2: A timeline of an actual Ava conversation for the data science task in Example 1.

Architecture



The Ava Storyboard Concept ...



Results

A group of 16 students with some ML background (via coursework) and Python proficiency were asked to do supervised learning on a Kaggle Dataset.

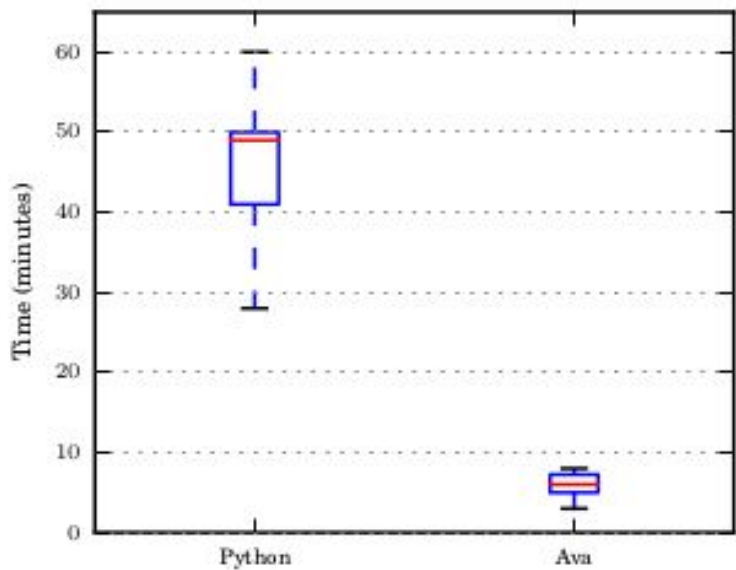


Figure 6: Distribution of the time taken by participants to complete the first model.

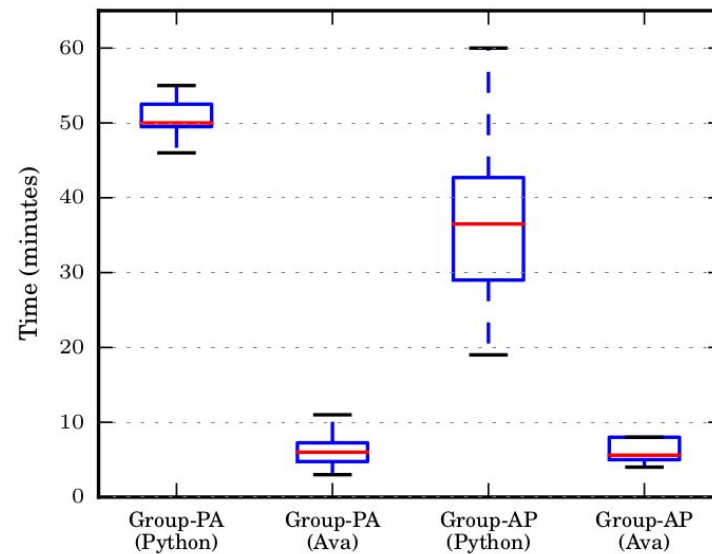


Figure 7: Distribution of the time taken by participants to complete the first model.

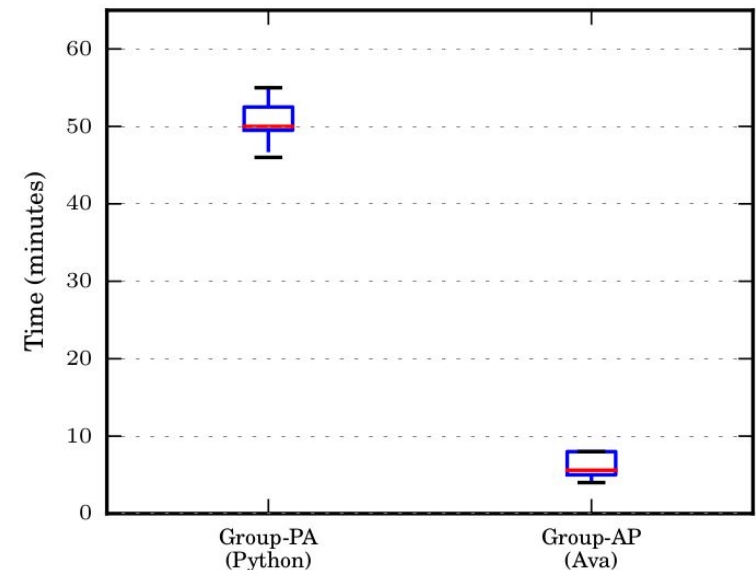


Figure 8: Distribution of unbiased task completion times.

Issues and Enhancements

- ❖ Accuracy of the AVA models versus human models
- ❖ The addition of templates to the repository can be automated.
- ❖ Work on the knowledge-base based recommendation system?
- ❖ Handling unstructured data:
 - A customizable file-parser
- ❖ Handling larger than memory input data
- ❖ Uncertainty quantification in the output as a model guideline
- ❖ Where is the Code?