Predicting Service Disruptions

Hello!

Zia Khan

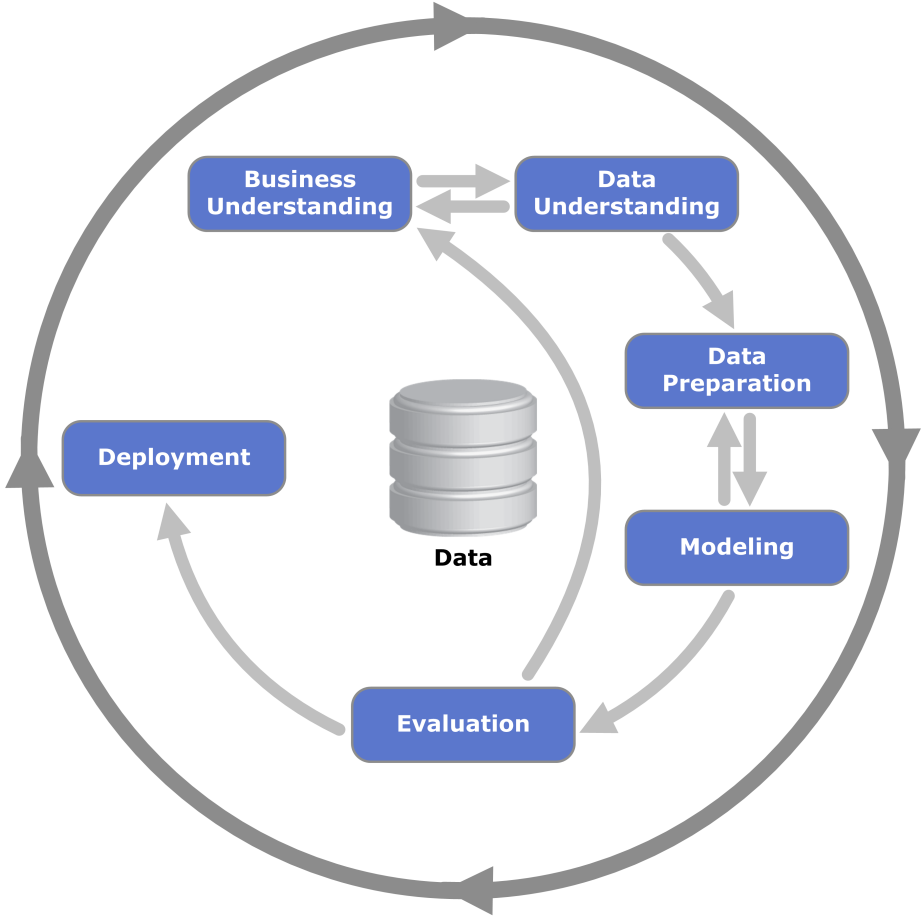
zia@thedevmasters.com

CRISP-DM	1
Business Objective	2
Data Understanding	3
Data Preparation	4
Modeling	5

CRISP-DM

Overview

- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- Evaluation
- Deployment



CRISP-DM 1

Business Objective 2

Data Understanding 3

Data Preparation 4

Modeling 5

2. Business Objective

A telecom company is interested in developing an advanced predictive model to predict service disruptions based on the log files generated by multiple devices.

We have to output a csv file that can be handed to the operations team so they can priorities dispatch of technicians based on fault_severity prediction and its probability.

CRISP-DM 1

Business Objective 2

Data Understanding 3

Data Preparation 4

Modeling 5

Data Understanding

The data set is in a relational format, split among multiple files. The following provides a description of data in each file.

- **Event Type Data**
- **Log Feature Data**
- **Resource Type Data**
- **Severity Type Data**
- **Training Data**

Data Fields	Definition
ID	identifies a unique location-time point
Location	identifier of location
Fault_Severity	categorical. 0: no fault, 1: a few faults, 2: many faults

Data Understanding

Data Dictionary

- Event Type Data

Data Fields	Definition
ID	identifies a unique location-time point
EventType	type of event that occured at that ID (can be multiple events per ID)

- Log Feature Data

Data Fields	Definition
ID	identifies a unique location-time point
Log_Feature	type of feature logged for that ID
Volume	number of times the feature was logged for that ID

- Resource Type Data

Data Fields	Definition
ID	identifies a unique location-time point
Resource_Type	type of resource assocaited with that ID

- Severity Type Data

Data Fields	Definition
ID	identifies a unique location-time point
Severity_Type	type of severity level logged for that ID

CRISP-DM	1
Business Objective	2
Data Understanding	3
Data Preparation	**4**
Modeling	5

Data Preparation

- Step 1: Import Modules
- Step 2: Import Datasets

Each row in the main dataset (train.csv, test.csv) represents a location and a time point. They are identified by the "id" column, which is the key "id" used in other data files. Fault severity has 3 categories: 0,1,2 (0 meaning no fault, 1 meaning only a few, and 2 meaning many). “fault_severity” is a measurement of actual reported faults from users of the network and is the target variable (in train.csv).

	id	location	fault_severity
0	14121	location 118	1
1	9320	location 91	0
2	14394	location 152	1

	id	location
0	11066	location 481
1	18000	location 962
2	16964	location 491

Data Preparation

- Step 3: Data preprocessing
- Step 4: Data merging to create a single record (CAR)

	id	event_type	resource_type	severity_type	log_feature	volume
0	6597	event_type 11	resource_type 8	severity_type 2	feature 68	6
1	8011	event_type 15	resource_type 8	severity_type 2	feature 68	7
2	2597	event_type 15	resource_type 8	severity_type 2	feature 68	1
3	5022	event_type 15	resource_type 8	severity_type 1	feature 172	2
4	5022	event_type 15	resource_type 8	severity_type 1	feature 56	1

Data Preparation

- Step 5: Remove text from variables

  Features are extracted from log files and other sources: event_type.csv, log_feature.csv, resource_type.csv, severity_type.csv. All above features are categorical except for "volume".

	id	event_type	resource_type	severity_type	log_feature	volume
0	6597	11	8	2	68	6
1	8011	15	8	2	68	7
2	2597	15	8	2	68	1
3	5022	15	8	1	172	2
4	5022	15	8	1	56	1

Data Preparation

- Step 6: Drop ”fault_severity” from train dataset as it is the target variable

	id	location
0	14121	location 118
1	9320	location 91
2	14394	location 152
3	8218	location 931
4	14804	location 120

	id	location
0	14121	118
1	9320	91
2	14394	152
3	8218	931
4	14804	120

Data Preparation

- Step 7: Merge the train dataframe without the "fault_severity" column and the combined dataframe of "event_type … etc"

	id	location	event_type	resource_type	severity_type	log_feature	volume
0	14121	118	34	2	2	312	19
1	14121	118	34	2	2	232	19
2	14121	118	35	2	2	312	19
3	14121	118	35	2	2	232	19
4	9320	91	34	2	2	315	200

Data Preparation

- Step 8: Create dummy variables
- Step 9: groupby “id”

	id	0	1	2	3	4	5	6	7	8	...	1572	1573	1574	1575	1576	1577	1578	1579	1580	vol
0	14121	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	76
1	9320	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	632
2	14394	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	4
3	8218	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	44
4	14804	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	96

CRISP-DM	1
Business Objective	2
Data Understanding	3
Data Preparation	4
Modeling	**5**

Modeling

Algorithm Selection

A wide class of models can be used for Network Disruption Prediction

- Logistic Regression
- Random Forest
- Gradient Boosting
- K-Nearest Neighbors
- Decision Trees

Among Ensemble Methods, we have chosen **Gradient Boosting Classifier** because generally its better performance.

Gradient Boosting Algorithm

The overall parameters can be divided into 3 categories:

- **Tree-Specific Parameters:** These affect each individual tree in the model
  Min_samples_split, min_samples_leaf, min_weight_fraction_leaf, max_depth, max_leaf_nodes, max_features

- **Boosting Parameters:** These affect the boosting operation in the model
  Learning_rate, n_estimators, subsample

- **Miscellaneous Parameters:** Other parameters for overall functioning
  Loss, init, random_state, verbose, warm_start, presort

Prediction Probability

Prediction probability

```
1 y_pred_proba_gbc = gbcmodel.predict_proba(X_test)
2
3 y_pred_proba_rf = rfmodel.predict_proba(X_test)
```

Output file

Creating a dataframe with predictions and probability of prediction

```
result = pd.DataFrame({
        "id": X_test.id,
        "Predicted fault_severity": y_predgbc,
        "prediction_probability_0": y_pred_proba_gbc[:,0],
        "prediction_probability_1": y_pred_proba_gbc[:,1],
        "prediction_probability_2": y_pred_proba_gbc[:,2]
    },columns=['id','Predicted fault_severity','prediction_probability_0','prediction_probab

result.head()
```

	id	Predicted fault_severity	prediction_probability_0	prediction_probability_1	prediction_probability_2
2309	6120	1	0.424851	0.516432	0.058717
6089	16118	0	0.824986	0.153196	0.021818
824	2214	1	0.239115	0.489265	0.271619
6519	17291	0	0.923364	0.058295	0.018341
5860	15512	0	0.608872	0.342786	0.048342

Thanks!

Any questions?

zia@thedevmasters.com