Machine Learning Engineer Nanodegree

Capstone Proposal

Srinivasa Amirapu
July 24th, 2018

Proposal

Telstra Network Disruptions

The goal of the problem is to predict Telstra network's fault severity at a time at a particular location based on the log data available. Using a dataset of features from their service logs, you're tasked with predicting if a disruption is a momentary glitch or a total interruption of connectivity

Personal Motivation:

This Kaggle challenge was crafted by Telstra as a simulation of the type of their production problem. Enhancing customer experience is major factor for success of any organization. This could drive customer advocacy by developing a more advanced predictive model for service disruptions and can help to better serve customers by offering them free or discounted offers on the data. Since this problem is closest to production problem, I am motivated to build predictive model for service disruptions which will help companies to provide better customer service and reduce customer churn.

References:

1) https://www.kaggle.com/c/telstra-recruiting-network

2) https://www.telstra.com.au/consumer-advice/customer-service/mass-service-disruption

3) http://gereleth.github.io/Telstra-Network-Disruptions-Writeup/

Domain Background

Telstra is on a journey to enhance the customer experience - ensuring everyone in the company is putting customers first. In terms of its expansive network, this means continuously advancing how it predicts the scope and timing of service disruptions. Telstra wants to see how you would help it drive customer advocacy by developing a more advanced predictive model for service disruptions and to help it better serve its customers.

Problem Statement

The task is to predict the severity of service disruptions on Telstra's network. It is a multi-class classification problem to be evaluated on multiclass logloss to predict 3 categories of faults: 0, 1, 2 (0 meaning no fault, 1 meaning only a few, and 2 meaning many) at a time at a particular location based on the log data available. In other words, we are supposed to predict if a disruption occurred is a momentary glitch or total interruption of connectivity. To make this easy, a data set of service log is provided.

Further research in the internet show below parameters can potentially influence service disruptions:

Location – Some locations might be more prone to failure than other

Weather – There are higher chances of failure in overcast or rainy conditions

Time of the Day – Failure chances may be higher during peak hours when there is high traffic on network

Maintenance Activity – Telstra might be running some background checks and analysis which can cause the network to hang up momentarily.

Power Failure – Though servers always have power backups, but they might fail in extreme conditions

Server Breakdown – Hardware fault

Natural Disaster – Disasters like earthquakes can severely impact connectivity

Number of Users at a Location – Areas with high number of users might be better connected as

Will perform feature engineering on the data set provided with relevant features discussed above to build better model.

Datasets and Inputs

Datasets can be downloaded from Kaggle: https://www.kaggle.com/c/telstra-recruiting-network/data

•train.csv - the training set for fault severity

•test.csv - the test set for fault severity

•sample_submission.csv - a sample submission file in the correct format

•event_type.csv - event type related to the main dataset

•log_feature.csv - features extracted from log files

•resource_type.csv - type of resource related to the main dataset

•severity_type.csv - severity type of a warning message coming from the log

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).Different types of features are extracted from log files and other sources: event_type.csv, log_feature.csv, resource_type.csv, severity_type.csv.Note: “severity_type” is a feature extracted from the log files (in severity_type.csv). Often this is a severity type of a warning message coming from the log. "severity_type" is categorical. It does not have an ordering. “fault_severity” is a measurement of actual reported faults from users of the network and is the target variable (in train.csv).

Solution Statement

The goal of the problem is to predict Telstra network's fault severity at a time at a particular location based on the log data available.

For training models I will compare SVC,LogisticRegression,GradientBoostClassifier since this is a multi-class classification problem. Finally I will select the best model for this problem and fine tune parameters to get best accuracy by using gridSearchCV.

Benchmark Model

Algorithm Selection

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

- Generalized Linear Models

- Support Vector Machines

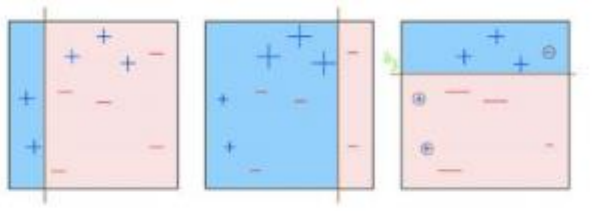
- K-Nearest Neighbors

- Naïve Bayes

- Decision Trees and Ensemble Methods

Among Ensemble Methods, we have chosen Gradient Boosting Classifier as benchmark model because of its better performance and accuracy.

Boosting is a sequential technique which works on the principle of ensemble. It combines a set of weak learners and delivers improved prediction accuracy.



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

Evaluation Metrics

Model accuracy can be evaluated using the multi-class logarithmic loss. Each data row has been labeled with one true class. For each row, you must submit a set of predicted probabilities (one for every fault severity). The formula is then,

logloss=−1N ∑ i=1 N ∑ j=1 M y ij log(p ij),

logloss=−1N∑i=1N∑j=1Myijlog(pij),

where N is the number of rows in the test set, M is the number of fault severity classes, log

log is the natural logarithm, y ij is 1 if observation i

belongs to class jj and 0 otherwise, and p ij

pij is the predicted probability that observation i belongs to class j

The submitted probabilities for a given row are not required to sum to one because they are rescaled prior to being scored (each row is divided by the row sum). In order to avoid the extremes of the log function, predicted probabilities are replaced with max(min(p,1−10 −15),10 −15)

Project Design

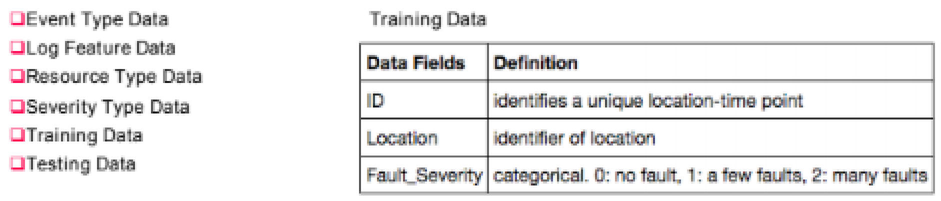
Programming language: *Python 3.0*

Libraries *:Scikit-Learn*

WorkFlow:

1)Data Understanding

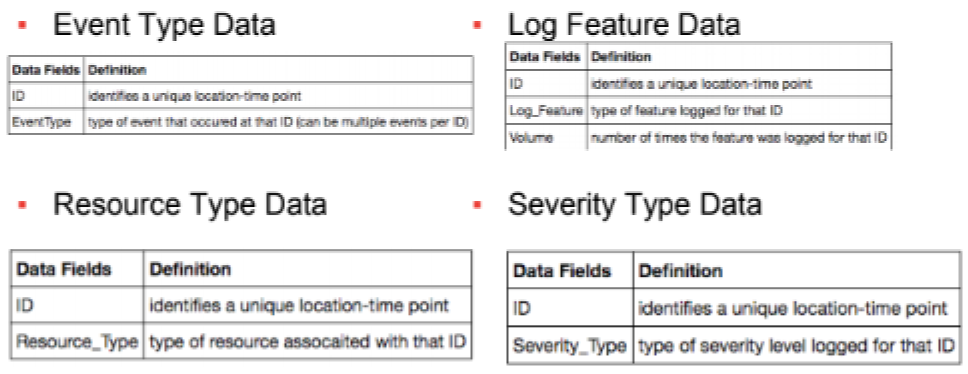
The data set is in a relational format, split among multiple files. The following provides a description of data in each file.



- ❑Event Type Data
- ❑Log Feature Data
- ❑Resource Type Data
- ❑Severity Type Data
- ❑Training Data
- ❑Testing 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



- 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

2) Data Preparation

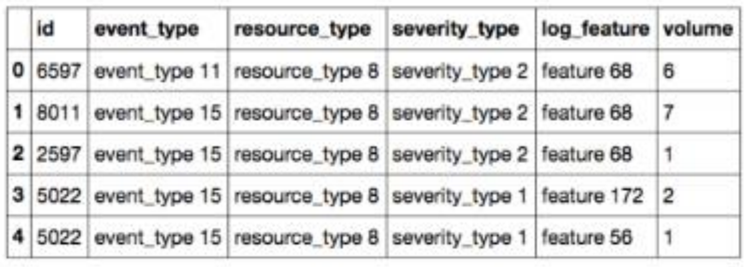
Step 1: Import Modules

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).

Step 3: Data preprocessing

Step 4: Data merging to create a single record

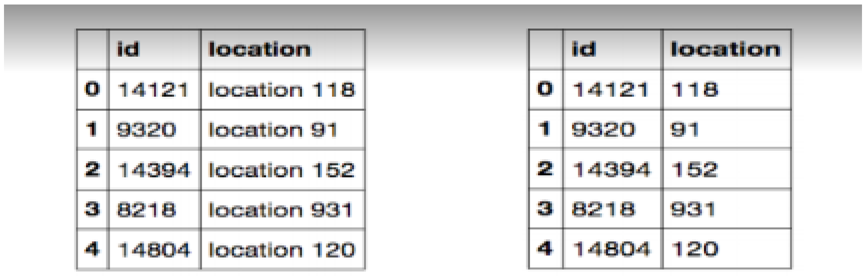


	id	event_type	resource_type	severity_type	log_feature	volume
0	6597	event_type 11	resource_type 8	severity_type 2	feature 68	6
1	8011	event_type 15	resource_type 8	severity_type 2	feature 68	7
2	2597	event_type 15	resource_type 8	severity_type 2	feature 68	1
3	5022	event_type 15	resource_type 8	severity_type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

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".

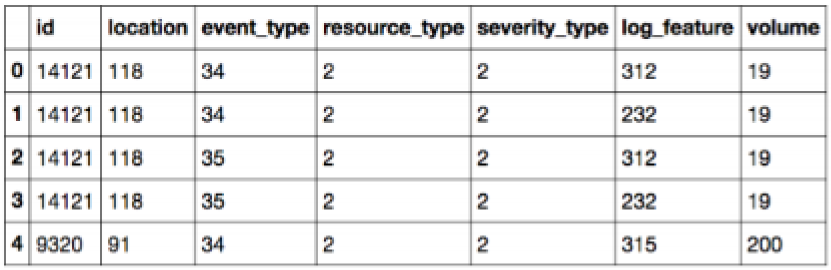
Step 6: Drop ”fault_severity” from train dataset as it is the target variable



	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

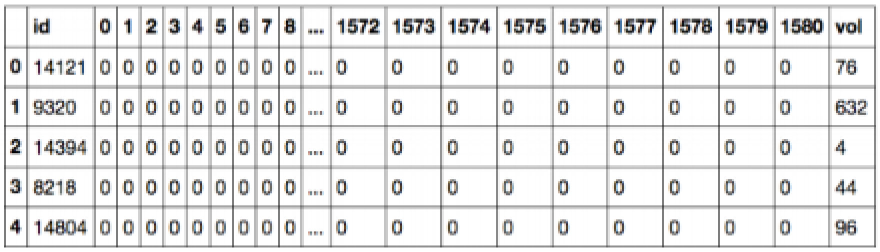
Step 7: Merge the train dataframe without the “fault_severity” column and the combined dataframe of “event_type … etc”



	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

Step 8: Convert Categoricals using Get_Dummies

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

3) Split data using train_test_split method from sklearn.model_selection

4) Model Selection: Choose the best model with best accuracy (fbeta_score) and performance. Among Ensemble Methods, we have chosen Gradient Boosting Classifier as benchmark model because of it generally has better performance and accuracy.

5) Model Training: Train the gradient boost classifier model using optimal parameters (best fit model) obtained from GridSearchCV technique.

6) Predict: Calculate predicted values along with probabilities for test data set.

7) Measure accuracy: Calculate Precision Recall metrics for multi-class classification problem.