**Step 1: Approach used:**

* Based on the data explanation “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.” It was assumed that ‘id’ referred to the time point. This was then used to link and group the remaining features in the other files
* Created ‘examine\_data.py’ to examine all data files to get an overview of available data and possible features
* Created ‘feature\_ extraction.py’ to extract features from data files
* This involved 3 of the ‘log’ files from the dataset (severity, event and resource) and splitting them as separate features based on their feature ids such as severity\_type 1, severity\_type 2. Basic count and frequency metrics by id were also calculated
* Location and log features were treated slightly differently as they seemed to provide richer data. Where possible count, max, min, mean and median features were calculated. An ordering column was added based on the log features ordered by id. Finally the features locations were ranked in ascending and descending order and by relative rank based on location id and log order.
* All the extracted features (~500) were then output to a .csv file called ‘features\_extracted.csv’
* Created the file ‘feature\_imp.py’ in which an XGBoost algorithm was used to rank the features in terms of importance and this was output to another.csv file ('feature\_importance.csv'). In the final model this ranking was used to eliminate roughly half of the features generated
* Finally, wrote a ‘csv\_submission.py ‘file to run the model including 10 fold cross validation and create a .csv file to submit to Kaggle
* Again this involved running an XGBoost algorithm and testing that the logloss was acceptable

**Step 2:**

In addition to my approach further in depth feature engineering could have been employed to useful effect. While I split the features into severity type, event type, resource type and by location id more robust analysis of their aggregated statistics could have proved valuable. I really only performed basic statistical analysis on the supplied files and didn’t ‘sweat’ the features too hard due to time constraints. A less naïve approach would also have looked at feature correlations and any possible lagging effects. An improved approach could look at various data shifts to spot any possible trends in leading or lagging indicators.

Further algorithms such as neural nets, extra tree classifiers and random forest models could all have been tried and tested for improvements. I imagine an sort of ensemble method would have pushed the model accuracy further. However, again due to time constraints these were not pursued.

**Step 3:**

I am assuming this question is unrelated to the Telstra example.

For any new dataset I would talk with the expert to find the key metrics that they employ currently and the type of problems they have in their business case and then design a data based solution. This may vary from measuring active engaged users on a website or product. Trying to predict the path to purchase. Building predictive models of customer behavior or using clustering techniques to build recommender systems to push more product. Every business has a unique problem and is usually seeking to leverage their existing data to gain more insight into how they can better manage their service or product offerings.

Once the available dataset is established and the problem identified the next step is to examine the data type and its dimensionality. Questions I would ask include: Are there lots of missing or incomplete data that need to be dealt with? Are the data interactions linear or non linear? What is the scale? (millions or billions of datapoints). Does the data have high dimensionality such as images or audio files? What is the available training data size? Are there highly correlated variables? Can I easily calculate some basic statistics (mean, mode, median, standard deviation)? Can I plot the data on a graph? Is the dataset balanced, or suffering a class imbalance? Are the variables categorical or continuous?

The answers to all those questions will inform the tools that I would employ to work with the data. I would also look at whether any models I produce are to go straight into production or operate in real time. I would need to speak with the data owner’s engineering team to see what development stack they use and if my solution would be compatible.