TELSTRA CHALLENGE APPROACH

Brief Summary: My approach for Telstra challenge started with understanding the problem from business point of view and identify the possible reasons for the problem they are facing. After this I wanted to become more familiar with the provided datasets and find out which variables are most important for our analysis. I looked for relationships among variables, find missing values and outliers and generate new features. This was the most important step and I came back to this step several times during modeling the data. For modeling, I worked with 3 classifiers i.e. Random Forest, Neural Network and XGBoost. I also performed ensembling of Neural Net and XGBoost model, although the performance of individual XGBoost, having a score of 0.449, was better than the ensembled model.

Below I have provided a detailed description of my approach and what other steps I would have taken for improvement.

Detailed Description:

Step 1: Understanding problem and generate a hypothesis

Client: Telstra, Australia's largest telecommunications and media company

Problem: They want to predict if a disruption occurred is a momentary glitch or total interruption of connectivity. A data set of service log is provided.

I searched for possible reasons for a network failure, which include network congestion, location, server breakdown, weather, etc.

Step 2: Data Exploration

- Target variable: fault_severity having 3 outcomes. So, it's a multi-class classification problem. Classes are imbalanced.
- All the tables have 18552 unique id's which is equal to train and test unique id's combined. So, merge train and test data sets.
- Calculated statistical quantities like min, max, mean, etc. for resource type, event type and log_feature.
- Looked for relationship among variables.
- No missing data – Great!
- Since the train data doesn't directly have any feature apart from location, so we need to create features from other files (for each 'id') and map them back into this file.

Step 3: Data Preparation and Feature Engineering:

- Make sure fault_severity is of integer type.
- Clean data from event_type, resource_type, severity_type and log_features tables and use one-hot encoding to generate sparse matrices.
- Save the order of 'id' as it is same in all tables, seems to be an important feature.
- Concatenate all tables with (train+test) data.
- On sorting the merged dataset by the order of 'id' as saved previously, I found out that the rows of this merged dataset are already sorted by location.

- So, I generated a new feature using cumcount to group them by location and then normalized the count within each location.
- Now, let’s build model on this dataset and then we will see if more feature engineering is needed.

Step 4: Model Building

I implemented 3 different classifiers on the data i.e. Random Forest, Neural Network and XGBoost.

My first choice was to build a **Random Forest classifier** because it is fast and fairly insensitive to tuning. I have used *sklearn.ensemble.RandomForestClassifier* with default parameters and took an average of score of 10 classifiers. I achieved a score of 0.53, which is neither too bad nor too good.

Next, I planned to use **Neural Networks** because they can solve really interesting problems once they are trained. I have used *sklearn.neural_network.MLPClassifier* with ‘logistic’ activation function, ‘sgd’ solver and learning_rate_init having value of 0.001. There was overfitting to the model as I achieved a score of 0.801

Improvement: Need to add regularization to cost function to reduce overfitting to the training data.

Now, it was clear that I should use boosting techniques to achieve better results. So, I used **XGBoost** classifier for building the model with “multi:softprob” as loss function and “mlogloss” as evaluation metric, since it is a 3 class classification problem.

For tuning the parameters of XGB model, I have used Hyperopt library for minimizing the objective function using tpe.suggest algorithm. The parameters that I tuned included ‘eta’ (i.e. learning rate) and ‘max_depth’. We can also tune other parameters like ‘subsample’, ‘colsample_bytree’, ‘lambda’ and ‘alpha’.

As expected XGBoost outperformed both Random Forest and Neural Network by giving a score of 0.449.

Step 5: Ensembling

Now I was interested to see if we can improve the performance by combining the XGBoost and Random Forest Classifier. So, I took the average of predictions of both models and achieved a score of 0.455. A better approach for ensembling could have been to build different models using different parameters and different seeds for XGBoost and then average their results.

Additional steps for improvement:

1. We can perform more statistical operations on the data and search for any other hidden pattern inside the data apart from the location. Enriching the dataset by combining data from other sources is also an option. If we can find any past analysis related to fault severity and location, then it can be beneficial for our problem.

2. Tune regularization parameters like lambda, alpha for XGBoost which can help reduce model complexity, prevent overfitting and enhance performance. The tree-specific parameters such as max_depth, min_child_weight have the highest impact on model outcome.

3. We can use other algorithms such as Logistic Regression, k-Nearest Neighbors and Support Vector Machine to train the model and if results are good enough then ensemble the results with XGBoost. We can also assign different weights to different models while ensembling. Blending predictions from different runs is also an option.

4. We can also use Stacking on the above-mentioned classifiers to see if there is any further improvement. Remember to use out-of-fold predictions, else there will be serious over-fitting.

General Approach for Data Exploration

Data exploration is the most important step in a data analysis process. A significant amount of time should be spent on exploration and analyzing data, once you have got your business hypothesis ready.

Step 1: First, identify Predictor (Input) and Target (output) variables. Next, identify the data type and category of the variables.

Step 2: Explore the variables one by one. For continuous variables, we need to understand central tendency and spread of the variable using Boxplots and Histograms. For categorical variables, we can use Bar Charts to understand distribution of each category.

Step 3: Next, we need to find out the relationship between two or more variables. This includes finding correlation among variables using Scatter Plots and performing statistical tests such as t-test, ANOVA, chi-square test.

Step 4: Now comes the most important step i.e. Missing Value Treatment. Missing data can lead to a biased model and reduce the fit of the model. There are several ways to treat the missing values depending on certain conditions:

i. List-wise Deletion
ii. Pair-wise Deletion
iii. Mean/Median/Mode Imputation
iv. kNN Imputation
v. Creating Prediction Model to estimate missing values from available values

Step 5: Outlier Treatment is as equally important as treating missing values. Outliers can be detected using boxplots, scatter plots and histograms. Generally, data points that are 3 or more standard deviations away from mean (i.e. outside 5th and 95th percentile) are considered outliers. There are several ways to treat the missing values depending on certain conditions:

i. Deleting Observations
ii. Mean/ Median/Mode Imputation
iii. Transforming and binning the values
iv. Treating Separately

Step 6: The art of extracting more information from existing data is what makes a Data Scientist differ from a Data Analyst. Although, feature engineering is the final step in data exploration but you will need to come back again and again to this step to refine your model. This step involves two parts i.e. Variable Transformation and Feature Creation. Variable Transformation is needed when we want to change the scale of variables or transform a complex non-linear relationship into linear relationship. It can be performed by taking the log, square/cube root or binning the values. Creating new features includes methods like converting categorical variables into dummy numerical variables.

General Approach for Data Analysis

Step 1: The very first thing I would do is clearly define the problem or objective so I have a solid direction. We need to understand the project objectives and requirements from a business perspective, and then convert this knowledge into a data mining problem definition.

Step 2: The next step is to understand and explore the dataset. This involves steps such as Variable Identification, Uni-variate and Bi-variate Analysis, Missing value treatment, Outlier detection, Variable Transformation and Feature Creation. We can also enrich our dataset with other data sources.

Step 3: After processing the data, we are now ready to build model. In this phase, various modeling techniques are selected and applied, and their parameters are calibrated to optimal values. We often need to step back to the data preparation phase based on the performance and results. Ensembling and Stacking can be applied on the top classifiers. We should keep modeling the data until we find most significant and valuable results.

Step 4: Before proceeding to final deployment, the model should be thoroughly evaluated and it should be made sure that it properly achieves the business objectives.

Step 5: Lastly, I would implement the model and track my results.

In my opinion the first two steps are the most important steps in a data analysis process. If you are not clear with the project requirements, then you are just wasting your time. And certainly, good data is more worth than a good algorithm. So, more time should be spent on feature engineering rather than trying bunch of different algorithms to get good results.