StumbleUpon Evergreen Classification Challenge

My approach to solve the problem was to minimize the amount of data being input to the model based on its importance. Hence, I decided to make the predictions based on just the boilerplate data and ignore most of the other columns as textual data is in case more relevant than any other data. So, I used the text data and split it into training and validation data. The next step was to clean the data and then tokenize it. I cleaned and tokenized the data by taking vocabulary of 20000 words to remove words which do not occur very often using the tensorflow tokenizer library and the python nltk library. After which I used **Bidirectional LSTM layers in my model along with dense layers**. I also added a dropout layer to drop some of the units in order to avoid overfitting. I used RMSProp optimizer which has momentum, learning rate and rho attributes which can be tuned. I used accuracy, AUC, precision and recall as metrics to evaluate the model.

I achieved the following results with an epoch of 25.

Training data :

- Accuracy: 0.9238
- AUC: 0.9238
- Precision_2: 0.9303
- Recall_2: 0.9209

Validation Data :

- Accuracy: 0.7654
- AUC: 0.7682
- Precision_2: 0.7620
- Recall_2: 0.7813