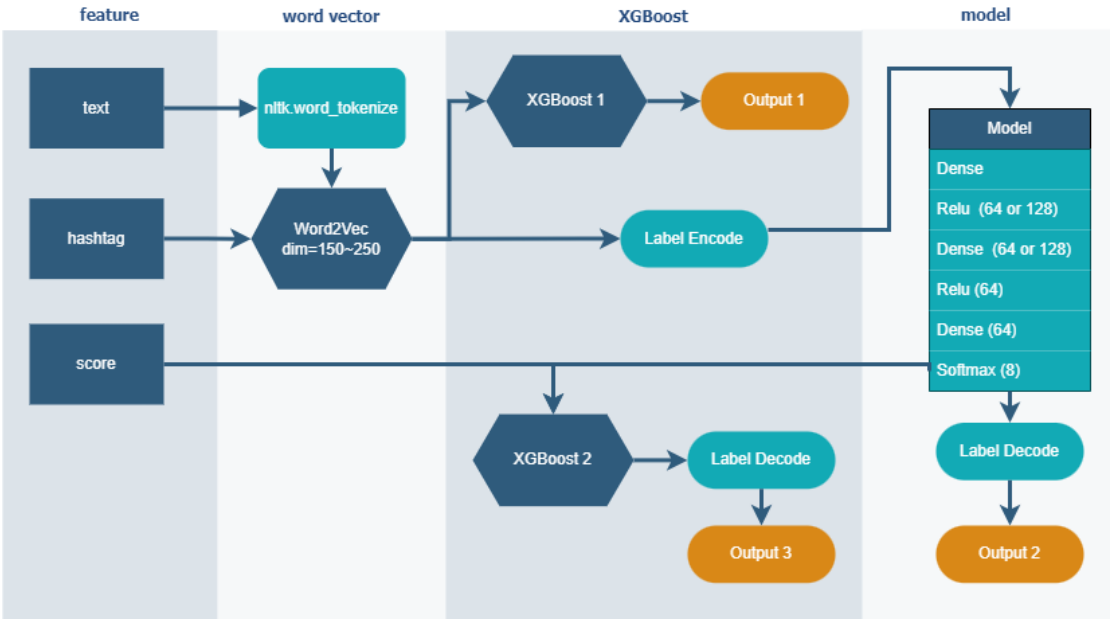
Name: 鄭竹淇

Student ID: 109020032

Feature engineering and Model



split into train data, test data, validation data

```
df_merg = pd.merge(df, data_ident, on=['tweet_id', 'tweet_id'])
train_data = pd.merge(df_merg[df_merg["identification"]=='train'], emotion, on=['tweet_id', 'tweet_id'] )
test_data = df_merg[df_merg["identification"]=='test']

x_train, x_val, y_train, y_val = train_test_split(train_data.loc[:,['score','hashtags','text_token']],
                                                  train_data['emotion'], test_size=0.2, random_state=33
                                                  )
```

get feature(text & hashtags)

```
def get_sent_vec(model, text_list):
        len_text = len(text_list)
        sentence_vector = np.zeros(dim_size)
        if len_text ==0:
                return sentence_vector
        for word in text_list:
                if word in model.wv:
                        sentence_vector += model.wv[word]
        sentence_vector = sentence_vector/len_text
        return sentence_vector

def get_feature(data, alpha=0.4):
        data['sent_vec'] = data['text_token'].apply(lambda x: get_sent_vec(word2vec_model,x))
        data['tags_vec'] = data['hashtags'].apply(lambda x: get_sent_vec(word2vec_model,x))
        return alpha*data['sent_vec']+(1-alpha)*data['tags_vec']
```

Conclusion and Findings:

The optimal Word2Vec dimension is 250, and the best accuracy is achieved when the alpha value (proportion of text vectors in features) is set to 0.4. However, when the data is further processed through the XGBoost2 model, the accuracy tends to decrease. This could be attributed to the complexity of the model being too high.