Untitled

September 29, 2020

[3]:
```
data = pd.read_csv("stack overflow data.csv")
```

[4]:
```
display(data.info(),data.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60000 entries, 0 to 59999
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Id            60000 non-null  int64 
 1   Title         60000 non-null  object
 2   Body          60000 non-null  object
 3   Tags          60000 non-null  object
 4   CreationDate  60000 non-null  object
 5   Y             60000 non-null  object
dtypes: int64(1), object(5)
memory usage: 2.7+ MB
```

```
None
```

```
         Id                                              Title  \
0  34552656             Java: Repeat Task Every Random Seconds   
1  34552974  How to get all the child records from differen...   
2  34553034                  Why are Java Optionals immutable?   
3  34553174  Text Overlay Image with Darkened Opacity React...   
4  34553318         Why ternary operator in swift is so picky?   

                                                Body  \
0  <p>I'm already familiar with repeating tasks e...   
1  I am having 4 different tables like \r\nselect...   
2  <p>I'd like to understand why Java 8 Optionals...   
3  <p>I am attempting to overlay a title over an ...   
4  <p>The question is very simple, but I just cou...   

                                     Tags         CreationDate  \
0                   <java><repeat>  2016-01-01 00:21:59   
1                 <sql><sql-server>  2016-01-01 01:44:52   
2                  <java><optional>  2016-01-01 02:03:20   
```

```
3  <javascript><image><overlay><react-native><opa...  2016-01-01 02:48:24
4  <swift><operators><whitespace><ternary-operato...  2016-01-01 03:30:17

          Y
0  LQ_CLOSE
1   LQ_EDIT
2         HQ
3         HQ
4         HQ
```

[5]:
```
dummy_data = data[['Title','Body','Tags','Y']]
dummy_data['Y'] = dummy_data['Y'].map({'LQ_CLOSE':0, 'LQ_EDIT':1, 'HQ':2})
dummy_data.head()
```

[5]:
```
                                                Title  \
0             Java: Repeat Task Every Random Seconds
1  How to get all the child records from differen…
2                  Why are Java Optionals immutable?
3  Text Overlay Image with Darkened Opacity React…
4         Why ternary operator in swift is so picky?

                                                 Body  \
0  <p>I'm already familiar with repeating tasks e…
1  I am having 4 different tables like \r\nselect…
2  <p>I'd like to understand why Java 8 Optionals…
3  <p>I am attempting to overlay a title over an …
4  <p>The question is very simple, but I just cou…

                                                 Tags  Y
0                                   <java><repeat>  0
1                                 <sql><sql-server>  1
2                                  <java><optional>  2
3  <javascript><image><overlay><react-native><opa…  2
4  <swift><operators><whitespace><ternary-operato…  2
```

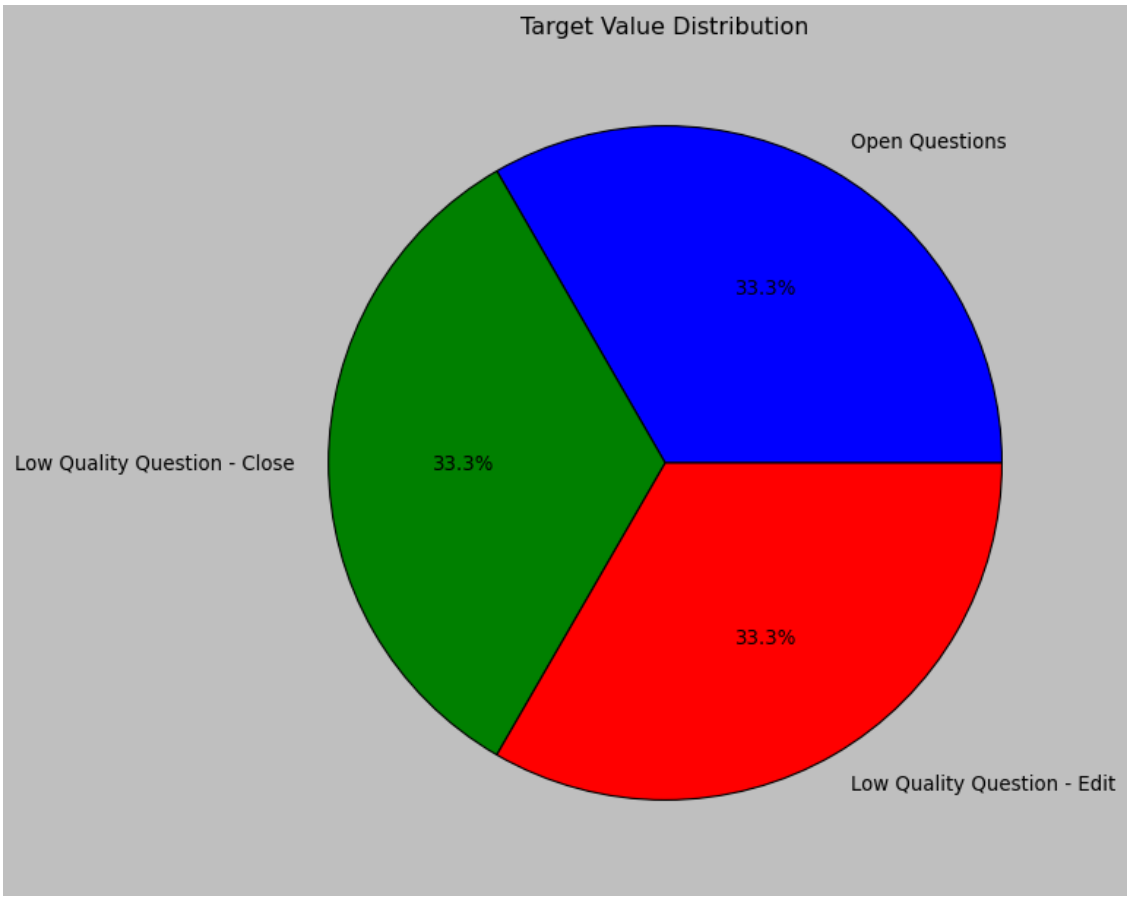
[6]:
```
data = data.drop(['Id', 'Tags', 'CreationDate'], axis=1)
data['Y'] = data['Y'].map({'LQ_CLOSE':0, 'LQ_EDIT': 1, 'HQ':2})
data.head()
```

[6]:
```
                                                Title  \
0             Java: Repeat Task Every Random Seconds
1  How to get all the child records from differen…
2                  Why are Java Optionals immutable?
3  Text Overlay Image with Darkened Opacity React…
4         Why ternary operator in swift is so picky?

                                                 Body  Y
```

```
0  <p>I'm already familiar with repeating tasks e…  0
1  I am having 4 different tables like \r\nselect…  1
2  <p>I'd like to understand why Java 8 Optionals…  2
3  <p>I am attempting to overlay a title over an …  2
4  <p>The question is very simple, but I just cou…  2
```

[7]:
```
labels = ['Open Questions', 'Low Quality Question - Close', 'Low Quality␣
 ↪Question - Edit']
values = [len(data[data['Y'] == 2]), len(data[data['Y'] == 0]),␣
 ↪len(data[data['Y'] == 1])]
plt.style.use('classic')
plt.figure(figsize=(16, 9))
plt.pie(x=values, labels=labels, autopct="%1.1f%%")
plt.title("Target Value Distribution")
plt.show()
```



[8]:
```
data['text'] = data['Title'] + ' ' + data['Body']
data = data.drop(['Title', 'Body'], axis=1)
data.head()
```

[8]:

	Y	text
0	0	Java: Repeat Task Every Random Seconds <p>I'm …
1	1	How to get all the child records from differen…
2	2	Why are Java Optionals immutable? <p>I'd like …
3	2	Text Overlay Image with Darkened Opacity React…
4	2	Why ternary operator in swift is so picky? <p>…

[9]:
```
def clean_text(text):
    text = text.lower()
    text = re.sub(r'[^(a-zA-Z)\s]','', text)
    return text
data['text'] = data['text'].apply(clean_text)
```

[10]:
```
# Define how much percent data you wanna split
split_pcent = 0.20
split = int(split_pcent * len(data))

# Shuffles dataframe
data = data.sample(frac=1).reset_index(drop=True)

# Training Sets
train = data[split:]
trainX = train['text']
trainY = train['Y'].values

# Validation Sets
valid = data[:split]
validX = valid['text']
validY = valid['Y'].values

assert trainX.shape == trainY.shape
assert validX.shape == validY.shape

print(f"Training Data Shape: {validX.shape}\nValidation Data Shape: {validX.
 ↪shape}")
```

```
Training Data Shape: (12000,)
Validation Data Shape: (12000,)
```

[13]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()
trainX = vectorizer.fit_transform(trainX)
validX = vectorizer.transform(validX)
```

[15]:
```
from xgboost import XGBClassifier
xg_classifier = XGBClassifier()
xg_classifier.fit(trainX, trainY)
```

[15]:
```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.300000012, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=nan, monotone_constraints='()',
              n_estimators=100, n_jobs=0, num_parallel_tree=1,
              objective='multi:softprob', random_state=0, reg_alpha=0,
              reg_lambda=1, scale_pos_weight=None, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=None)
```

[16]:
```
print(f"Validation Accuracy of XGBoost Clf. is: {(xg_classifier.score(validX,␣
↪validY))*100:.2f}%")
```

```
Validation Accuracy of XGBoost Clf. is: 87.89%
```