Practical Lessons from Predicting Clicks on Ads at Facebook

GBDT+LR

GBDT

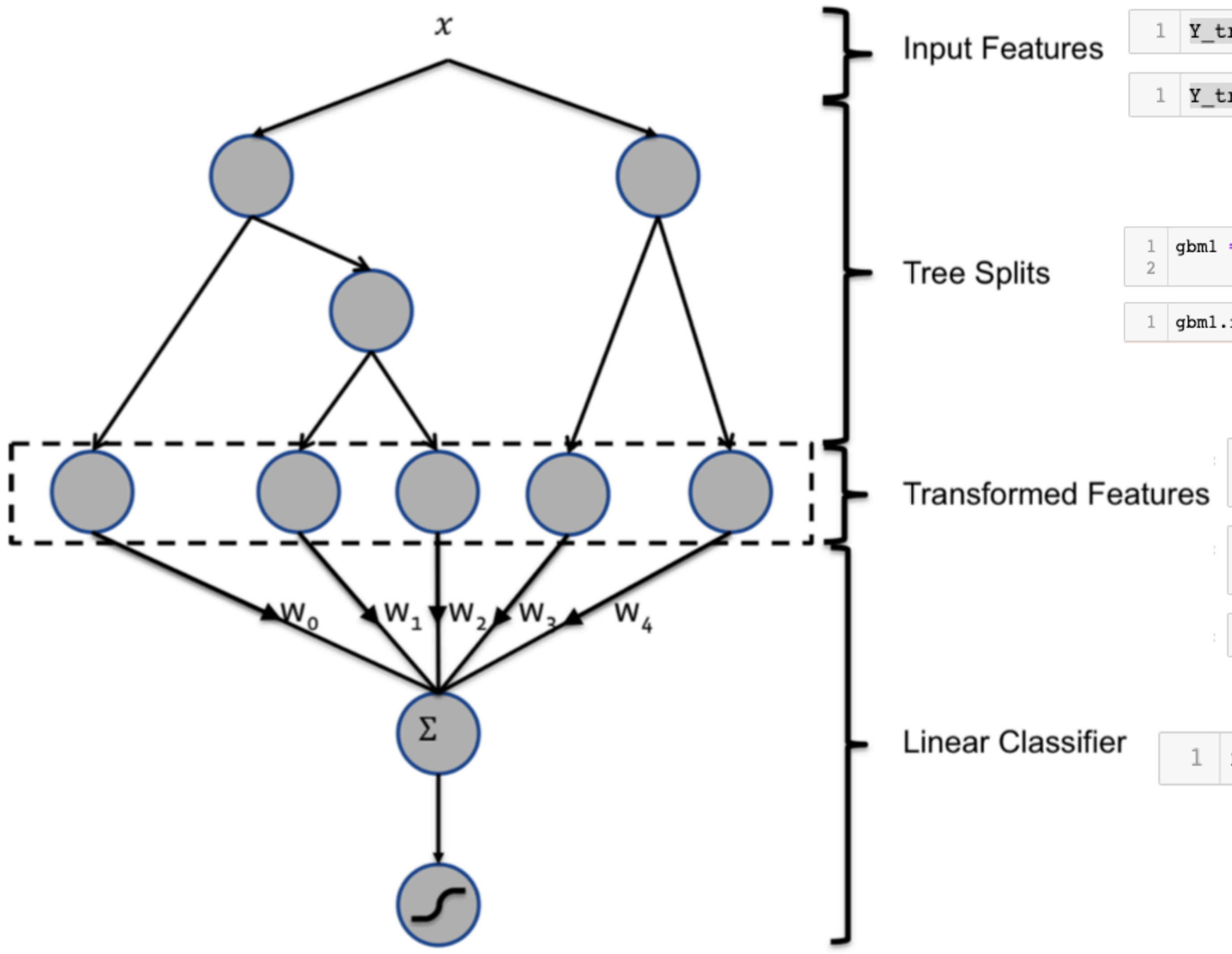
Gradient Boosting Decision Tree

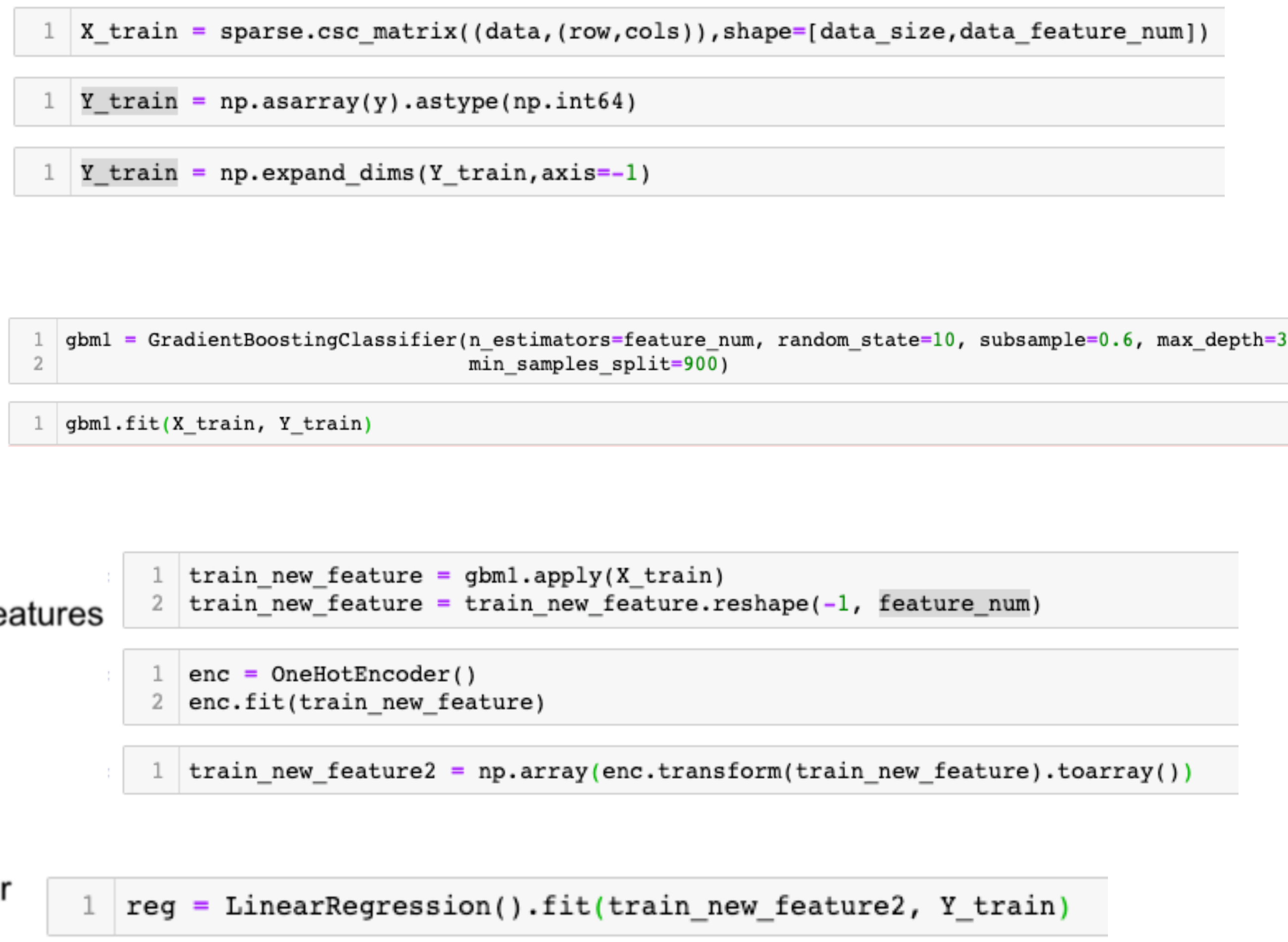
$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F$$

NE

Normalized Entropy

$$NE = \frac{-\frac{1}{N}\sum_{i=1}^{n}\left(\frac{1+y_i}{2}log(p_i) + \frac{1-y_i}{2}log(1-p_i)\right)}{-(p * log(p) + (1-p) * log(1-p))}$$





```
X_train = sparse.csc_matrix((data,(row,cols)),shape=[data_size,data_feature_num])
```

```
Y_train = np.asarray(y).astype(np.int64)
```

```
Y_train = np.expand_dims(Y_train,axis=-1)
```

```
gbml = GradientBoostingClassifier(n_estimators=feature_num, random_state=10, subsample=0.6, max_depth=3
                                   min_samples_split=900)
```

```
gbml.fit(X_train, Y_train)
```

```
train_new_feature = gbml.apply(X_train)
train_new_feature = train_new_feature.reshape(-1, feature_num)
```

```
enc = OneHotEncoder()
enc.fit(train_new_feature)
```

```
train_new_feature2 = np.array(enc.transform(train_new_feature).toarray())
```

```
reg = LinearRegression().fit(train_new_feature2, Y_train)
```