PMDS505P Data Mining and Machine Learning
Experiment 8

February 2025

1 Work to do today

Note: Make a single pdf file of the work you are doing in jupyter notebook. Upload with proper format. Please mention your name and roll no properly with Experiment number in the first page of your submission.

Decision Tree: Gradient boosting

Q1. Today we will try to see how gradient boosting can be implemented both manually and using the inbuilt classes.

- Download the Book1.cs, houseprice data set.
- Perform Min-Max scaling
- Do the train test split of the data with test size 20%.
- Fit the Decision tree model to the training data.
- Print the testing error.

Now we will see how we can implement the gradient boosting technique with inbuilt class.

- Import GradientBoostingRegressor from sklearn.ensemble
- Create an object of same and fit the model with the parameters n_estimators = 100, learning_rate=0.01, max_depth=3 and random_state =0
- Check the error improvement in this case. Compared to the basic DecisionTreeRegressor you have used previously.
- Perform hyperparameter tuning using GridsearchCV by giving an parameter grid with the hyperparameters n_estimators, learning_rate and max_depth. Each parameter should be having ateast 10 values in the parameter grid.

- Similarly you can import GradientBoostingClassifier from sklearn.ensemble. Use the liver patient dataset and fit a Decision tree and GradientBoostingClassfier.

Q2. Perform hyperparameter tuning on with the hyperparameters n_estimators, learning_rate and max_depth. Each parameter should be having ateast 10 values in the parameter grid. Find the best combination of the parameters to get the better accuracy.

Reguarization techniques: Ridge and lasso regression.

Now we will try to look at ridge and lasso regression which are again regularization techniques used to minimize the variance or reduce overfitting of data. The lasso regression also kind of helps to know the best features in the modeling. Because it will take some coefficients of the model which are not that relevant to zero.

Q3. To perform ridge and lasso regression download the Book1.csv dataset to do house price prediction.

- Do the min max scaling of the data and split the data to train and test with 20% testsize.
- Define the grid values for the hyperparameter α and you can fit both the cases as given below and check the coefficients of the models in each case, in ridge as well as in lasso.

```
# Define alpha values for tuning
alpha_values = np.logspace(-2, 4, 100)  # 10 values from 10^-2 to 10^4
print(alpha_values)
# ------------------ Ridge Regression with RidgeCV ------------------
ridge_cv = RidgeCV(alphas=alpha_values, store_cv_values=True)
ridge_cv.fit(X_train, y_train)
ridge_pred = ridge_cv.predict(X_test)
print("Best Ridge Alpha:", ridge_cv.alpha_)
print("Ridge Regression MSE:", mean_squared_error(y_test, ridge_pred))
print("Ridge Coefficients:", ridge_cv.coef_)
# ------------------ Lasso Regression with LassoCV ------------------
lasso_cv = LassoCV(alphas=alpha_values, cv=5, random_state=0)
lasso_cv.fit(X_train, y_train)
lasso_pred = lasso_cv.predict(X_test)
print("Best Lasso Alpha:", lasso_cv.alpha_)
print("Lasso Regression MSE:", mean_squared_error(y_test, lasso_pred))
print("Lasso Coefficients:", lasso_cv.coef_)
```

Figure 1: Enter Caption

Now the coefficients that you see are the coefficients of your multiple regression model you have obtained after applying the modification in the loss functions.