Hyperparameter Optimization: Takeaways

Syntax

- Calculating the pairwise correlation of columns:

```
correlations = banking_df.corr()
```

- Instantiating `GridSearchCV`:

```
grid_params = {"n_neighbors": range(1, 10),
               "metric": ["minkowski", "manhattan"]
               }
knn = KNeighborsClassifier()
knn_grid = GridSearchCV(knn, grid_params, scoring="accuracy")
```

- `GridSearchCV`'s best model:

```
knn_grid.best_estimator_
```

- `GridSearchCV`'s best model's accuracy score:

```
knn_grid.best_score_
```

- `GridSearchCV`'s best model's parameters:

```
knn_grid.best_params_
```

- Evaluating test set using `GridSearchCV`'s best model:

```
knn_grid.best_estimator_.score(X_test_scaled, y_test)
```

Concepts

- **Feature selection** is the process of identifying relevant features on which the model can be trained. It can improve the model's performance.
- Some ways to identify relevant features:
    - Randomly selecting and experimenting with different sets of features.
    - Having domain expertise.
    - Calculating which features strongly correlate to the target variable.
- **Hyperparameters** are parameters that we input or set before training the model. These parameters can influence the training process and have an impact on the model's performance.
- **Hyperparameter Optimization** or **Hyperparameter Tuning** is the process of tuning the hyperparameter values in order to maximize the model's performance.
- **Grid search** is a hyperparameter optimization technique in which we train and evaluate a model on a subset of hyperparameter values in order to identify the values that yield the best performing model.

Resources

- Bank Marketing Dataset

- pandas' get_dummies() function
- pandas' corr() function
- scikit-learn's train_test_split() function
- scikit-learn's MinMaxScaler
- scikit-learn's KNeighborsClassifier
- scikit-learn's GridSearchCV
- scikit-learn's score() function