

Experiment Tracking Report: California Housing Price Prediction using MLFlow

Experiment Overview

This experiment focuses on predicting house prices using the California Housing dataset. Two machine learning models, **Linear Regression** and **Random Forest**, were used to predict the target variable (house prices). MLflow was employed for experiment tracking, enabling the logging of metrics and model parameters.

Dataset

- **Dataset Source:** California Housing dataset from sklearn.
- **Target Variable:** `target` (housing prices).
- **Features:**
 - **MedInc:** Median income of households in a block.
 - **HouseAge:** Median age of houses in a block.
 - **AveRooms:** Average number of rooms per household.
 - **AveBedrms:** Average number of bedrooms per household.
 - **Population:** Block population.
 - **AveOccup:** Average number of household members.
 - **Latitude:** Block latitude.
 - **Longitude:** Block longitude.

Objectives

- Compare the performance of Linear Regression and Random Forest Regressor models.
- Track and log metrics such as **Mean Squared Error (MSE)**, **Mean Absolute Error (MAE)**, and **R-squared (R^2)** using MLflow.

Setup

- **Training-Testing Split:** 80% training, 20% testing.
- **Feature Scaling:** Standardization was applied to numerical features using `StandardScaler`.

Models Trained

1. **Linear Regression**
 - A linear model to predict the target variable based on the features.
2. **Random Forest Regressor**
 - An ensemble-based decision tree model that combines multiple trees to make predictions.

Evaluation Metrics

- **Mean Squared Error (MSE):** Measures the average squared difference between predicted and actual values. Lower is better.

- **Mean Absolute Error (MAE):** Measures the average absolute difference between predicted and actual values. Lower is better.
- **R-squared (R²):** Explains how well the features explain the variance in the target variable. Higher is better, with 1 being a perfect score.

Results

Model	MSE	MAE	R ²
Linear Regression	0.5559	0.5332	0.5758
Random Forest	0.2555	0.3276	0.8050

California Housing Regression >

wise-moose-419

Overview

Model metrics

System metrics

Artifacts

Description

No description

Details

Created at	2024-10-01 16:31:54
Created by	denzel
Experiment ID	912973484134883571
Status	Finished
Run ID	6b04db717a4c4890b73157ba3e729b15
Duration	24.9s
Datasets used	dataset (fd65955b) Eval +1
Tags	estimator_class: sklearn.ensemble_forest.Random... estimator_name: RandomForestRegressor
Source	train.py 9a3c54a
Logged models	sklearn
Registered models	

Parameters (21)		Metrics (3)	
Parameter	Value	Metric	Value
bootstrap	True	training_r2_score	0.9735692800638676
min_weight_fraction_leaf	0.0	training_mean_absolute_error	0.12207767957848875
max_leaf_nodes	None	training_mean_squared_error	0.035332010532530315
verbose	0	training_root_mean_squared_error	0.18796811041378886
random_state	42	training_score	0.9735692800638676
warm_start	False		
max_features	1.0		
n_jobs	None		
oob_score	False		
fit_intercept	True		
max_depth	None		
min_samples_leaf	1		
min_impurity_decrease	0.0		
positive	False		
monotonic_cst	None		
ccp_alpha	0.0		
copy_X	True		
max_samples	None		
n_estimators	100		



Observations:

- **Random Forest** outperforms **Linear Regression** in all evaluation metrics. It has a significantly lower MSE and MAE and a higher R² score, indicating better performance.
- **Linear Regression** shows moderate predictive power but is less robust compared to Random Forest, as indicated by its lower R² score (0.5758) and higher error rates

MLflow Tracking

Each experiment was tracked using MLflow. Key metrics, parameters, and the models were logged, providing insights into the performance of both models over the same dataset.

MLflow Logs:

- **Metrics Tracked:**
 - Mean Squared Error (MSE)
 - Mean Absolute Error (MAE)
 - R² Score
- **Parameters Logged:**
 - Model type (Linear Regression, Random Forest)

- Hyperparameters (Random Forest hyperparameters such as `n_estimators`, `random_state`)

Model Comparison in MLflow:

MLflow allows for easy comparison of runs, enabling us to visualize and compare the performance of the two models. Random Forest clearly provided better generalization and predictive performance across all metrics.

Conclusion

The Random Forest model performed significantly better than the Linear Regression model, reducing prediction errors and achieving a higher R^2 score. It is recommended to use **Random Forest** for future predictions on this dataset.

Appendix: Code Snippets

```
python
Copy code
# Train Linear Regression
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

# Train Random Forest Regressor
rf_model = RandomForestRegressor(random_state=42)
rf_model.fit(X_train, y_train)

# Evaluate models
y_pred_lr = lr_model.predict(X_test)
y_pred_rf = rf_model.predict(X_test)

# Log metrics in MLflow
mlflow.log_metric("mse_lr", mean_squared_error(y_test, y_pred_lr))
mlflow.log_metric("mae_lr", mean_absolute_error(y_test, y_pred_lr))
mlflow.log_metric("r2_lr", r2_score(y_test, y_pred_lr))

mlflow.log_metric("mse_rf", mean_squared_error(y_test, y_pred_rf))
mlflow.log_metric("mae_rf", mean_absolute_error(y_test, y_pred_rf))
mlflow.log_metric("r2_rf", r2_score(y_test, y_pred_rf))
```

This report summarizes the model training process and the results logged with MLflow.