

Regression Algorithms in Machine Learning

By Md Anique Zzama

Introduction to Regression in ML

- Regression is a supervised learning technique used to predict continuous values.

Types of Regression Algorithms

- Common types: Linear Regression, Polynomial Regression, Decision Trees, Random Forest, SVR, Ridge, Lasso, etc.

Step 1: Data Preprocessing

- Handling missing values, encoding categorical variables, feature scaling.
- Code:
- `from sklearn.preprocessing import StandardScaler`
- `scaler = StandardScaler()`
- `X_scaled = scaler.fit_transform(X)`

Step 2: Train-Test Split

- Splitting data into training and testing sets.
- Code:
- `from sklearn.model_selection import train_test_split`
- `X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)`

Linear Regression

- A simple model to find the best-fit line.
- Code:
- `from sklearn.linear_model import
LinearRegression`
- `model = LinearRegression().fit(X_train,
y_train)`

Multiple Linear Regression

- Extends Linear Regression with multiple features.
- Code:
- `from sklearn.linear_model import LinearRegression`
- `model = LinearRegression().fit(X_train, y_train)`

Polynomial Regression

- A nonlinear regression approach.
- Code:
- `from sklearn.preprocessing import PolynomialFeatures`
- `poly = PolynomialFeatures(degree=2)`
- `X_poly = poly.fit_transform(X)`

Decision Tree Regression

- A tree-based algorithm that splits features.
- Code:
- ```
from sklearn.tree import
DecisionTreeRegressor
```
- ```
model = DecisionTreeRegressor().fit(X_train,  
y_train)
```

Random Forest Regression

- An ensemble of decision trees.
- Code:
- `from sklearn.ensemble import
RandomForestRegressor`
- `model = RandomForestRegressor().fit(X_train,
y_train)`

Support Vector Regression (SVR)

- Uses a margin-based approach for regression.
- Code:
- `from sklearn.svm import SVR`
- `model = SVR().fit(X_train, y_train)`

Ridge Regression

- A regularized regression model.
- Code:
- `from sklearn.linear_model import Ridge`
- `model = Ridge().fit(X_train, y_train)`

Lasso Regression

- Performs feature selection by shrinking coefficients.
- Code:
- `from sklearn.linear_model import Lasso`
- `model = Lasso().fit(X_train, y_train)`

Elastic Net Regression

- Combines Ridge and Lasso.
- Code:
- `from sklearn.linear_model import ElasticNet`
- `model = ElasticNet().fit(X_train, y_train)`

Bayesian Ridge Regression

- A probabilistic approach to linear regression.
- Code:
- `from sklearn.linear_model import BayesianRidge`
- `model = BayesianRidge().fit(X_train, y_train)`

Gradient Boosting Regression

- Boosting algorithm combining weak learners.
- Code:
- `from sklearn.ensemble import GradientBoostingRegressor`
- `model = GradientBoostingRegressor().fit(X_train, y_train)`

XGBoost Regression

- Optimized gradient boosting model.
- Code:
- `from xgboost import XGBRegressor`
- `model = XGBRegressor().fit(X_train, y_train)`

LightGBM Regression

- Lightweight boosting model.
- Code:
- `from lightgbm import LGBMRegressor`
- `model = LGBMRegressor().fit(X_train, y_train)`

CatBoost Regression

- Boosting model optimized for categorical data.
- Code:
- `from catboost import CatBoostRegressor`
- `model = CatBoostRegressor().fit(X_train, y_train)`

Neural Networks for Regression

- Deep learning-based regression.
- Code:
- `from tensorflow.keras.models import Sequential`
- `model = Sequential([...])`

Performance Metrics in Regression

- R^2 , MSE, RMSE, MAE.
- Code:
- `from sklearn.metrics import
mean_squared_error`
- `mean_squared_error(y_test, y_pred)`

Feature Selection in Regression

- Selecting the most important features.
- Code:
- `from sklearn.feature_selection import SelectKBest`
- `SelectKBest().fit(X, y)`

Hyperparameter Tuning in Regression

- Improving model performance.
- Code:
- `from sklearn.model_selection import GridSearchCV`
- `GridSearchCV(model, params).fit(X_train, y_train)`

Overfitting and Underfitting in Regression

- Balancing model complexity.

Handling Multicollinearity in Regression

- Checking correlation between features.

Case Study: Real-World Regression Example

- Applying regression in a real-world scenario.

Comparison of Different Regression Models

- Pros & cons of different regressors.

Deployment of Regression Models

- Deploying models using Flask or Streamlit.

Challenges in Regression & Best Practices

- Common pitfalls & how to avoid them.

Final Thoughts & Recommendations

- Choosing the right model for the right task.

Thank You

- By Md Anique Zzama