Introduction to Python

Fondren Library
Research Data Services

Basics of Machine Learning

Jobs in AI/Machine Learning focusing on Data Science

1. **Data Scientist:** This role involves analyzing complex data sets to derive insights, build predictive models, and create machine learning algorithms. It typically requires knowledge of statistical analysis, programming, and domain expertise.
2. **Machine Learning Engineer:** This position focuses on designing, building, and deploying machine learning models. It requires strong programming skills, understanding of machine learning algorithms, and experience with tools like TensorFlow, Scikit-learn, and PyTorch.
3. **Data Analyst:** Data analysts focus on interpreting data and providing actionable insights. They often use SQL, Excel, and visualization tools like Tableau or Power BI.
4. **Research Scientist:** This role involves conducting original research in machine learning and AI, often in an academic or corporate research setting. It requires deep knowledge of algorithms, theory, and experimentation.
5. **Business Intelligence Analyst:** This position focuses on analyzing business data to support decision-making processes. It involves creating reports, dashboards, and visualizations to communicate findings.

Key Concepts in Machine Learning

Important concepts in machine learning include overfitting, underfitting, bias, variance, and model evaluation metrics. Understanding these concepts is crucial for building robust and generalizable models.

Overfitting: Overfitting occurs when a machine learning model learns not only the underlying pattern in the training data but also the noise and outliers. This results in a model that performs exceptionally well on the training data but poorly on new, unseen data because it fails to generalize. # Issue that Focuses on the Dataset

Underfitting: Underfitting happens when a machine learning model is too simple to capture the underlying pattern in the data. This leads to poor performance on both the training data and new data, as the model fails to represent the complexity of the data. # Issue that Focuses on the Dataset

Bias: Bias refers to the error introduced by approximating a real-world problem, which may be complex, by a simplified model. High bias can cause an algorithm to miss relevant relations between features and target outputs (underfitting).

Variance: Variance refers to the error introduced by the model's sensitivity to small fluctuations in the training set. High variance can cause an algorithm to model the random noise in the training data rather than the intended outputs (overfitting).

Key Concepts in Machine Learning

Important concepts in machine learning include overfitting, underfitting, bias, variance, and model evaluation metrics. Understanding these concepts is crucial for building robust and generalizable models.

Overfitting: Overfitting occurs when a machine learning model learns not only the underlying pattern in the training data but also the noise and outliers. This results in a model that performs exceptionally well on the training data but poorly on new, unseen data because it fails to generalize.

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split

# Dummy Data
X = [[i] for i in range(10)]
y = [5, 3, 6, 4, 7, 8, 5, 3, 6, 7]

# Train-test split data
X_train, X_test, y_train, t_test = train_test_split(X, y, test_size=0.2)

# Model training
model = RandomForestRegressor(n_estimators=100)
```

Key Concepts in Machine Learning

Model Evaluation: Model evaluation involves assessing the performance of a machine learning model using metrics like accuracy, precision, recall, F1 score, and ROC-AUC. Evaluation is critical to understand how well the model is likely to perform on unseen data.

Generalization: Generalization is the model's ability to perform well on new, unseen data after being trained on a training set. A model that generalizes well captures the underlying patterns of the training data without overfitting to noise or outliers.’

Feature Engineering: Feature engineering is the process of using domain knowledge to create new features that make machine learning algorithms work better. It involves transforming raw data into features that better represent the underlying problem to the predictive models.

Cross-Validation: Cross-validation is a technique for assessing how the results of a statistical analysis will generalize to an independent dataset. It is mainly used in settings where the goal is prediction and involves partitioning a dataset into a training set and a validation set multiple times to get an average performance metric.

Key Concepts in Machine Learning

Hyperparameter Tuning: Hyperparameter tuning involves searching for the best parameters that control the learning process of a machine learning algorithm. It is usually done using techniques like GridSearchCV or RandomizedSearchCV to find the set of hyperparameters that result in the best performance on a validation set.

Regularization: Regularization involves adding a penalty to the loss function to prevent the model from overfitting by keeping the model coefficients small. Techniques like Lasso (L1) and Ridge (L2) regularization are commonly used to constrain the complexity of the model.

Normalization and Standardization: Normalization and standardization are techniques used to scale the features of a dataset. Normalization scales the data to a [0, 1] range, while standardization scales the data to have a mean of 0 and a standard deviation of 1. These techniques are crucial for algorithms sensitive to the scale of data, such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNNs).

Key Terms:

- **Feature:** A measurable attribute or variable in the data, like age or income.
- **Label:** The target variable the model predicts, like house prices in a prediction model.
- **Training Set:** Data used to teach the model, containing features and labels.
- **Test Set:** Data used to evaluate the model's performance, containing features and labels.
- **Validation Set:** Data used to fine-tune the model and adjust hyperparameters.
- **Hyperparameters:** Settings set before training that affect how the model learns, like learning rate or number of trees in a forest.
- **Metrics:** Measures to evaluate model performance, such as accuracy, precision, recall, and F1 score.

Scikit-Learn

Scikit-Learn is a powerful Python library for machine learning. It provides functions to support the entire machine learning process from data to prediction for many of the commonly used models including:

- Logistic Regression
- Linear Regression
- K-Means
- Principal Component Analysis
- Decision Trees
- Random Forest (Multiple Decision Trees)

```python
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Load and split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Initialize and train the model
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Make predictions and evaluate the model
predictions = model.predict(X_test)
accuracy = accuracy_score(y_test, predictions)
print(f"Model Accuracy: {accuracy}")
```

Scikit-Learn Functions

- **Fit():** The fit() function takes in the training data and corresponding labels to train the machine learning model.

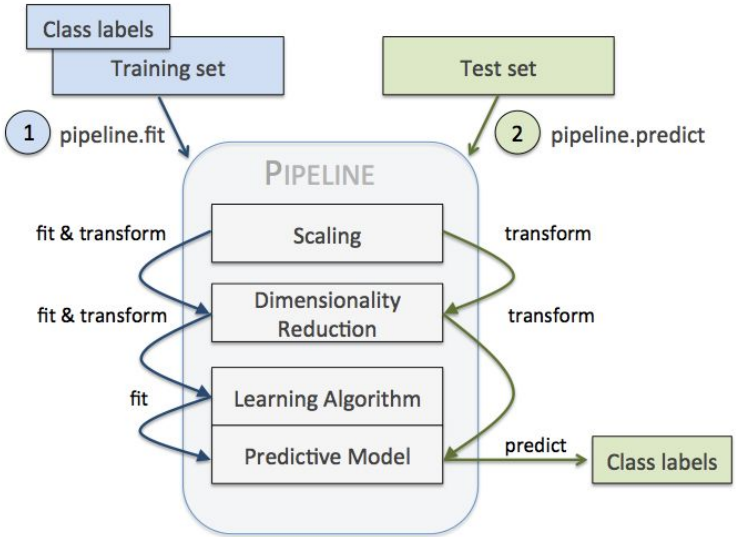
```
model = LogisticRegression()
model.fit(X_train, y_train)  # X_train and y_train are the training data and labels
```

- **Predict():**The predict() function uses the trained model to make predictions on new data.

```
predictions = model.predict(X_test)  # X_test is the test data
```

- **Transform():**The transform() function is used to apply the learned transformations (e.g., scaling, normalization) to the data.

```
X_train_scaled = scaler.transform(X_train)  # Apply scaling to the training data
X_test_scaled = scaler.transform(X_test)  # Apply the same scaling to the test data
```

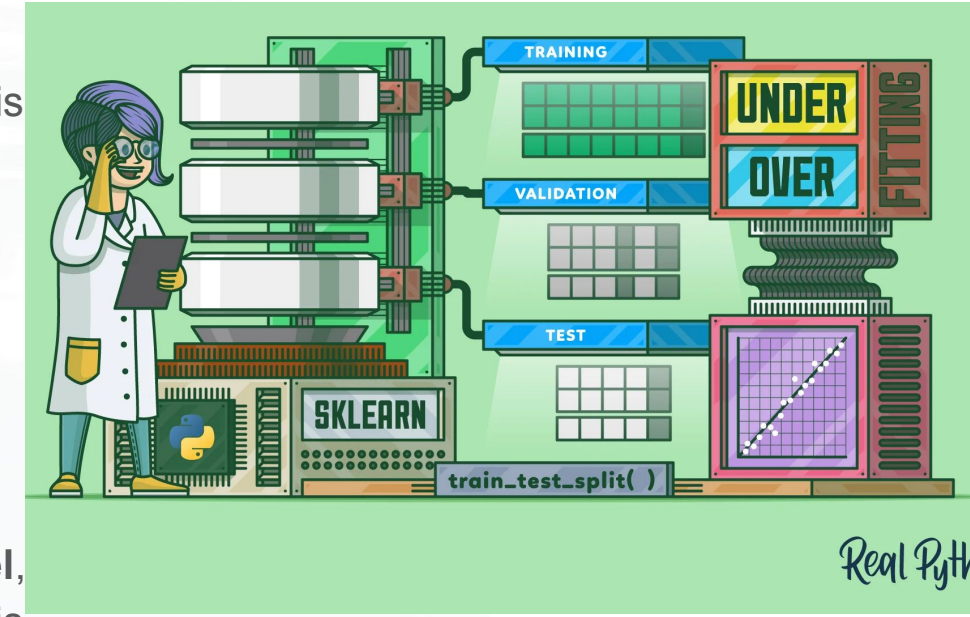


Splitting the data

Splitting the dataset into training and testing sets is crucial for evaluating model performance.

- **Training set:** Used to train the model.
- **Testing set:** Used to assess the model's accuracy and generalizability.
- **Validation set:** Used during development to tune hyperparameters.

Data splits directly impact the fit of your model, whether it is overfit or underfit. Standard practice is to split the data 70-20-10, Training data, testing data, validation data.



Hyperparameters

Learning Rate:
Controls model update magnitude during training
Number of Trees (n_estimators) in Random Forest:
Defines the number of trees in the forest
Number of Clusters (n_clusters) in K-Means:
Determines the number of clusters to form
Maximum Depth (max_depth) in Decision Tree:
Sets the maximum depth of the tree
Min Samples Split (min_samples_split) in Decision Tree:
Minimum samples required to split an internal node
GridSearchCV is a function in Scikit Learn that does an exhaustive search over parameter values for better model performance.

Q&As