FEDOT Cheat Sheet

Quick Start

```python
# Installation
pip install fedot

# Basic usage example
from fedot.api.main import Fedot
model = Fedot(problem='classification')
model.fit(features=X_train, target=y_train)
prediction = model.predict(features=X_test)
```

Core Concepts

- **Pipelines**: Computational graphs that combine data operations and models
- **Nodes**: Individual operations within a pipeline (preprocessing or models)
- **Edges**: Connections between nodes that define data flow
- **Composer**: Algorithm that creates pipeline structures
- **Tuner**: Optimizes hyperparameters of models in a pipeline

Common Methods

Automatic Pipeline Construction

Method	Description	Example
`Fedot()`	Initialize the framework	`model = Fedot(problem='regression')`
`fit()`	Train the model	`model.fit(features=X, target=y)`
`predict()`	Make predictions	`predictions = model.predict(features=X_test)`
`save()`	Save the model	`model.save('model.pkl')`
`load()`	Load a saved model	`model = Fedot.load('model.pkl')`

Manual Pipeline Construction

Method	Description	Example
`Pipeline()`	Create empty pipeline	`pipeline = Pipeline()`
`Node()`	Create operation node	`node = Node(operation_type='xgboost')`

Method	Description	Example
add_node()	Add node to pipeline	pipeline.add_node(node)
add_edge()	Connect nodes	pipeline.add_edge(node_from, node_to)
fit()	Train pipeline	pipeline.fit(input_data=train_data)
predict()	Get predictions	predictions = pipeline.predict(input_data=test_data)

Configuration Options

```
# Configuring FEDOT
model = Fedot(
    problem='classification',
    preset='best_quality',  # Options: 'fast', 'stable', 'best_quality'
    timeout=5,  # Minutes for optimization
    n_jobs=-1,  # CPU cores to use (-1 = all)
    cv_folds=5,  # Cross-validation folds
    random_state=42,  # Random seed
    metric=["accuracy"]  # Set metrics to optimize
)
```

Available Models and Operations

Primary Nodes (Data Processing)

scaling • normalization • pca • kernel_pca • fast_ica • polynomial_features • lagged • smoothing

Secondary Nodes (Models)

linear • ridge • lasso • rf • xgboost • lgbm • catboost • knn • dt • mlp

Common Patterns

Time Series Forecasting

```
# Time series forecasting
model = Fedot(problem='ts_forecasting', forecast_length=10)
train_input, test_input = train_test_data_setup(dataframe)
model.fit(train_input)
forecast = model.predict(test_input)
```

Troubleshooting Tips

- **Memory Error**: Reduce `pop_size` and `num_of_generations` parameters
- **Poor time series performance**: Add lagged features with `lagged` operation nodes
- **Slow execution**: Set `preset='fast'` or reduce optimization parameters
- **Overfitting**: Use cross-validation or simpler pipelines
- **Model not improving**: Try different metrics or increase optimization time

Resources

- Documentation: fedot.readthedocs.io
- GitHub: github.com/aimclib/FEDOT