Karlsruher Institut für Technologie

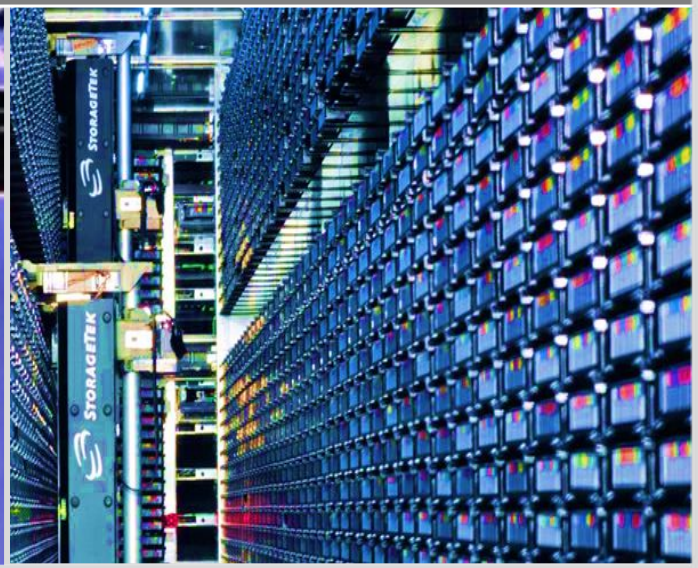
# AUTOML for walltime predictions

## Docotoral Reaserch Meeting 20.11.2019

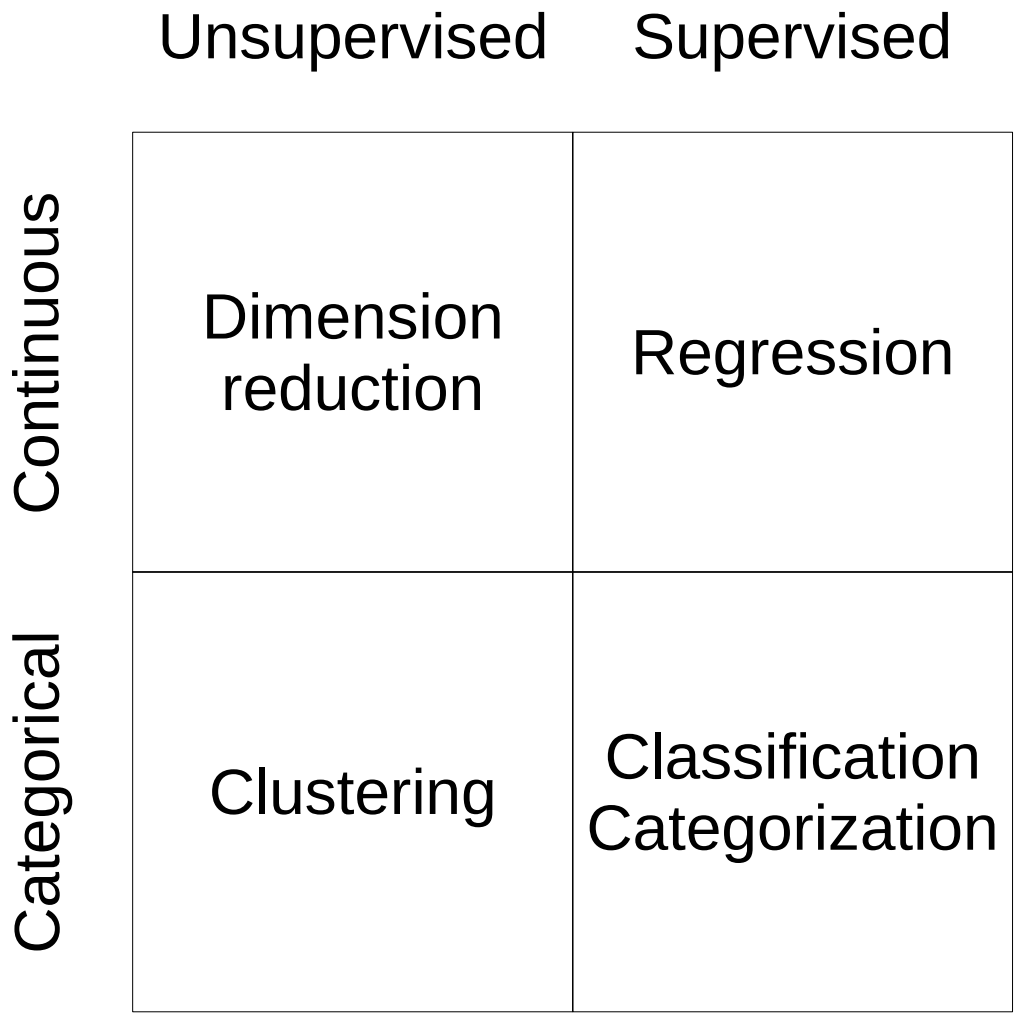

## Mehmet Soysal

STEINBUCH CENTRE FOR COMPUTING - SCC



# Machine Learning



# Why Automatic Machine Learning (AUTOML)

- Which algorithm ?
- Which Hyperparameter ?
- Which features are important ?
  - Optimizaion strategy
  - ...
- AUTOML:
  - TPOT: “Genetic” search algorithm
  - auto-sklearn : Gridsearch

# Example: without AUTOML

- Without AUTOML

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

df = pd.read_csv(myfile.csv)

<do some preprocessing>

Y = df.pop('target_values').values
X_train, X_test, y_train, y_test = train_test_split(df, Y, random_state = 42)
model = LinearRegression()
model.fit(X_train, y_train)
y_predict = model.predict(X_test)

<do some evaluation>
#compare y_predict  and y_test
```

# Example: without AUTOML

- Without AUTOML

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

df = pd.read_csv(myfile.csv)

<do some preprocessing>

Y = df.pop('target_values').values
X_train, X_test, y_train, y_test = train_test_split(df, Y, random_state = 42)
model = LinearRegression()
model.fit(X_train, y_train)
y_predict = model.predict(X_test)

<do some evaluation>
#compare y_predict  and y_test
```

# Example: with auto-sklearn

- Without AUTOML

```
import pandas as pd
from sklearn.model_selection import train_test_split
from autosklearn.regression import AutoSklearnRegressor

df = pd.read_csv(myfile.csv)

<do some preprocessing>

Y = df.pop('target_values').values
X_train, X_test, y_train, y_test = train_test_split(df, Y, random_state = 42)
model = AutoSklearnRegressor()
model.fit(X_train, y_train)
y_predict = model.predict(X_test)

<do some evaluation>
#compare y_predict  and y_test
```

# Example: with TPOT

- Without AUTOML

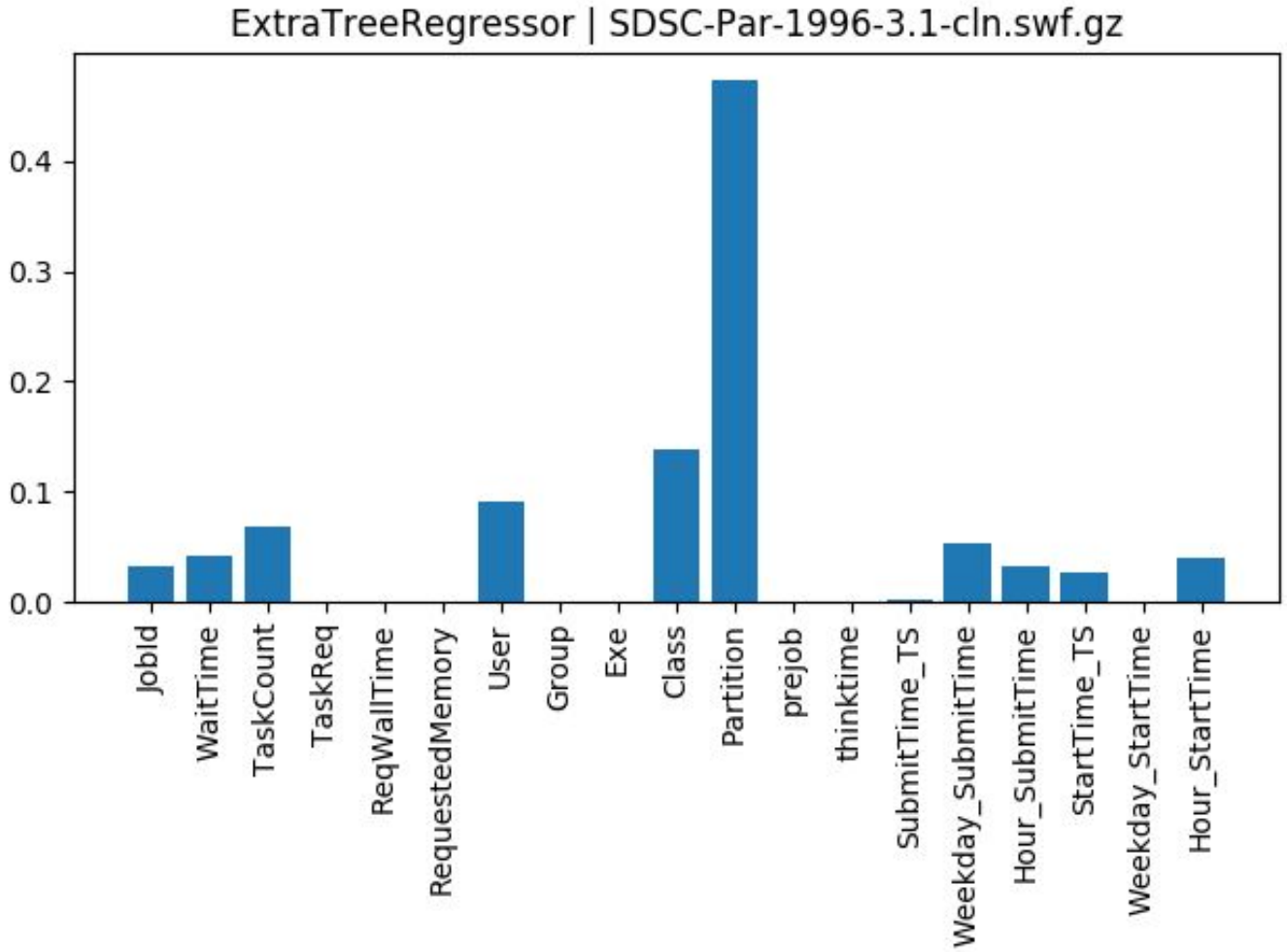
```
import pandas as pd
from sklearn.model_selection import train_test_split
from tpot import TPOTRegressor

df = pd.read_csv(myfile.csv)

<do some preprocessing>

Y = df.pop('target_values').values
X_train, X_test, y_train, y_test = train_test_split(df, Y, random_state = 42)
model =  TPOTRegressor()
model.fit(X_train, y_train)
y_predict = model.predict(X_test)

<do some evaluation>
#compare y_predict  and y_test
```

# AUTOML for HPC job walltime prediction

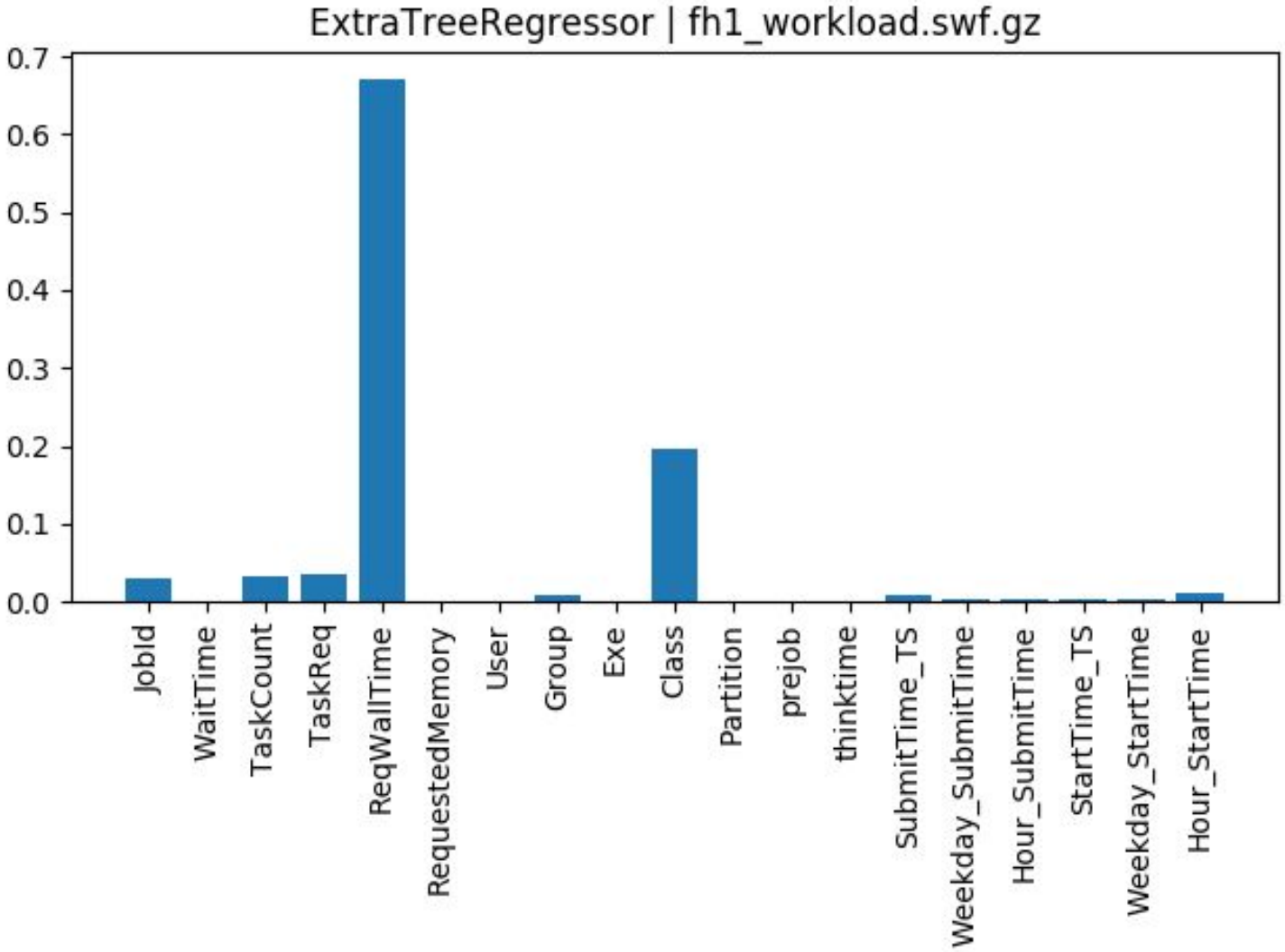
- Accurate job wall times are important
- User estimates are far from optimal
- Many approaches and methods
- Different systems policies / user mindsets  / configurations
- Historical data available[1]
  - 16 features from JobID, TaskCount …
  - Not all features are used!!
  - Also ForHLR I+II available
- Goal: predict the run time of the jobs

1) https://www.cse.huji.ac.il/labs/parallel/workload/logs.html

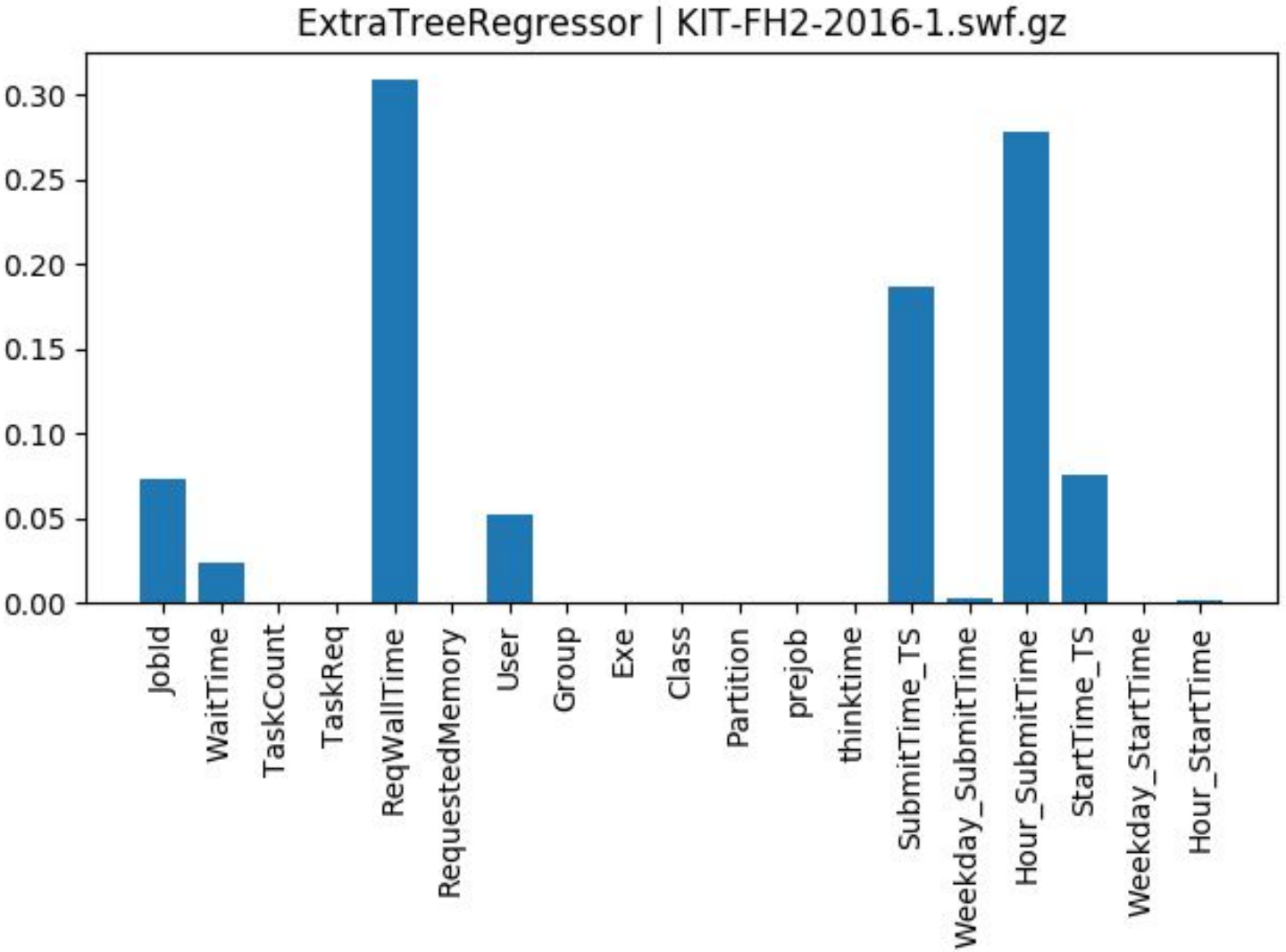
# Feature importance



# Feature importance



# Feature importance



# Walltime prediction

- Survey for up to date methods (paper in work)
  - Alea / simple averages on ratio run/request (last 5 jobs)
  - scikit (different regressors)
  - AUTOML with auto-sklearn / TPOT
  - Tensorflow + keras + hyperas
- Repo available here[1]
- Feel free to create your version
  - e.g. pytorch with a hyperparameter optimizer ;)

1) https://git.scc.kit.edu/az2556/walltime-prediction-tools

# Conclusion

- Sci-kit good starting point for ML
- Use visualization plots
- Save time with AUTOML
  - Store model for later usage
- Preprocessing data is time consuming