# Problem Statement:

Today's machine learning (ML) is not limited to research and development applications, but has already entered the enterprise field. However, the traditional ML process still relies on the experts, and not all companies have the resources to invest in an experienced data science team. Automated Machine Learning (AutoML) is an answer to this dilemma. AutoML is automating the end-to-end process of applying machine learning to real-world problems. However, two main problems persist even in AutoML:

1. The general data analysis pipeline includes 5 processes, they are preprocessing, feature engineering, feature selection, model selection and hyperparameter finetune. How to combine all these processes?
2. Facing the huge search space, how to shorten the search time it takes to find a better solution?

On the other hand, Monte Carlo Tree Search (MCTS), as a method of reinforcement learning, has great advantages in dealing with optimization problems. Its performance in Go games leads us to believe that MCTS can quickly find a better solution in a huge search space. In addition, his tree structure allows us to easily combine multiple processes in one pipeline and has better interpretability and parallelism.

Based on the facts above, MCTS can possibly be used to implement AutoML to achieve a better AutoML model.

Four hypotheses will be verified in this thesis：

1. The new AutoML model implemented by MCTS can find better solutions comparing with existing AutoML method, for example TPOT, in a given time.
2. The new AutoML model implemented by MCTS can find better solutions comparing with existing AutoML method, for example TPOT. (without time limitation).
3. The new AutoML model implemented by MCTS is interpretable.
4. The new AutoML model implemented by MCTS is parallelizable.

# IDEA:

The basic ideas include following three points:

1. Each process in the data analysis pipeline has many different possible operations, for example, when we do the feature engineering, we can use both differentiate and integrate. In our work we use the node (state) of the MCT to indicate the operations that has been selected, use the edge (action) of the MCT to indicate the operation to be added to the state, and use the dictionary to constrain the action that can be selected in the current state. When the node of the MCT reach some predefined conditions, we evaluate the node and record the result.
2. There are constraints and dependencies between different processes on the data analysis pipeline. For example, some hyperparameters only belong to a specific model. It is confusing to record all the memory (evaluation information) on the MCT. We try to establish a subtree for each process, and Combine all the subtrees into the acyclic graph.
3. In the general MCTS, the input of the neural network is the state of a certain node, therefore the neural network has to simultaneously take care of the two functions: memory storage and memory usage. We try to use Graphic convolution network (GCN) to extract information from the stored memory and input it into the network to simplify the network.

# METHODOLOGY

1. Collect and read papers which related to AutoML and Reinforcement.
2. Summarize the search space. for example, what kinds of classification model do we have.
3. Collect a large number of datasets ensuring that the collected datasets can cover as many areas in real-world as possible.
4. Collect existing AutoML packages that can be used directly with python and make use of them as comparisons in the experiment.
5. Implement AutoML with MCTS.
6. Design experiments to evaluate the generated AutoML model.

# CONTRIBUTION:

In this work, we attempt to improve the state of the art by using the graph to record the evaluation information and employing the Graph Convolutional Network (GCN) to extract the information. In addition, we try to simplify the network structure by separating the functions of the neural network, thereby shortening the time required for network training and reducing the amount of data required to train a qualified network. Through the new AutoML model, we can achieve end-to-end process automation that can be applied to real-world problems.

# WORKPLAN:

1. Research: read papers in domain reinforcement Learning (RL), AutoML, Data mining. (7W)
2. Collect: download dataset and AutoML packages that will be used in the experiments. (1W)
3. Program: implement AutoML model with MCTS. (7W)
4. Training: train the new AutoML model. (1W)
5. Experiment: design and run experiments to test the trained model. (4W)
6. Paper: write scientific paper. (4W)

# CORE TASK:

1. Compare existing AutoML methods.
2. Look for and download large amount of dataset.
3. Implement AutoML with MCTS.
4. Design and run experiments to test the new AutoML model.

# BASIC KNOWLEDGE:

1. AutoML
2. Data Mining
3. Reinforcement learning