1. Собрание *16.10.16*

# Задачи

1. Общая структура проекта
   1. Разделение ответственностей – UI, БД, Решатели, пред- и постобработка.
   2. Архитектура решения – типы проектов (exe, dll); средства разработки.
   3. Репозиторий.
2. Параметры решаемых задач
   1. Типы задач
   2. Количество выходных атрибутов
3. Определение общего интерфейса решателя.
   1. Типы входных и выходных данных.
   2. Хранение параметров решателя в БД.
4. Алгоритмы обучения
   1. Связь АО и АР, способ взаимодействия
   2. Взаимосвязь АО и датасетов – кроссвалидация, простое разделение выборки (обучающая/тестовая)
5. Предобработка
   1. Возможные типы входных данных. Поддерживаемые форматы файлов
   2. Возможные типы конвертирования и очищения данных.
   3. Взаимодействие с постобработкой – обратное конвертирование
   4. Визуализация данных
   5. Взаимодействие с БД, хранение конвертированных датасетов, разделенных выборок.
6. Постобработка
   1. Возможные типы постобработки и их визуализация
   2. Взаимодействие с БД

# Итоги

1. Общая структура проекта
   1. Ответственности: UI – **Смирнов М**, ViewModel’s – **Мошков Ю**. Классы для пред- и постобработки данных – **Малойкина Е**. Управление обучением и предоставление результатов – **Сиротин Н**. Все остальные ответственности согласно темам дипломных работ.
   2. IDE – **Visual Studio 2013** (2015). Проекты: (UI + ViewModel) – exe, Классы БД – библиотека классов, Классы обработки данных – библиотека классов, Нейронные сети, Деревья решений и др. решатели – **единая** dll, Алгоритмы Обучения – **единая** dll.
   3. Репозиторий – github (<https://github.com/UNN-Data-Mining-Group/Data-Mining-Tool-System>). Процесс добавления функциональности – через **pull request с локального fork’а**.
2. Параметры решаемых задач.
   1. Система позволяет решать задачи, для которых предоставлены выборки с **входными и выходными параметрами** – обучение с учителем и обучение с подкреплением.
   2. В система должна быть возможность решать задачи с несколькими **выходными атрибутами**
3. Определение общего интерфейса решателя
   1. Любой решатель имеет метод **solve**, принимающий на вход массив **float** и возвращающий массив **float**.
   2. Интерфейс решателя **не имеет** методов записи/чтения решателя из/в БД. Для записи решателя в БД объявляется интерфейс, имеющий соответствующий метод. Каждый конкретный решатель, который требуется записать в БД, необходимо наследовать как от интерфейса решателя, так и от данного интерфейса.
4. Алгоритмы обучения
   1. Решатели используются **внутри** АО. Каждому конкретному АО обучаемый АР передается как **параметр**.
   2. Управление процессом обучения осуществляется «выше» АО.
5. Предобработка
   1. Входные данные – **int, float, double, bool, enum** (строковый). Формат входных файлов – **csv**, с возможностью **настройки** (например, задания разделителя)
   2. Оставлено на будущее
   3. Методы управления обучением и методы решения задачи вызывают обратные конвертеры, объявленные в библиотеке предобработки.
   4. Не обсуждалась
   5. Все промежуточные и разделенные датасеты хранятся в БД.
6. Постобработка – решено включить в предобработку как обратные методы.

# План работ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Смирнов М** | **Винницкий В** | **Мошков Ю** | **Сабанов Д** | **Сиротин Н** | **Малойкина Л** |
| Подготовить репозиторий | Виртуальная машина и образ системы | Возможности вью-моделей приложения, визуализация результатов | Интерфейс библиотеки ДР | Интерфейс системы управления обучением | Обзор дипломных работ, предыдущей версии системы |
| Создать шаблон решения и UI приложения | Процесс загрузки решателя из БД | Плюсы и минусы реляционной БД | Обзор алгоритмов обучения ДР и леса | Интерфейс системы управления решением | Обзор предобработки данных в data mining |
| Интерфейс библиотеки НС | Плюсы и минусы файловой БД | Интерфейс классов взаимодействия с БД, организация БД. | Процесс загрузки решателя из БД | Визуализация результатов обучения / решения |  |
| Унификация АО для разных АР | Интерфейс библиотеки АО | Процесс загрузки решателя из БД |  | Обеспечение возможности обучить решатель, используя конкретные файлы обучающей и тестовой выборок |  |
|  | Интеграция С++ и C# |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  | Высокий приоритет |
|  |  |  |  |  | Средний приоритет |
|  |  |  |  |  | Низкий приоритет |
|  |  |  |  |  | После дождичка в четверг |