

Экземпляр звезды – объект, значение которого берется из множества классов «Типы звёзд» (например, «О», «В», «А»).

Описание свойств экземпляра – понятие, которое обозначает множество характеристик, описывающих данную звезду (например, «Температура», «Цвет», «Показатель цвета»).

Значение для экземпляра – понятие, которое описывает значения характеристик для конкретной звезды (например, «25 000 К», «Голубой», «-0,3»).

1.3.2 Ограничения целостности ситуаций

– Для каждого экземпляра звезды должно быть указано непустое множество свойств, описывающих этот экземпляр.

– Все значения свойств экземпляра звезды должны принадлежать множеству допустимых значений для соответствующих свойств, определённого в модели знаний.

1.4 Связь между знаниями и действительностью

– Все значения, заданные в конкретной ситуации, должны быть допустимыми в рамках "возможных значений" соответствующих свойств.

– Если "экземпляр звезды" отнесён к конкретному типу, то:

- его "описание свойств экземпляра" должно в точности совпадать с "описанием свойств типа";

- каждое "значение для экземпляра" должно входить в "значение для типа" для соответствующего свойства.

термин "возможные значения" обозначает функцию, которая сопоставляет каждому свойству звезды возможные значения этого свойства.

(2.4) сорт описание свойств типа: (типы звёзд \rightarrow ($\{\}$ свойства))

термин "описание свойств типа" обозначает функцию, которая сопоставляет каждому типу звезды подмножество множества свойств, описывающих этот тип звезды.

(2.5) сорт значение для типа: ($\{(v: (\times \text{ типы звёзд, свойства})) \mid \pi(2, v) \in \text{описание свойств типа } (\pi(1, v))\} \rightarrow \text{множество значений}$)

термин "значение для типа" обозначает функцию, которая сопоставляет каждому типу звезды свойство и его значение.

Ограничения целостности знаний

(3.1) (v : типы звёзд) (описание свойств типа (v) $\neq \emptyset$)

для каждого типа звезды множество свойств, принадлежащих описанию свойств этого типа, является непустым множеством. То есть для любого типа звезды должны быть определены хотя бы одно свойство.

(3.2) ($v1$: типы звёзд) ($v2$: описание свойств типа ($v1$)) (значение для типа ($v1$, $v2$) $\neq \emptyset$) & (значение для типа ($v1$, $v2$) \subseteq возможные значения ($v2$))

для любого типа звезды, для любого свойства, характерного для данного типа, множество значений этого свойства должно быть непустым подмножеством множества возможных значений этого свойства. То есть для каждого типа звезды и его свойства необходимо указать хотя бы одно допустимое значение, и это значение должно быть из предустановленного множества возможных значений для данного свойства.

2.1.3 Определение системы понятий действительности

Описание сортов терминов для описания ситуаций

(4.1) сорт экземпляр звезды: типы звёзд

Термин «экземпляр звезды» обозначает объект, относящийся к одному из типов звёзд.

(4.2) (v: экземпляр звезды) сорт описание свойств экземпляра(v): {} свойства

Термин «описание свойств экземпляра» обозначает множество свойств, заданных для конкретной звезды.

(4.3) (v: экземпляр звезды) (e: описание свойств экземпляра(v)) сорт значение для экземпляра(e): возможные значения(e)

Ограничения целостности ситуаций

(5.1) описание свойств экземпляра $\neq \emptyset$

(5.2) $\forall v \in \text{экземпляр звезды} \forall e \in \text{описание свойств экземпляра}(v) \exists z \text{ значение для экземпляра}(e) = z$

2.1.4 Связь между знаниями и действительностью

(6.1) $(\forall e \in \text{описание свойств экземпляра}) \text{ значение для экземпляра}(e) \in \text{возможные значения}(e)$

(6.2) $(\forall v \in \text{экземпляр звезды}) \text{ описание свойств экземпляра}(v) = \text{описание свойств типа } (v) \wedge (\forall e \in \text{описание свойств экземпляра}(v)) \text{ значение для экземпляра}(e) \in \text{значение для типа } (v, e)$

2.2 Построение модели знаний

(7.1) свойства $\equiv \{\text{температура, цвет, показатель цвета, абсолютная звёздная величина}\}$

рассматриваются четыре свойства для звёзд: температура, цвет, показатель цвета, абсолютная звёздная величина.

(7.2) типы звёзд $\equiv \{O, B, A, F, G, K, M\}$

рассматриваются типы звёзд: O, B, A, F, G, K, M.

(7.3) возможные значения $\equiv (\lambda(v: \{\text{температура, цвет, показатель цвета, абсолютная звёздная величина}\}) / (v \in \{\text{температура}\} \Rightarrow R[2500, 30000]) (v \in \{\text{цвет}\} \Rightarrow$

4.4.2 Подготовка обучающих данных

Обучение модели осуществляется на основе базы знаний, без привлечения сторонних астрономических датасетов. Это возможно благодаря тому, что в базе чётко указаны диапазоны и допустимые значения для каждого свойства каждого типа звезды.

Алгоритм генерации обучающей выборки:

1. Загружаются:
 - 1.1. Все свойства (числовые и перечислимые),
 - 1.2. Диапазоны допустимых значений из `knowledge_base.json`.
2. Для каждого типа звезды:
 - 2.1. Создаётся 2000 случайных примеров (параметр `samples_per_class`);
 - 2.2. Числовые свойства случайным образом выбираются в пределах заданных диапазонов;
 - 2.3. Перечислимые свойства выбираются случайным образом из допустимых значений;
 - 2.4. Для каждого признака с вероятностью 25% значение может быть помечено как отсутствующее:
 - 2.4.1. в этом случае вектора дополняются масками: 1.0, если признак задан, и 0.0, если отсутствует;
 - 2.4.2. отсутствующим признакам присваивается заглушка (например, 0.5 для нормализованного значения).
3. Каждое значение нормализуется:
 - 3.1. для чисел: перевод в диапазон $[0, 1]$;
 - 3.2. для перечислимых значений: индекс в массиве допустимых значений, делённый на $(N - 1)$.
4. Каждое значение признака представляется вектором из двух элементов: значение и маска.
5. Класс звезды сохраняется как метка.

Пример нормализации:

```
def normalize(val, min_val, max_val):  
    return (val - min_val) / (max_val - min_val)
```

4.4.3 Обучение и сохранение модели

Обучение происходит с использованием:

1. One-hot encoding меток (CategoricalCrossentropy);
2. Adam — адаптивный оптимизатор;
3. Эпохи: 100;
4. Размер батча: 32.

После завершения обучения:

1. Модель сохраняется в файл ai/star_model.keras;
2. Кодировщик меток LabelEncoder сериализуется в ai/label_encoder.pkl.

Эти файлы затем используются в рабочей системе при запуске нейросетевого классификатора (neural_classifier.py). Классификатор формирует входной вектор, соответствующий обучающей выборке, и возвращает наиболее вероятный тип звезды вместе с уровнем уверенности.

4.4.4 Использование в системе

При выборе опции "Использовать нейросеть" в интерфейсе DataInputWidget:

1. Значения нормализуются и кодируются.
2. Подготовленный вектор передаётся в модель.
3. Модель возвращает:
 - 3.1. Метку (тип звезды),
 - 3.2. Уверенность в предсказании (вероятность).

Результат выводится в виде:

Тип звезды (нейросеть): G (уверенность: 94.56%)

4.5 Особенности реализации

Реализация системы учитывает ряд ключевых инженерных решений, обеспечивающих её надёжность, расширяемость и удобство использования.

Все компоненты системы разделены на логические модули:

1. Интерфейсы (ввод данных, редактор знаний, просмотр),
2. Логика классификации (логическая и нейросетевая),
3. Работа с базой знаний,
4. Вспомогательные утилиты (валидация, стили, преобразование данных).

Такой подход упрощает сопровождение, тестирование и последующее расширение функциональности.

Программа ориентирована на две основные роли:

1. Эксперт — создаёт и редактирует базу знаний (типы звёзд, свойства, диапазоны);
2. Специалист — вводит характеристики звезды и получает классификацию.

Интерфейс каждого пользователя адаптирован под его задачи.

Система реализует:

1. Логический классификатор, исключающий неподходящие типы на основе жёстких правил;
2. Нейросетевой классификатор, способный учитывать погрешности и делать вероятностные предсказания.

Это делает систему более гибкой в использовании и надёжной при неидеальных данных.

Нейросеть обучается на искусственно сгенерированных данных, соответствующих описаниям из базы знаний. Это исключает необходимость поиска и подготовки внешних датасетов, делает обучение воспроизводимым и управляемым.

Интерфейс снабжён:

1. Валидацией ввода,
2. Подсказками и пояснениями,
3. Функцией генерации случайных данных для тестирования,
4. Возможностью быстро просматривать и проверять знания системы.