**Слайд 1:**

Уважаемые челены государственной комиссии. Я Нгуен Тхань Тхиен, представляю вашему вниманию дипломную работу на тему: Применение алгоритма случайного леса для машинного обучения при исключении добросовестных соискателей кредита.

**Слайд 2:**

Целью работы является Исследование алгоритма машинного обучения случайного леса с целью классификация риска соискателей в процессе одобрения кредита. Для достижения цели необходимо решить следующие задачи:

* Анализ алгоритма случайного леса.
* Выбор и построение дерева решений.
* Критерии разделения.
* Критерий остановки.
* Выбор данных.
* Подготовка данных.
* Построение модели случайного леса.
* Оценка эффективности обученной модели случайного леса.
* Создавать приложение, для прогнозирования риска соискателей в процессе одобрения кредита.

**Слайд 3:**

На самом деле в каждом лесу должно быть много деревьев. Алгоритм случайного леса также включает множество деревьев решений, отсюда и название «лес». А слова «случайный», это значит, что деревья в лесу разные (случайные), мы будем рассмотреть эту разницу более подробно позже.

Случайный лес – это алгоритм машинного обучения, предложенный  Лео Брейманом и Адель Катлер [по Википедия].

**Слайд 4:**

На рисунке, показано как построить случайный лес из исходных данных, то есть выбрать случайную выборку данных, а затем для каждой выборки мы построим дерево решений (это слово "случайный" я сказал). И результат случайного леса является результатом большинства этих деревьев решений (голосование).

**Слайд 5:**

Существует три метода построения дерева решений, свойства случайного леса зависит от свойств деревьев решений.

**Слайд 6:**

Сравнивая скорость между **id3** и **cart**, id3 медленнее из-за использования логарифмов для расчета. Но **cart** производит только бинарные деревья, поэтому метод **cart** не выбран. C4.5 имеет то преимущество, что позволяет избежать переобучения, но переобучение часто происходит, в небольших данных. Потому я выбрал ID3.

**Слайд 7:**

Это блок-схема метода ID3 (с использованием прироста информации).

**Слайд 8:**

Entropy указанно неупорядоченность признаков с метками. Если узел является чистым, тогда entropy имеет минимальное значение (0).

**Слайд 9:**

Следующей концепцией, позволяющей определить, хорош ли разрез, является приростом информации, чем больше Gain(S, A), тем более однородны элементы.

**Слайд 10:**

Чтобы дерево не росло до бесконечности, должны быть заданы критерии остановки. Есть четыре условия остановки:

* Ограничение максимальной глубины дерева.
* Ограничение минимального числа объектов в листе.
* Ограничение максимального количества листьев в дереве.
* Останов в случае, если все объекты в листе относятся к одному классу.

**Слайд 11:**

Это данные, используемые в данном работы. Включает 9709 строк, включает такую ​​информацию, как: пол, возраст, профессия, образование и метку.

**Слайд 12:**

В данных наблюдается дисбаланс, более 8000 меток 0 и чуть более 1000 меток 1.

**Слайд 13:**

Есть 2 способа для решить дисбаланс данных: Избыточная выборка и недостаточная выборка. Метод избыточная захватывает больший объем данных.

**Слайд 14:**

Вот лог-текст процесса построения случайного леса. Случайный лес из 100 деревьев решений, построенный за 16 часа. Модель имеет точность 96,5%.

**Слайд 15:**

Чтобы оценивают эффективность модели, оценивают недостаточно. Поэтому я использую матрицы путаницы. Помогите узнать точное количество правильных и неправильных ответов, в случае отрицательных и положительных меток.

**Слайд 16:**

График показывает нам:

* Точность в случае положительного: 95.2%
* Точность в случае отрицательного: 97.8%

**Слайд 17:**

Это изображение приложения предсказало рисков на случай, у соискатели высокий риск.

**Слайд 18:**

Это изображение приложения предсказало рисков на случай, у соискатели низкий риск.

**Слайд 19:**

В результате выполнения работы поставленная цель была достигнута, а также были решены все задачи. Модель случайного леса была успешно построена с высокой эффективностью классификации (97,8%).

**Слайд 20:**

Дальнейшее развитие, связанные с повышением скорости обучения, например: Написать исходный код на языках более низкого уровня, де-рекурсия.