Министерство образования Московской области
Государственное бюджетное профессиональное образовательное учреждение
Московской области «Физико-технический колледж»

ЗАЩИЩЕНО

ОЦЕНКА_____

Руководитель специальности
09.02.07 Информационные
системы и программирование

_____ _____

СИСТЕМА ОПРЕДЕЛЕНИЯ СКЛОННОСТИ КЛИЕНТА К ПРИОБРЕТЕНИЮ МАШИНОМЕСТА

Пояснительная записка к курсовому проекту
по ПМ.05 Разработка и администрирование систем искусственного интеллекта

Руководитель КП
преподаватель
ГБПОУ МО «Физтех-колледж»
_____ Г.В. Базяк

Разработал
студент группы ИСП 4-5
_____И.Р. Доля

Долгопрудный, 2024

Министерство образования Московской области
Государственное бюджетное профессиональное образовательное учреждение Московской области «Физико-технический колледж»

Специальность 09.02.07
Информационные системы и Программирование
Квалификация «Программист»

УТВЕРЖДЕНО
цикловой комиссией специальности 09.02.07
«____» _____2024 года
Руководитель специальности
_____ _____

ЗАДАНИЕ

Для курсового проектирования по МДК 05.02 «Разработка и администрирование систем искусственного интеллекта» студенту 4 курса группы ИСП 4-5

Доли Ивану Романовичу
(фамилия, имя, отчество)

Тема задания: Система определения склонности клиента к приобретению машиноместа

I Пояснительная записка
1 Анализ предметной области
1.1 Описание предметной области
1.2 Определение назначения системы и целевой группы пользователей
1.3 Определение ограничений проектного решения
2 Определение требований к системе
2.1 Функциональные требования
2.2 Требования к графическому интерфейсу
2.3 Техническое задание
3 Этапы разработки
3.1 Сбор и анализ данных
3.2 Разработка модели
3.3 Разработка интерфейса
3.4 Тестирование системы
Заключение
Список используемых источников
Приложение
Приложение А.
Приложение Б.

Дата выдачи: « » 2024г. Срок окончания: « » 2024 г

Руководитель курсового проектирования _____ Г.В. Базяк

СОДЕРЖАНИЕ

Введение

Современный мир характеризуется быстрым развитием технологий и ростом урбанизации, что приводит к увеличению числа автомобилей и, как следствие, к необходимости оптимизации использования парковочных мест. Для бизнеса, связанного с продажей или сдачей в аренду машиномест, крайне важно эффективно прогнозировать спрос на их услуги. Это требует использования современных аналитических подходов и инструментов, позволяющих оценивать склонность клиентов к приобретению машиномест.

Целью данной курсовой работы является разработка системы определения склонности клиента к приобретению машиноместа на основе анализа данных с использованием методов машинного обучения. Это позволит не только повысить эффективность работы компании, предлагающей парковочные места, но и улучшить качество обслуживания клиентов за счет индивидуального подхода.

Для выполнения задачи предполагается использование существующего набора данных, содержащего информацию о клиентах. Набор данных не подвергался предварительной обработке, что создает дополнительные вызовы, связанные с очисткой, анализом и подготовкой данных для машинного обучения. В рамках работы планируется исследовать основные этапы подготовки данных, выбрать подходящие алгоритмы машинного обучения и оценить их качество. Особое внимание будет уделено интерпретируемости полученной модели, чтобы бизнес мог использовать ее выводы для принятия обоснованных решений.

Актуальность исследования обусловлена возрастающей ролью анализа данных и технологий машинного обучения в управлении бизнес-процессами. Разработка подобной системы позволит не только

оптимизировать использование парковочных площадей, но и повысить экономическую эффективность организации.

1 Анализ предметной области

1.1 Описание предметной области

Управление парковочным пространством в городских условиях — одна из ключевых задач, возникающих на пересечении урбанистики и коммерции. С ростом плотности населения и количества автомобилей необходимость оптимального использования парковочных площадей становится все более актуальной. На практике это включает обеспечение клиентов подходящими условиями для размещения их транспортных средств, что требует учета множества факторов: местоположения, доступности, стоимости и индивидуальных предпочтений клиентов.

Приобретение машиномест часто связано с решением долгосрочных задач пользователей, таких как обеспечение удобного доступа к своему автомобилю или защита от неблагоприятных внешних факторов. В данном случае в качестве клиентов рассматриваются частные лица, владельцы автомобилей, которые принимают решение о покупке машиноместа на основе индивидуальных потребностей, таких как удобство, финансовая доступность, близость к месту жительства или работы.

Одним из ключевых аспектов предметной области является динамика спроса на машиноместа, зависящая от таких факторов, как уровень дохода клиента, расстояние от места проживания или работы до парковочной зоны, сезонные колебания и наличие альтернативных решений (например, использование общественного транспорта или аренда временного парковочного пространства). Понимание этих факторов играет решающую роль при разработке системы анализа и прогнозирования поведения клиентов.

Таким образом, предметная область находится на стыке анализа данных, маркетинга и управления недвижимостью. Ее изучение позволяет разработать эффективные инструменты, способные учитывать широкий спектр характеристик клиентов и факторов, влияющих на их склонность к покупке машиноместа.

1.2 Определение назначения системы и целевой группы пользователей

Назначение системы заключается в определении склонности клиентов фирмы, например, строительной компании или застройщика, к приобретению машиноместа. Система будет служить инструментом для анализа данных о потенциальных покупателях и предоставления прогноза, указывающего, с какой вероятностью конкретный клиент совершит покупку. Это позволит компании оптимизировать процесс работы с клиентами, сконцентрировав усилия на тех, кто наиболее вероятно заинтересован в приобретении машиноместа.

Целевая группа пользователей системы включает сотрудников компании, ответственных за продажи и управление клиентскими базами данных. Среди них:

- Менеджеры по продажам, которые используют прогнозы системы для приоритизации работы с клиентами. На основе предоставленных предположений менеджеры могут сосредотачиваться на взаимодействии с клиентами, имеющими высокую вероятность покупки.

- Руководители отделов продаж, которые могут использовать агрегированные результаты работы системы для планирования ресурсов и повышения общей эффективности процесса продаж.

- Аналитики, занимающиеся оценкой эффективности продаж и анализа клиентской базы. Система предоставляет им дополнительные данные для проверки гипотез и понимания факторов, влияющих на успешность сделок.

Система предназначена исключительно для прогнозирования вероятности покупки машиноместа конкретным клиентом. Она фокусируется на индивидуальном уровне принятия решения.

Таким образом, ключевое назначение системы заключается в упрощении принятия решений для сотрудников компании, связанных с продажами, и в повышении эффективности взаимодействия с клиентами на основе данных.

1. 3. Определение ограничений проектного решения

Разработка системы определения склонности клиента к приобретению машиноместа предполагает наличие ряда ограничений, связанных как с особенностями предметной области, так и с применением технологий машинного обучения. Основные ограничения проектного решения включают следующие аспекты:

1. Ограничения на исходные данные

- Качество данных: Так как данные для модели собираются автоматически, их качество напрямую влияет на точность прогнозов.

Ошибки при сборе данных, неполные или несбалансированные выборки могут существенно снизить производительность системы.

- Динамичность данных: Изменения в источниках данных или в их структуре могут исказить результаты работы системы. В некоторых случаях это потребует полной перенастройки модели или даже ее повторного обучения на новом наборе данных.

- Недостаток признаков: Если автоматически собираемые данные не включают критически важные признаки, влияющие на склонность клиента к покупке (например, данные о доходах или местоположении), это ограничит точность модели.

2. Ограничения модели

- Сложность интерпретации: Алгоритмы машинного обучения, особенно сложные (например, градиентный бустинг или глубокие нейронные сети), могут быть трудны для интерпретации, что затрудняет понимание причин конкретного прогноза.

- Переобучение: При работе с автоматически собранными данными существует риск переобучения модели, что приводит к снижению ее эффективности на новых данных.

- Генерализация: Модель, обученная на данных одной выборки, может быть плохо применима к другим наборам данных, особенно при изменении структуры или содержания исходных данных.

3. Ограничения использования

- Требования к актуальности: Автоматически собираемые данные могут быстро устаревать, что снизит эффективность прогнозов. Для поддержания актуальности модели потребуется регулярное обновление данных и, возможно, повторное обучение.

- Учет человеческого фактора: Система оценивает вероятность покупки на основе данных, но не может учитывать все субъективные и эмоциональные аспекты, влияющие на принятие решений клиентом.

- Ограниченная функциональность: Система предоставляет только вероятностную оценку склонности клиента к покупке и не формирует рекомендации для менеджеров по продажам.

2 Определение требований к системе

2.1 Функциональные требования

Система определения склонности клиента к приобретению машиноместа должна обеспечивать выполнение следующих функциональных задач:

- Принимать собранные данные о клиентах в установленном формате (файл формата csv)
- Проводить предварительную обработку данных, включая обработку пропусков и нормализацию числовых данных.
- На основе входных данных возвращать бинарный результат: "купит" или "не купит" ли клиент машиноместо.
- Сохранять результаты прогнозов для последующего анализа.

2.2 Требования к графическому интерфейсу

Графический интерфейс системы должен быть интуитивно понятным и удобным для пользователя, выполнять следующие функции:

- позволять загрузить данные
- обработать их
- предварительно просмотреть результат
- сохранить результат

2.3 Техническое задание

Цели:

Создать систему, которая позволит предсказать склонен ли клиент к покупке машиноместа.

Основные функции системы: создание прогноза на основе автоматически собранных данных. Интерфейс для загрузки и обработки данных, предпросмотра и сохранения результата.

Платформа и инструменты:

Язык программирования: Python.

Библиотеки для обработки данных и машинного обучения: pandas, NumPy, CatBoostClassifier, pickle.

Библиотека для создания интерфейса: tkinter

Архитектура системы: монолитная, десктопное приложение.

3 Этапы разработки

3.1 Сбор и анализ данных

В данном случае сбор данных не требуется, т. к. используются уже готовые датасеты предоставленные компанией «Самолет». Большая часть данных представляет собой просто пронумерованные столбцы, т. е. существуют множества значений, которые нет возможности идентифицировать, поэтому понять какие факторы повлияют на прогноз — практически невозможно.

В ходе анализа данных было решено удалить столбцы, где пропущено больше 50% Также в ходе анализа был удален столбец содержащий неизвестные хеши ввиду невозможности определить что в них зашифровано, а следовательно невозможности правильно перекодировать данные для

обучения модели. Далее все пропуски были заполнены нулевыми значениями. На этом предобработка данных была завершена.

3.2 Разработка модели

Для решения поставленной задачи был выбран алгоритм градиентного бустинга CatBoostClassifier. Данный выбор обусловлен следующими причинами:

- Высокая точность прогнозов: CatBoostClassifier демонстрирует высокую точность прогнозов благодаря использованию градиентного бустинга и различных методов регуляризации.

- Скорость обучения и масштабируемость: Алгоритм оптимизирован для работы с большими объемами данных и позволяет строить модели в разумные сроки.

- Удобство использования: Простой интерфейс библиотеки CatBoost и гибкие настройки гиперпараметров делают алгоритм удобным в использовании.

3.3 Разработка интерфейса

Интерфейс на tkinter устроен следующим образом:

Сверху в ряд располагаются три кнопки: "Загрузить файл", "Обработать данные" и "Сохранить файл".

Ниже, в ряд располагается три надписи (объект label), в них отображается статус выполнения программы.

В самом низу располагается таблица, для предпросмотра результатов.

В начале работы с приложением активна только одна кнопка: "Загрузить файл". Загрузка выполняется посредством диалогового окна выбора файла. После загрузки становится доступна кнопка "Обработать данные", а после обработки соответственной и "Сохранить файл". Сохранение выполняется посредством соответствующего диалогового окна. После обработки данных, они отображаются в таблице. В надписи под кнопкой загрузки изначально содержится текст "Загрузите данные", после загрузки он меняется на "Данные загружены!". После обработки под кнопкой обработки появляется текст "Данные обработаны!", а под кнопкой сохранения "Сохранить данные?". После сохранения она меняется на "Данные сохранены!". При повторной загрузке данных последние две надписи очищаются.

Таким образом получается минималистичный и интуитивно понятный интерфейс.

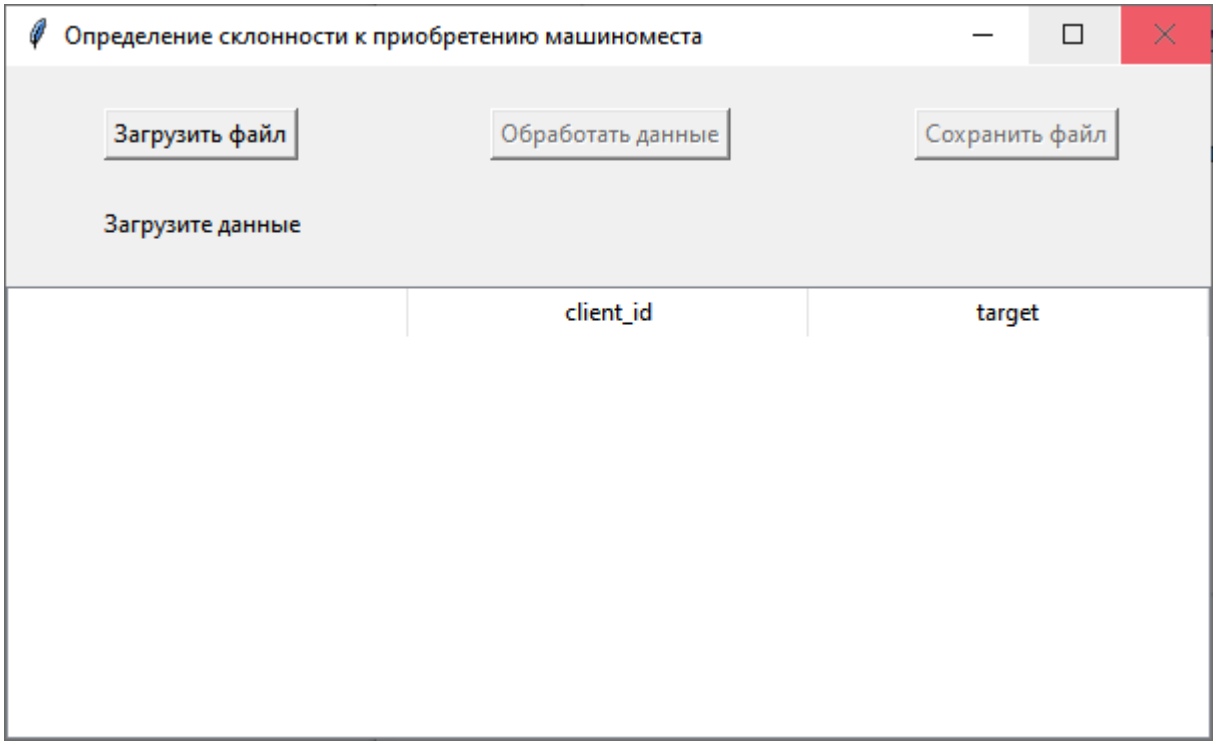


Рис. 1 Пользовательский интерфейс.

3.4 Тестирование системы

Объяснение результатов:

ROC-кривая (Receiver Operating Characteristic) – это графический инструмент, который позволяет оценить качество бинарного классификатора. Она отображает соотношение между долей истинно положительных классификаций (True Positive Rate, TPR) и долей ложно положительных классификаций (False Positive Rate, FPR) при различных порогах классификации.

Порог для "хорошего" значения AUC зависит от конкретной задачи и допустимого уровня ошибок. Однако, в общем случае:

- AUC < 0.7: Неудовлетворительный результат, модель работает плохо и требует существенной доработки.
- 0.7 < AUC < 0.8: Удовлетворительный результат, модель может быть использована, но возможно потребуется дополнительная настройка.
- AUC > 0.8: Считается хорошим результатом, модель демонстрирует высокую способность различать классы.
- AUC > 0.95: Может считаться отличным результатом, хотя с большей вероятностью свидетельствует о переобучении модели.

В нашем случае ROC-кривая показала 0.74, что говорит об удовлетворительном результате.

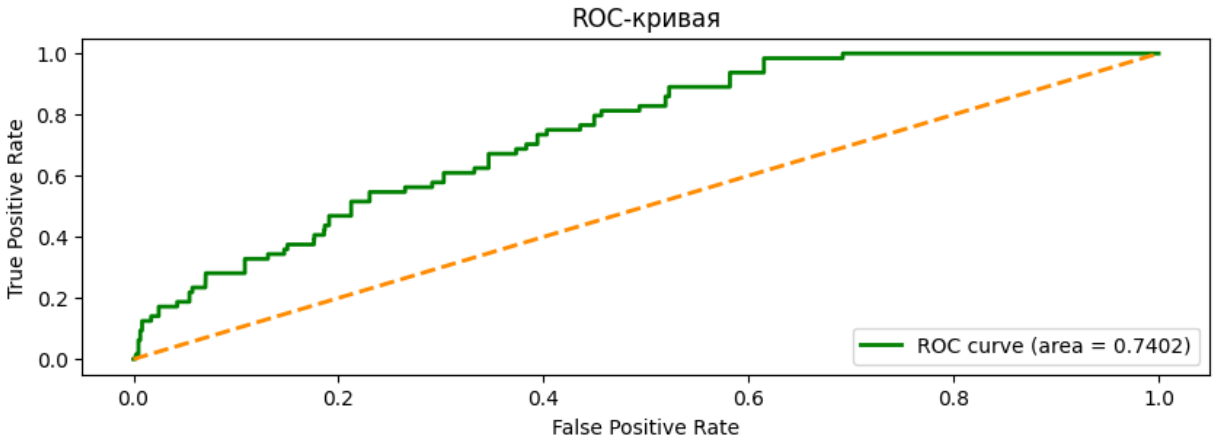


Рис. 2 ROC-Кривая модели

Заключение

В рамках данной курсовой работы была разработана и внедрена система прогнозирования склонности клиентов к приобретению машиноместа основанная на использовании методов машинного обучения. Система ориентирована на обработку данных, предоставленных компанией, и позволяет предсказать вероятность покупки машиноместа для каждого клиента на основе множества факторов.

Для построения модели был выбран алгоритм CatBoostClassifier, который несмотря на не очень высокое качество данных продемонстрировал отличные результаты при работе с ними и способен эффективно обрабатывать как малые, так и большие объемы данных. Алгоритм продемонстрировал хорошую точность и устойчивость к переобучению, что является важным аспектом в задачах классификации, особенно когда данные далеки от идеала.

На основе собранных данных был произведен анализ и предварительная обработка, в ходе которой были устранены ошибки и выбраны важнейшие признаки, которые могли бы повлиять на результат модели. Важно, работа с данными была организована таким образом, чтобы повысить стабильность и надежность прогнозов, а также уменьшить риски ошибок, связанных с неправильной обработкой данных.

Процесс тестирования системы показал ее способность точно и быстро выполнять прогнозы, что подтверждается хорошими значениями показателей точности, таких как AUC-ROC, и стабильностью результатов на разных выборках. Несмотря на это, система имеет потенциал для дальнейшего развиватия и требует дальнейшего тестирования и оптимизации

на новых данных для улучшения точности и адаптации к возможным изменениям в поведении клиентов.

В будущем следует рассмотреть возможность реализации автоматического анализа данных по мере их попадания в базу данных компании, а также автоматическое дообучение на свежих данных. Это позволит еще сильнее повысить её эффективность и точность.

В результате работы была создана система, которая может быть успешно использована для анализа клиентских данных и предсказания их склонности к приобретению машиномест. Это открывает возможности для повышения эффективности маркетинговых кампаний и продаж, а также для улучшения взаимодействия с клиентами компании, что в свою очередь может привести к росту бизнес-показателей и конкурентоспособности фирмы на рынке недвижимости.

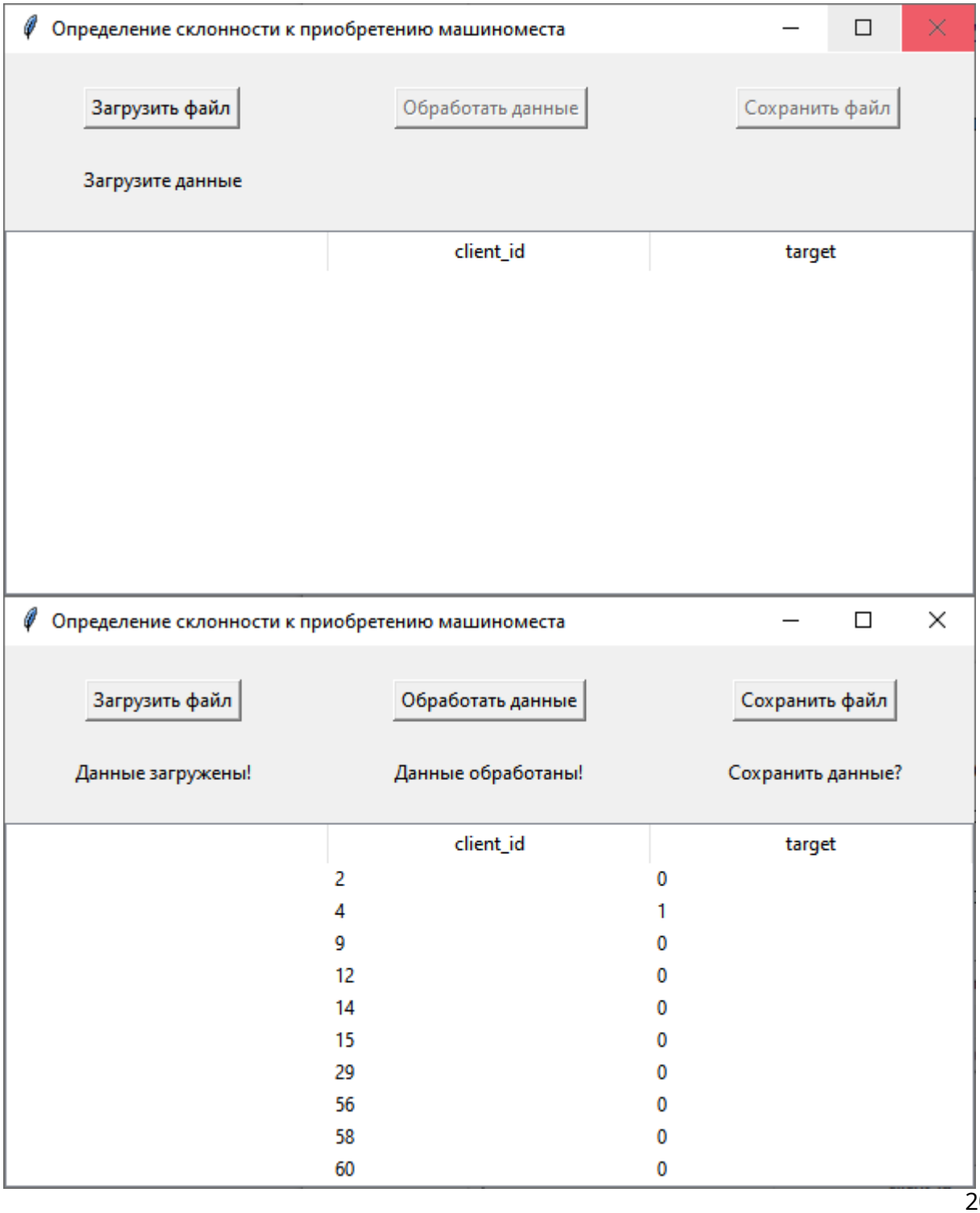
Список используемых источников

1. Официальная документация Python [Электронный ресурс]. URL: https://docs.python.org/3/ (Дата обращения 12.12.2024).
2. W3Schools Python Tutorial [Электронный ресурс]. URL: https://www.w3schools.com/python/ (Дата обращения 12.12.2024).
3. Официальная документация Pandas [Электронный ресурс]. URL: https://pandas.pydata.org/pandas-docs/stable/ (Дата обращения 13.12.2024).
4. TutorialsPoint: Pandas Tutorial [Электронный ресурс]. URL: https://www.tutorialspoint.com/python_pandas/index.htm (Дата обращения 13.12.2024).
5. Официальная документация Python по модулю Pickle [Электронный ресурс]. URL: https://docs.python.org/3/library/pickle.html (Дата обращения 15.14.2024).
6. Официальная документация Tkinter [Электронный ресурс]. URL: https://docs.python.org/3/library/tkinter.html (Дата обращения 15.14.2024).
7. Руководство по программированию на Tkinter и Python [Электронный ресурс]. URL: https://metanit.com/python/tkinter/ (Дата обращения 14.12.2024).
8. Курс по машинному обучению от Google (TensorFlow) [Электронный ресурс]. URL: https://www.tensorflow.org/tutorials (Дата обращения 15.12.2024).
9. Официальная документация библиотеки CatBoost [Электронный ресурс]. URL: https://catboost.ai/docs/ (Дата обращения 15.12.2024).

Приложение

Приложение А
(обязательное)

Интерфейс



Приложение Б

(обязательное)

Листинг программы

Обучение модели

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
cd /content/drive/My Drive/ML/3/v2/
```

```
/content/drive/My Drive/ML/3/v2
```

```
df = pd.read_csv('train.csv')
```

```
<ipython-input-14-cd75a77447fc>:1: DtypeWarning: Columns (51,52,53,54,55,56,57
  df = pd.read_csv('train.csv')
```

```
df.head()
```

	report_date	client_id	target	col1	col2	col3	col4	col5	col6	col7	...
0	2022-11-01	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
1	2022-11-01	5	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
2	2022-05-01	6	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
3	2022-09-01	7	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
4	2022-08-01	8	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...

5 rows × 2666 columns

В датафрейме слишком много пропусков в столбцах. Необходимо убрать столбцы в которых слишком много пропусков, в остальных заполнить пустые значения.

```
#Удаление столбцов, где > 50% пропусков
threshold = len(df) * 0.5
df=df.dropna(axis=1, thresh=threshold)
```

```
#Удаление дублирующихся строк
df = df.drop_duplicates()
df.shape
```

```
df.shape
```

```
(14456, 343)
```

Как выяснилось, дубликатов в датафрейме не было

```
#Проверка к каким типам принадлежат столбцы
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14456 entries, 0 to 14455
Columns: 343 entries, report_date to col2663
dtypes: float64(338), int64(3), object(2)
memory usage: 37.8+ MB
```

```
#Определение столбцов принадлежащих типу object
df.select_dtypes(include=['object']).columns
```

```
Index(['report_date', 'col1454'], dtype='object')
```

```
df['col1454']
```

	col1454
0	00f3d719-a6ec-4960-ba01-8372eb705443
1	01febac0-b083-494e-8589-f98400074b94
2	0278175e-c0bb-4e1e-bfe5-20db6811d3e2
3	0364571d-5325-ed11-b823-005056b825cd
4	065957a6-9b2a-435a-b561-024e9e8d9ad7
...	...
14451	fc4b5aa0-c4d6-4394-93db-57a1505e66c4
14452	fcc827d8-b64b-4b59-b718-87f2a51d77a4
14453	fd5f22f5-36f6-4dca-9896-8c9e90bbb702
14454	fdf04fb7-5404-4a9c-ae5b-e1e893060631
14455	fe392549-9af9-4b12-921b-b391693b8b27

14456 rows × 1 columns

dtype: object

```
#Удаление столбца т.к. нет возможности определить, что там зашифровано
df=df.drop('col1454', axis=1)
```

```
#Заполнение пропусков нулями
df = df.fillna(0)
df.head()
```

	report_date	client_id	target	col1453	col1455	col1456	col1457	col1458
0	2022-11-01	1	0	0	0.0	0.0	0.0	0.0
1	2022-11-01	5	0	1	0.0	0.0	0.0	0.0
2	2022-05-01	6	0	0	0.0	0.0	0.0	0.0
3	2022-09-01	7	0	0	1.0	1.0	1.0	1.0
4	2022-08-01	8	0	0	0.0	0.0	0.0	0.0

5 rows × 342 columns

```
df.set_index('report_date', inplace = True)
```

```
#Количество уникальных клиентов
df['client_id'].nunique()
```

```
4817
```

```
#Массив уникальных id клиентов
unique_clients = df['client_id'].unique()
```

```
#Разделение датафрейма на train и test
np.random.seed(12832)
indices=np.random.choice(len(unique_clients), int(len(unique_clients)*0.8), replac
train_unique_clients = unique_clients[indices]
test_unique_clients=np.delete(unique_clients, indices)

train=df[df['client_id'].isin(train_unique_clients)]
test=df[df['client_id'].isin(test_unique_clients)]
```

```
x_train = train.drop('target', axis=1)
y_train = train['target']

x_test = test.drop('target', axis=1)
y_test = test['target']
```

```
pip install catboost
```

```
Collecting catboost
  Downloading catboost-1.2.7-cp310-cp310-manylinux2014_x86_64.whl.metadata (1.
```

```
  Downloading catboost-1.2.7-cp310-cp310-manylinux2014_x86_64.whl.metadata (1.
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: numpy<2.0,>=1.16.0 in /usr/local/lib/python3.10
Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pythor
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dis
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/d
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/di
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/d
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/di
Downloading catboost-1.2.7-cp310-cp310-manylinux2014_x86_64.whl (98.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.7/98.7 MB 9.1 MB/s eta 0:00:00
Installing collected packages: catboost
Successfully installed catboost-1.2.7
```

```
from catboost import CatBoostClassifier, Pool
from sklearn.metrics import classification_report
from sklearn.utils.class_weight import compute_class_weight


# Расчет дисбаланса классов
classes = np.unique(y_train)

# Вычисление весов для балансировки классов
weights = compute_class_weight(class_weight='balanced', classes=classes, y=y_train

# Создание словаря с весами классов
class_weights = dict(zip(classes, weights))


model = CatBoostClassifier(eval_metric = "AUC", class_weights=class_weights)
model.fit(x_train, y_train, eval_set=(x_test, y_test)) # обучение классификатора
prediction = model.predict(x_test) # передача тестовой выборки в модель
model
```

```
Learning rate set to 0.058002
0:      test: 0.6960678 best: 0.6960678 (0)     total: 223ms    remaining: 3m
1:      test: 0.7052945 best: 0.7052945 (1)     total: 327ms    remaining: 2m
2:      test: 0.7153648 best: 0.7153648 (2)     total: 440ms    remaining: 2m
3:      test: 0.7280511 best: 0.7280511 (3)     total: 536ms    remaining: 2m
4:      test: 0.7347113 best: 0.7347113 (4)     total: 611ms    remaining: 2m
5:      test: 0.7344963 best: 0.7347113 (4)     total: 714ms    remaining: 1m
6:      test: 0.7291841 best: 0.7347113 (4)     total: 788ms    remaining: 1m
7:      test: 0.7290876 best: 0.7347113 (4)     total: 897ms    remaining: 1m
8:      test: 0.7328037 best: 0.7347113 (4)     total: 1s       remaining: 1m
```

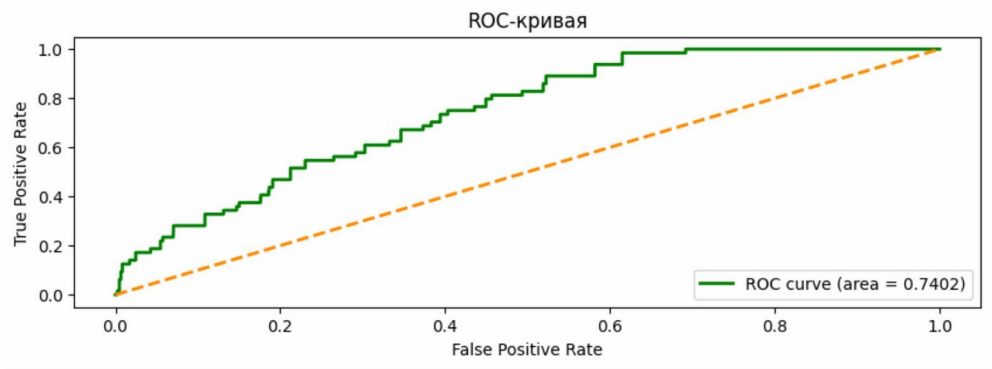
```
9:      test: 0.7322827 best: 0.7347113 (4)     total: 1.09s    remaining: 1m
10:     test: 0.7199464 best: 0.7347113 (4)     total: 1.16s    remaining: 1m
11:     test: 0.7299036 best: 0.7347113 (4)     total: 1.27s    remaining: 1m
12:     test: 0.7344632 best: 0.7347113 (4)     total: 1.35s    remaining: 1m
13:     test: 0.7296362 best: 0.7347113 (4)     total: 1.48s    remaining: 1m
14:     test: 0.7253358 best: 0.7347113 (4)     total: 1.56s    remaining: 1m
15:     test: 0.7247348 best: 0.7347113 (4)     total: 1.65s    remaining: 1m
16:     test: 0.7251704 best: 0.7347113 (4)     total: 1.71s    remaining: 1m
17:     test: 0.7237231 best: 0.7347113 (4)     total: 1.78s    remaining: 1m
18:     test: 0.7274667 best: 0.7347113 (4)     total: 1.85s    remaining: 1m
19:     test: 0.7346672 best: 0.7347113 (4)     total: 1.93s    remaining: 1m
20:     test: 0.7378043 best: 0.7378043 (20)    total: 1.98s    remaining: 1m
21:     test: 0.7342206 best: 0.7378043 (20)    total: 2.03s    remaining: 1m
22:     test: 0.7305101 best: 0.7378043 (20)    total: 2.08s    remaining: 1m
23:     test: 0.7302951 best: 0.7378043 (20)    total: 2.16s    remaining: 1m
24:     test: 0.7332889 best: 0.7378043 (20)    total: 2.22s    remaining: 1m
25:     test: 0.7341324 best: 0.7378043 (20)    total: 2.31s    remaining: 1m
26:     test: 0.7401916 best: 0.7401916 (26)    total: 2.39s    remaining: 1m
27:     test: 0.7390890 best: 0.7401916 (26)    total: 2.46s    remaining: 1m
28:     test: 0.7386148 best: 0.7401916 (26)    total: 2.53s    remaining: 1m
29:     test: 0.7336362 best: 0.7401916 (26)    total: 2.61s    remaining: 1m
30:     test: 0.7301242 best: 0.7401916 (26)    total: 2.71s    remaining: 1m
31:     test: 0.7267830 best: 0.7401916 (26)    total: 2.81s    remaining: 1m
32:     test: 0.7237893 best: 0.7401916 (26)    total: 2.93s    remaining: 1m
33:     test: 0.7261931 best: 0.7401916 (26)    total: 3.03s    remaining: 1m
34:     test: 0.7250739 best: 0.7401916 (26)    total: 3.14s    remaining: 1m
35:     test: 0.7226645 best: 0.7401916 (26)    total: 3.25s    remaining: 1m
36:     test: 0.7212255 best: 0.7401916 (26)    total: 3.35s    remaining: 1m
37:     test: 0.7198472 best: 0.7401916 (26)    total: 3.38s    remaining: 1m
38:     test: 0.7261214 best: 0.7401916 (26)    total: 3.45s    remaining: 1m
39:     test: 0.7241917 best: 0.7401916 (26)    total: 3.54s    remaining: 1m
40:     test: 0.7238003 best: 0.7401916 (26)    total: 3.59s    remaining: 1m
41:     test: 0.7241972 best: 0.7401916 (26)    total: 3.65s    remaining: 1m
42:     test: 0.7285969 best: 0.7401916 (26)    total: 3.69s    remaining: 1m
43:     test: 0.7275715 best: 0.7401916 (26)    total: 3.74s    remaining: 1m
44:     test: 0.7304274 best: 0.7401916 (26)    total: 3.79s    remaining: 1m
45:     test: 0.7223723 best: 0.7401916 (26)    total: 3.85s    remaining: 1m
46:     test: 0.7196707 best: 0.7401916 (26)    total: 3.9s     remaining: 1m
47:     test: 0.7247596 best: 0.7401916 (26)    total: 3.95s    remaining: 1m
48:     test: 0.7268134 best: 0.7401916 (26)    total: 4.01s    remaining: 1m
49:     test: 0.7264770 best: 0.7401916 (26)    total: 4.07s    remaining: 1m
50:     test: 0.7228768 best: 0.7401916 (26)    total: 4.14s    remaining: 1m
51:     test: 0.7240897 best: 0.7401916 (26)    total: 4.17s    remaining: 1m
52:     test: 0.7228768 best: 0.7401916 (26)    total: 4.22s    remaining: 1m
53:     test: 0.7255563 best: 0.7401916 (26)    total: 4.28s    remaining: 1m
54:     test: 0.7242083 best: 0.7401916 (26)    total: 4.33s    remaining: 1m
55:     test: 0.7256583 best: 0.7401916 (26)    total: 4.38s    remaining: 1m
56:     test: 0.7260056 best: 0.7401916 (26)    total: 4.44s    remaining: 1m
```

```
print(classification_report(y_test, prediction))
```

```
              precision    recall  f1-score   support

           0       0.98      0.83      0.90      2834
           1       0.05      0.38      0.08        64
```

```
                _       ...       ...       ...       ...

         accuracy                           0.82      2898
        macro avg       0.52      0.60      0.49      2898
     weighted avg       0.96      0.82      0.88      2898
```

```
from sklearn.metrics import roc_auc_score, roc_curve

y_prediction = model.predict_proba(x_test)[:,1]

lr_auc = roc_auc_score(y_test, y_prediction)

fpr, tpr, _ = roc_curve(y_test, y_prediction)
roc_auc = roc_auc_score(y_test, y_prediction)

plt.figure(figsize=(10, 3))
plt.plot(fpr, tpr, color='green',
             lw=2, label='ROC curve (area = %0.4f)' % roc_auc)

plt.plot([0, 1], [0, 1], color='darkorange', lw=2, linestyle='--')

plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC-кривая')
plt.legend(loc="lower right")
plt.show()
```




```
import pickle

# Сохранить модель в файл
with open('model.pkl', 'wb') as file:
    pickle.dump(model, file)
```

Код приложения

```
import tkinter as tk
from tkinter import filedialog
from tkinter import ttk
import pandas as pd
import numpy as np
import pickle
from catboost import CatBoostClassifier, Pool

def load_data():
    global input
    file_path = filedialog.askopenfilename(
        title="Выберите файл",
        filetypes=(("Таблицы", "*.csv"), ("Все файлы", "*.*"))
    )
    if file_path:  # Если файл выбран
        input = pd.read_csv(file_path)
        table.delete(*table.get_children())
        process_button.config(state=tk.NORMAL)
        lbl4.config(text="Данные загружены!")
        lbl5.config(text="")
        lbl6.config(text="")
        save_button.config(state=tk.DISABLED)


def process_data():
    global results

    # предобработка
    input_ = input.fillna(0)
    input_.set_index('report_date', inplace=True)
    # Удаление столбцов не подходящих модели
    columns_to_keep = ['report_date']  # Исключаемый столбец
    columns_to_drop = [
        col for col in input_.select_dtypes(include='object').columns
        if col not in columns_to_keep
    ]
    input_ = input_.drop(columns=columns_to_drop)

    # Анализ данных
    predictions = model.predict(input_)

    # Результат
    client_ids = input['client_id']
    results = pd.DataFrame({
        'client_id': client_ids,
        'target': predictions
    })
```

```
    lbl5.config(text="Данные обработаны!")
    lbl6.config(text="Сохранить данные?")
    # Вывод в таблицу
    for row in table.get_children():
        table.delete(row)
    for _, row in results.iterrows():
        table.insert("", "end", values=(row['client_id'], row['target']))

    save_button.config(state=tk.NORMAL)


def save_data():
    file_path = filedialog.asksaveasfilename(defaultextension=".csv")
    if file_path:
        results.to_csv(file_path, index=False)
        lbl6.config(text="Данные сохранены!")


with open('model.pkl', 'rb') as file:
    model = pickle.load(file)



# Создаем главное окно
root = tk.Tk()
root.title("Определение склонности к приобретению машиноместа")

# Создание отступов
lbl1 = tk.Label(root)
lbl1.grid(row=4)

lbl2 = tk.Label(root)
lbl2.grid(row=0)

lbl3 = tk.Label(root)
lbl3.grid(row=2)


# Кнопки для загрузки, сохранения и обработки
load_button = tk.Button(root, text="Загрузить файл", command=load_data)
load_button.grid(row=1, column=0)


process_button = tk.Button(root, text="Обработать данные", command=process_data,
state=tk.DISABLED)
process_button.grid(row=1, column=1)


save_button = tk.Button(root, text="Сохранить файл", command=save_data,
state=tk.DISABLED)
```

```
save_button.grid(row=1, column=2)


# Создание полей для вывода статуса
lbl4 = tk.Label(root, text='Загрузите данные')
lbl4.grid(row=3, column=0)


lbl5 = tk.Label(root, text='')
lbl5.grid(row=3, column=1)

lbl6 = tk.Label(root, text='')
lbl6.grid(row=3, column=2)

# Создание таблицы
table = ttk.Treeview(root, columns=("Column1", "Column2"))
table.heading("Column1", text="client_id")
table.heading("Column2", text="target")
table.grid(row=5, column=0, columnspan=3)


root.mainloop()
```