B2B Sales Predictive Model

Your Name

November 8, 2024

Contents

1 Project Overview

This project involves building a predictive model to estimate monthly sales and classify leads into sales tiers for B2B e-commerce lead generation. The model aims to:

- **Predict Monthly Sales**: Provide accurate estimates for sales prioritization.
- **Classify Sales Tiers**: Segment leads into meaningful tiers for tailored strategies.
- **Generalize to Non-Customers**: Apply the model to potential leads.
- **Provide Actionable Insights**: Identify key factors influencing sales.

2 Folder Structure

The repository is organized as follows:

```
b2b-sales-predictive-model/
|-- data/
|   |-- raw/                # Original data files (not tracked by Git)
|   |-- processed/          # Preprocessed data files
|-- notebooks/
|   |-- 01_data_exploration.ipynb
|   |-- 02_data_preprocessing.ipynb
|   |-- 03_modeling.ipynb
|   |-- 04_evaluation.ipynb
|   |-- 05_insights.ipynb
|-- scripts/
|   |-- data_preprocessing.py
|   |-- modeling.py
|   |-- evaluation.py
|-- reports/
|   |-- presentation.pdf    # Final presentation
|-- venv/                   # Virtual environment directory (not tracked by Git)
|-- requirements.txt        # Python dependencies
|-- setup.sh                # Bash script for automated setup
|-- README.tex              # This README file in LaTeX format
|-- .gitignore
|-- LICENSE                 # Optional
```

3 Prerequisites

- **Operating System**: macOS
- **Python Version**: Python 3.6 or higher installed on your system

4 Setup Instructions

4.1 Clone the Repository

Open your terminal and clone the repository:

```
git clone https://github.com/yourusername/b2b-sales-predictive-model.git
cd b2b-sales-predictive-model
```

Note: Replace `https://github.com/yourusername/b2b-sales-predictive-model.git` with the actual URL of your GitHub repository.

4.2 Create and Activate the Virtual Environment

Create a Python virtual environment using `venv`:

```
python3 -m venv venv
```

Activate the virtual environment:

```
source venv/bin/activate
```

4.3 Install Dependencies

Upgrade `pip` and install the required packages:

```
pip install --upgrade pip
pip install -r requirements.txt
```

4.4 Set Up Jupyter Kernel

Add the virtual environment as a Jupyter kernel:

```
python -m ipykernel install --user --name=b2b-sales-env
```

4.5 Launch Jupyter Notebook

Start the Jupyter Notebook server:

```
jupyter notebook
```

When you open a notebook, select the kernel named `b2b-sales-env`:

- Navigate to **Kernel** → **Change kernel** → `b2b-sales-env`

5 Usage

With the environment set up, you can proceed to:

1. **Data Exploration**: Open `notebooks/01_data_exploration.ipynb` to explore the data.
2. **Data Preprocessing**: Use `notebooks/02_data_preprocessing.ipynb` to clean and preprocess the data.
3. **Modeling**: Build predictive models in `notebooks/03_modeling.ipynb`.
4. **Evaluation**: Evaluate model performance in `notebooks/04_evaluation.ipynb`.
5. **Insights**: Analyze feature importance and derive insights in `notebooks/05_insights.ipynb`.

6 Bash Script for Automated Setup

To automate the setup process, you can run the provided `setup.sh` bash script. This script will:

- Create the required folder structure
- Create and activate the virtual environment
- Install dependencies
- Set up the Jupyter kernel

6.1 Running the Script

Make sure you have execution permissions and then run the script:

```
chmod +x setup.sh
./setup.sh
```

6.2 `setup.sh` Contents

Listing 1: setup.sh

```
#!/bin/bash

# Navigate to the project directory
cd "$(dirname "$0")"

echo "Creating folder structure..."

# Create folders
mkdir -p data/raw data/processed notebooks scripts reports
```

```
# Create notebook files
touch notebooks/01_data_exploration.ipynb \
      notebooks/02_data_preprocessing.ipynb \
      notebooks/03_modeling.ipynb \
      notebooks/04_evaluation.ipynb \
      notebooks/05_insights.ipynb

# Create script files
touch scripts/data_preprocessing.py \
      scripts/modeling.py \
      scripts/evaluation.py

# Create other files
touch requirements.txt LICENSE

echo "Adding content to requirements.txt..."

# Add content to requirements.txt
cat > requirements.txt << EOL
pandas
numpy
scikit-learn
matplotlib
seaborn
jupyter
ipykernel
xgboost
lightgbm
shap
EOL

echo "Configuring .gitignore..."

# Create .gitignore if it doesn't exist
if [ ! -f .gitignore ]; then
    cat > .gitignore << EOL
# Ignore data files
/data/raw/
/data/processed/

# Ignore virtual environment folder
venv/

# Ignore Jupyter Notebook checkpoints
.ipynb_checkpoints
```

```
# Ignore system files
.DS_Store
EOL
fi

echo "Initializing git repository..."

# Initialize git repository if not already initialized
if [ ! -d .git ]; then
    git init
    git add .
    git commit -m "Initial commit with folder structure and basic files"
fi

echo "Setting up virtual environment..."

# Create virtual environment and activate it
python3 -m venv venv
source venv/bin/activate

echo "Upgrading pip and installing dependencies..."

# Upgrade pip
pip install --upgrade pip

# Install dependencies
pip install -r requirements.txt

echo "Registering Jupyter kernel..."

# Register the virtual environment kernel for Jupyter
python -m ipykernel install --user --name=b2b-sales-env

echo "Setup complete. You can now launch Jupyter Notebook using 'jupyter notebook

# Deactivate virtual environment after setup
deactivate
```

7 License

This project is licensed under the MIT License – see the `LICENSE` file for details.

8 Troubleshooting

8.1 Permission Denied Error When Running `setup.sh`

Ensure the script has execute permissions:

```
chmod +x setup.sh
```

8.2 Jupyter Kernel Not Showing Up

If the `b2b-sales-env` kernel doesn't appear in Jupyter Notebook, try reinstalling the kernel:

```
python -m ipykernel install --user --name=b2b-sales-env --display-name "Python (b2b-sales-en
```

8.3 Python Version Issues

Make sure you're using the correct version of Python. Check your Python version with:

```
python3 --version
```

8.4 Deactivating the Virtual Environment

When you're done working, deactivate the virtual environment:

```
deactivate
```

8.5 Reactivating the Virtual Environment

Before starting work in a new terminal session:

```
source venv/bin/activate
```

8.6 Installing Additional Packages

If you need to install new packages, make sure the virtual environment is activated and then install the package:

```
pip install package_name
```

Update the `requirements.txt` file:

```
pip freeze > requirements.txt
```