CREDIT CARD DETECTION

Author: Badal Singh

Email: singh.badal3375@gmail.com

⚠ *These are synthetic features (not real-world transaction fields like amount, time, etc.), because the demo model was trained on synthetic data. In a real project, you would replace these with actual transaction features (e.g., PCA components or domain-specific features).*

```
# Credit Card Detection using Machine Learning

This project focuses on detecting potential fraudulent credit card transactions using **M

---

## 🚀 Features
- Data preprocessing and cleaning with **Pandas** and **NumPy**
- Data visualization with **Matplotlib** and **Seaborn**
- Feature extraction/engineering and text/vector transformations
- Machine learning model training using **Scikit-learn**
- Model saving and loading with **Pickle**
- Interactive **Streamlit** web app for real-time predictions

---

## 🛠 Tech Stack
- **Programming Language**: Python 3.x
- **Libraries**:
  - Pandas (data manipulation)
  - NumPy (numerical computation)
  - Matplotlib & Seaborn (visualization)
  - Scikit-learn (ML algorithms, evaluation metrics)
  - Pickle (model saving/loading)
  - Vectorizers (feature transformation)
- **Frontend/UI**: Streamlit

---

## 📂 Project Structure
```

credit-card-detection/
│
├── data/ # Dataset files (CSV or others)
├── notebooks/ # Jupyter notebooks for exploration
├── models/ # Saved pickle model files
├── src/ # Source code for preprocessing & training
│ ├── preprocess.py # Data cleaning & feature engineering
│ ├── train_model.py # ML training script
│ ├── evaluate_model.py # Model evaluation
├── app.py # Streamlit web application
├── requirements.txt # Dependencies
└── README.md # Project documentation

```
---

## ⚙ Installation & Setup

1. **Clone the repository**
```

git clone https://github.com/your-username/credit-card-detection_1.git
cd credit-card-detection

```
2. **Create virtual environment & install dependencies**
```

pip install -r requirements.txt

```
3. **Run the Streamlit app**
```

streamlit run app.py

```
---

## Workflow

1. **Data Preprocessing & Cleaning**
- Handle missing values, duplicates, and scaling.
- Convert categorical data using encoders/vectorizers.

2. **Exploratory Data Analysis (EDA)**
- Visualize distributions, correlations, and patterns using Seaborn & Matplotlib.

3. **Model Training & Evaluation**
- Use ML algorithms from Scikit-learn (Logistic Regression, Random Forest, etc.).
- Evaluate with accuracy, precision, recall, F1-score, ROC curve, etc.
```

```
4. **Model Saving & Deployment**
- Save trained models with **Pickle**.
- Connect with Streamlit frontend for real-time predictions.

---

## 🌐 Frontend (Streamlit)
The Streamlit app provides a user-friendly web interface where users can:
- Input transaction details.
- Get real-time fraud detection results.
- Visualize prediction probabilities and model performance.

---

## 🔮 Future Improvements
- Integration with deep learning methods (e.g., TensorFlow/PyTorch).
- Deployment on cloud platforms (Heroku, AWS, GCP).
- Real-time API integration for production use.

---

## 🤝 Contributing
Contributions are welcome! Please fork the repo and submit a pull request.

---

## 📜 License
This project is open-source under the **MIT License**.

---

## 📦 requirements.txt
```

pandas2.2.2
numpy1.26.4
matplotlib3.9.2
seaborn0.13.2
scikit-learn1.5.1
streamlit1.38.0
pickle-mixin==1.0.2

```
If you use **Jupyter notebooks**, also add:
```

notebook
ipykernel

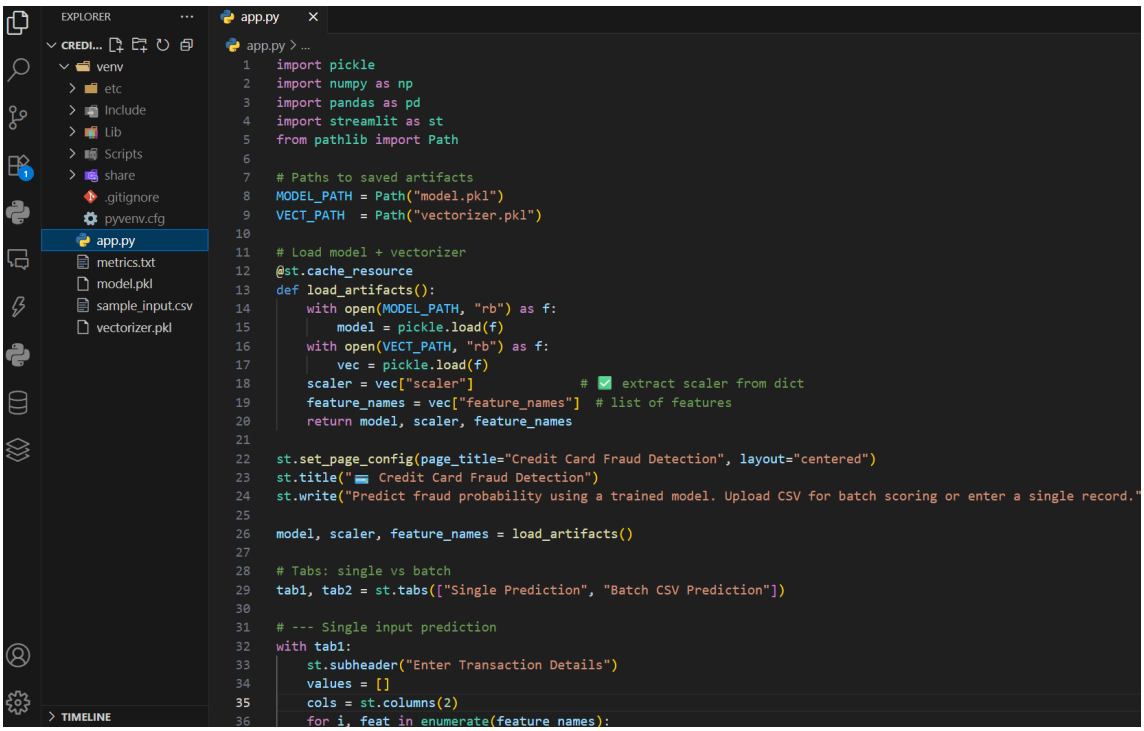
```
```

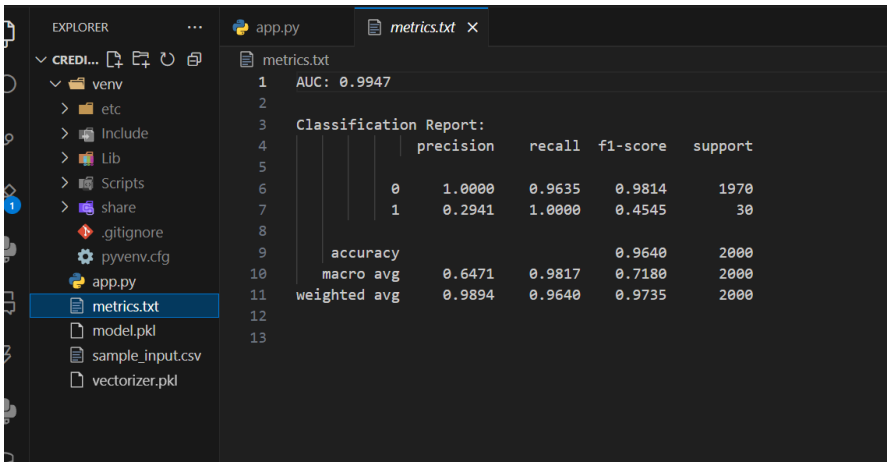
To Convert to PDF:

Save this content to a file named `README.md`, then run

```
pip install markdown-pdf
python -c "from markdown_pdf import MarkdownPdf, Section; pdf=MarkdownPdf(); pdf.add_sect
```

This will produce a high-quality PDF with your author information at the top.





Final result