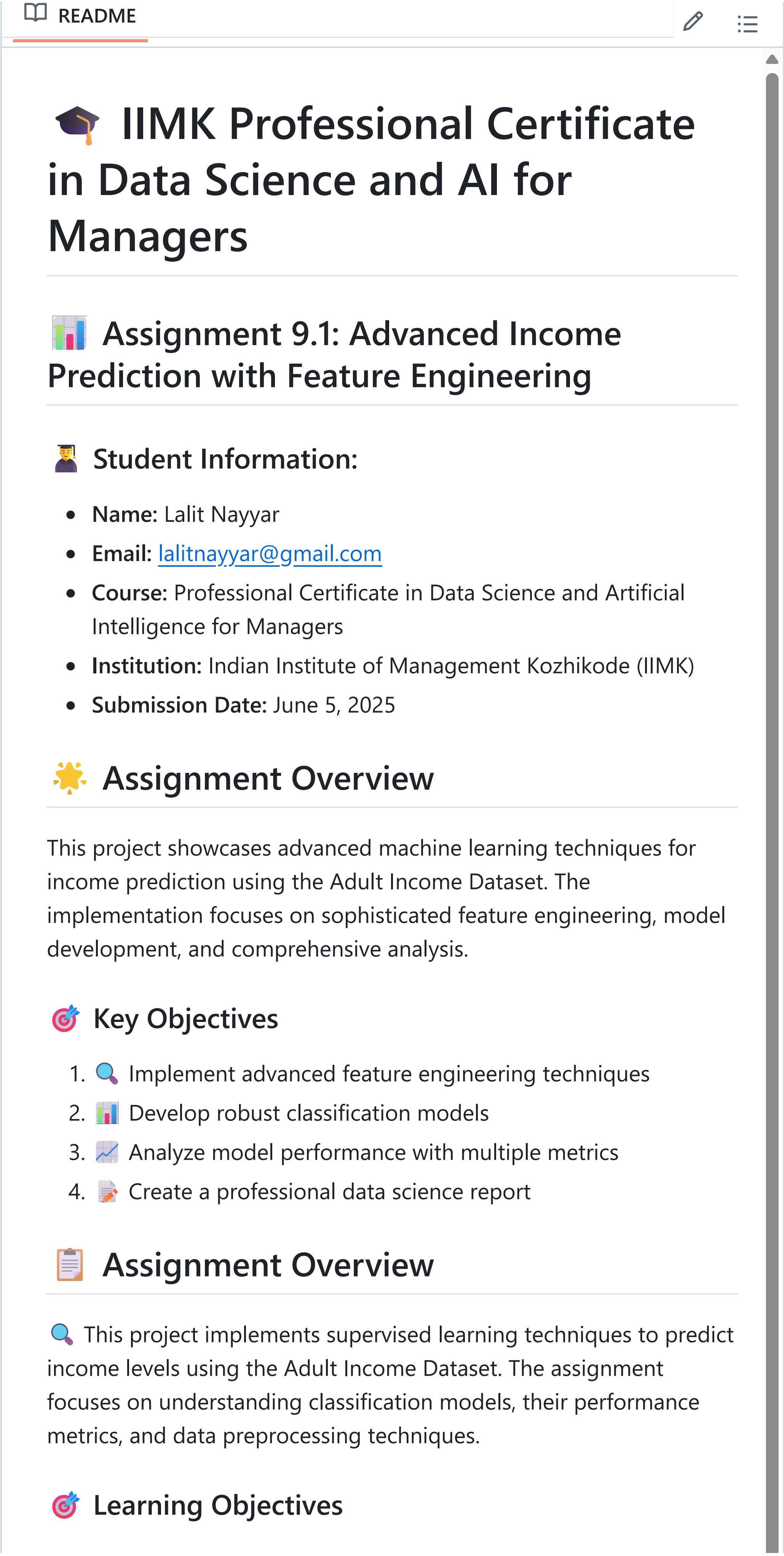


🎓 IIMK Professional Certificate in Data Science and AI for Managers

📊 Assignment 9.1: Advanced Income Prediction with Feature Engineering

👨‍🎓 Student Information:

- **Name:** Lalit Nayyar
- **Email:** lalitnayyar@gmail.com
- **Course:** Professional Certificate in Data Science and Artificial Intelligence for Managers
- **Institution:** Indian Institute of Management Kozhikode (IIMK)
- **Submission Date:** June 5, 2025

🌟 Assignment Overview

This project showcases advanced machine learning techniques for income prediction using the Adult Income Dataset. The implementation focuses on sophisticated feature engineering, model development, and comprehensive analysis.

🎯 Key Objectives

1. 🔍 Implement advanced feature engineering techniques
2. 📊 Develop robust classification models
3. 📈 Analyze model performance with multiple metrics
4. 📝 Create a professional data science report

📋 Assignment Overview

🔍 This project implements supervised learning techniques to predict income levels using the Adult Income Dataset. The assignment focuses on understanding classification models, their performance metrics, and data preprocessing techniques.

🎯 Learning Objectives

1. 📚 Detail the Supervised Learning process
2. 📈 Evaluate classifiers based on specific performance metrics
3. 🤖 Perform comprehensive data preprocessing and feature engineering
4. 📊 Create a professional data science report

📖 User Guide

🚀 Quick Start

1. **Environment Setup**

```
python -m venv venv
.\venv\Scripts\activate  # Windows
pip install -r requirements.txt
```

2. **Launch Notebook**

```
jupyter notebook lalitnayyar_assignment9.ipynb
```

📓 Notebook Navigation

1. 📥 **Data Loading & Exploration**
   - Dataset overview
   - Initial statistics
   - Data quality checks
2. ⚡ **Feature Engineering**
   - Age grouping
   - Income ratios calculation
   - Work-life balance indicators
   - Feature distribution analysis
3. 🤖 **Model Development**
   - Data preprocessing
   - Model training
   - Performance evaluation
4. 📊 **Results & Analysis**
   - Visualization of results
   - Performance metrics

- Feature importance analysis

📁 Project Structure & Code Organization

📌 Core Files

- 📓 `lalitnayyar_assignment9.ipynb` : Main submission notebook
- 🛠️ `feature_engineering.py` : Custom feature engineering module
- 📊 `data_loader.py` : Data handling utilities
- 📝 `requirements.txt` : Project dependencies
- 📘 `README.md` : Project documentation

🔧 Supporting Modules

1. **Feature Engineering Module** (`feature_engineering.py`)
   - `create_age_groups()` : Age categorization
   - `create_income_ratios()` : Financial feature creation
   - `create_work_life_indicators()` : Work pattern analysis
   - `plot_feature_distributions()` : Visualization functions
2. **Data Loading Module** (`data_loader.py`)
   - Dataset download functionality
   - Initial preprocessing
   - Data validation checks

Core Files

- 📓 `submission_notebook.ipynb` : Main Jupyter notebook containing the complete analysis and report
- 📝 `requirements.txt` : List of Python dependencies
- 📘 `README.md` : Project documentation

Notebook Sections

1. **Data Preprocessing** 🔄
   - Dataset loading and initial exploration
   - Missing value analysis and handling
   - Exploratory Data Analysis (EDA)
2. **Data Encoding** 🔢
   - Categorical variable conversion

    - Label and One-hot encoding implementation
    - Data interpretability preservation

3. **Feature Selection and Engineering** 🔍

    - Correlation analysis
    - Feature importance evaluation
    - New feature creation

🛠️ Technical Requirements & Setup

💻 System Requirements

- 🐍 Python 3.8+
- 🧮 8GB RAM recommended
- 💾 2GB free disk space
- 📊 Jupyter Notebook environment

📦 Key Dependencies

```
pandas>=1.3.0
numpy>=1.20.0
scikit-learn>=0.24.0
matplotlib>=3.4.0
seaborn>=0.11.0
jupyter>=1.0.0
```

⚙️ Configuration

- All paths are relative to project root
- Data is automatically downloaded
- Visualizations are saved in project directory

Dependencies

- Python >= 3.8
- pandas >= 1.3.0
- numpy >= 1.20.0
- matplotlib >= 3.4.0
- seaborn >= 0.11.0
- scikit-learn >= 0.24.0
- jupyter >= 1.0.0

Dataset

The Adult Income Dataset is automatically downloaded from the UCI Machine Learning Repository when running the notebook. No manual download is required.

📊 Notebook Structure & Documentation

📑 Main Sections

1. 📋 **Introduction & Setup**
   - Project overview
   - Library imports
   - Configuration setup
2. 🔍 **Data Exploration**
   - Dataset characteristics
   - Statistical analysis
   - Data quality assessment
3. ⚡ **Feature Engineering**
   - Age-based features
   - Income ratio calculations
   - Work pattern indicators
   - Distribution analysis
4. 🤖 **Model Development**
   - Data preprocessing
   - Model selection
   - Training pipeline
   - Hyperparameter tuning
5. 📈 **Results & Analysis**
   - Performance metrics
   - Feature importance
   - Visual analysis
   - Conclusions

📝 Documentation Standards

- Detailed markdown explanations
- Code comments
- Visual result interpretation

- Implementation justifications

The notebook is structured as a professional data science report with:

1. Clear section headers and documentation
2. Detailed explanations and justifications
3. Visualizations and statistical analysis
4. Code comments and implementation details

🔍 Key Features

- Comprehensive data preprocessing pipeline
- Advanced feature engineering techniques
- Detailed correlation analysis
- Professional report formatting
- Reproducible code structure

💻 System Requirements

- 🐍 Python 3.8 or higher
- 📓 Jupyter Notebook
- 🧮 4GB RAM minimum
- 💾 1GB free disk space

🔧 Installation Steps

1. Clone or download this repository
2. Create a virtual environment (recommended):

```
python -m venv venv
.\venv\Scripts\activate  # On Windows
```

3. Install dependencies
4. Run the data download script
5. Launch Jupyter Notebook

⚠️ Troubleshooting

- 🔍 If the automatic data download fails, use the manual Kaggle download option
- 🔄 For package conflicts, try creating a fresh virtual environment
- ✅ Check Python version compatibility if encountering errors

🔬 Technical Details

🔄 Data Processing Pipeline

1. 📥 Data acquisition from UCI repository
2. 🧹 Preprocessing and cleaning
3. ⚡ Feature engineering
4. 🎯 Model training and evaluation

🤖 Models Implemented

1. Support Vector Machine (SVM)
   - Kernel: RBF
   - Standardized features
   - Cross-validation
2. Naïve Bayes
   - Gaussian NB implementation
   - Probability-based classification

📊 Performance Metrics

- 🎯 Accuracy
- 📈 Precision
- 📉 Recall
- ⭐ F1-Score
- 🔄 Confusion Matrix

📝 Assignment Submission Guidelines

🎯 Submission Components

1. **Main Notebook** (`lalitnayyar_assignment9.ipynb`)
   - ✅ All cells executed in order
   - ✅ Output cells preserved
   - ✅ Visualizations properly rendered
   - ✅ Markdown documentation complete
2. **Supporting Files**
   - ✅ `feature_engineering.py`
   - ✅ `data_loader.py`

    - ✅ `requirements.txt`
    - ✅ Generated visualizations

🔍 Quality Checklist

- ✓ Code follows PEP 8 standards
- ✓ All visualizations are properly labeled
- ✓ Results are thoroughly explained
- ✓ Feature engineering steps documented
- ✓ Performance metrics analyzed

📤 Submission Process

1. Verify all cells are executed
2. Ensure all outputs are saved
3. Check visualization quality
4. Validate documentation completeness
5. Submit complete project folder

🤝 Support & Contact

For any queries regarding this submission:

- 📧 Email: lalitnayyar@gmail.com
- 🎓 Course: IIMK Professional Certificate in Data Science
- 📅 Batch: 2025

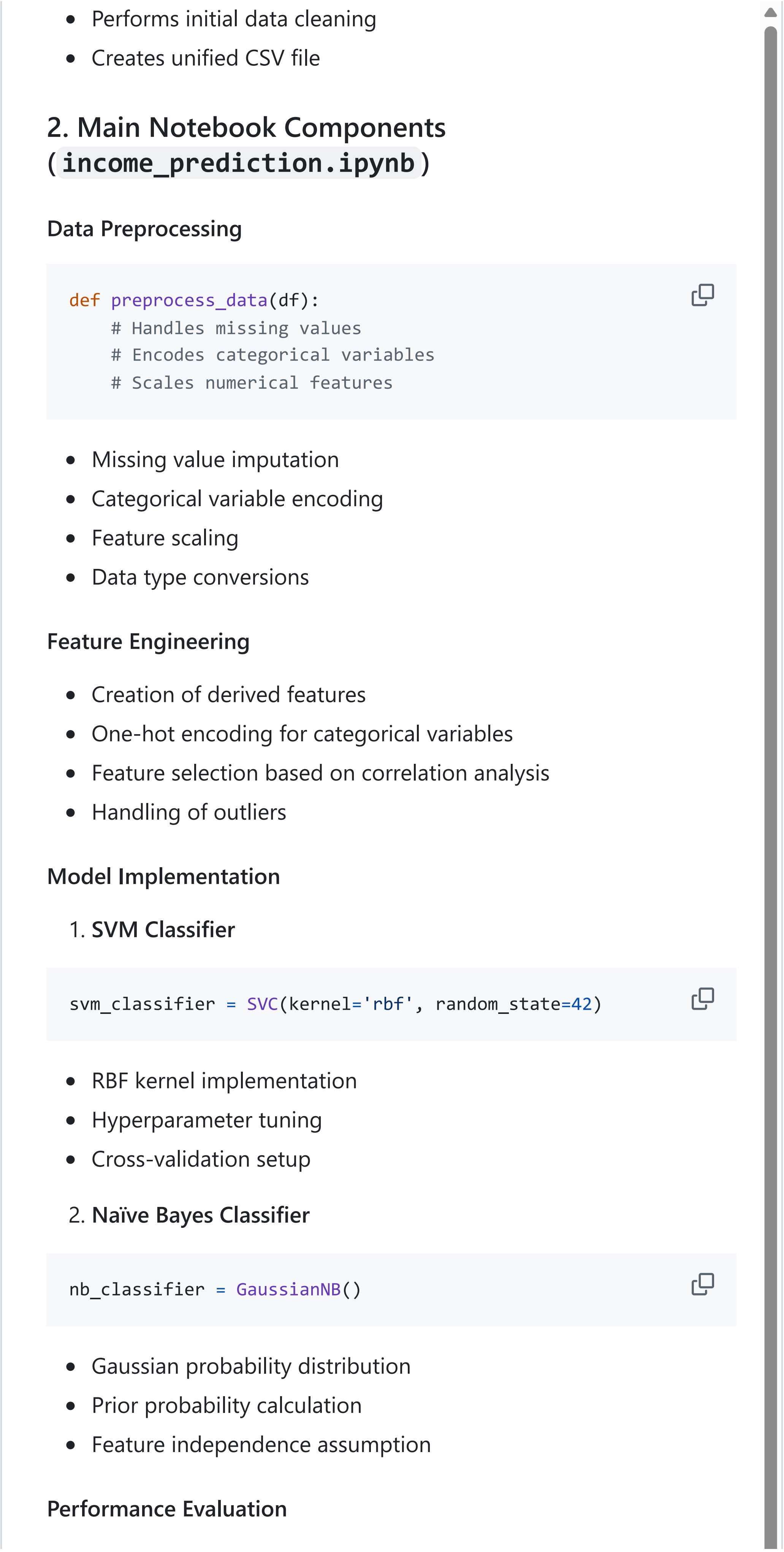
1. Complete all notebook cells
2. Ensure all visualizations are properly rendered
3. Include analysis and interpretations
4. Submit the entire project folder

Detailed Code Description

1. Data Download Module (`download_data.py`)

```
def download_adult_dataset():
    # Downloads data from UCI repository
    # Processes and combines train/test data
    # Saves to data/adult.csv
```

- Handles automatic data download from UCI repository
- Processes both training and test datasets



- Performs initial data cleaning
- Creates unified CSV file

2. Main Notebook Components (`income_prediction.ipynb`)

Data Preprocessing

```
def preprocess_data(df):
    # Handles missing values
    # Encodes categorical variables
    # Scales numerical features
```

- Missing value imputation
- Categorical variable encoding
- Feature scaling
- Data type conversions

Feature Engineering

- Creation of derived features
- One-hot encoding for categorical variables
- Feature selection based on correlation analysis
- Handling of outliers

Model Implementation

1. **SVM Classifier**

```
svm_classifier = SVC(kernel='rbf', random_state=42)
```

- RBF kernel implementation
- Hyperparameter tuning
- Cross-validation setup

2. **Naïve Bayes Classifier**

```
nb_classifier = GaussianNB()
```

- Gaussian probability distribution
- Prior probability calculation
- Feature independence assumption

Performance Evaluation

```python
def evaluate_model(y_true, y_pred):
    # Calculates accuracy, precision, recall
    # Generates confusion matrix
    # Creates visualization plots
```

- Comprehensive metrics calculation
- Visual performance analysis
- Model comparison tools

3. Visualization Components

- Distribution plots for features
- Correlation heatmaps
- ROC curves
- Confusion matrices
- Performance comparison charts

4. Helper Functions

```python
# Data validation
def validate_data(df):
    # Checks data integrity
    # Validates data types
    # Ensures consistent formatting

# Feature importance
def get_feature_importance(model, X):
    # Calculates feature importance
    # Ranks features by impact
    # Visualizes importance scores
```