Data science project

Disclaimer

This document is part of the Industry Job Package from www.academiatoindustry.com, created by Dr. Aleena Baby. The content and project ideas presented here are for educational and portfolio-building purposes only and are not affiliated with or endorsed by any specific company or organization.

The dataset mentioned is publicly available and should be used in accordance with the terms and conditions set by the data provider. For permissions regarding data usage, please contact the respective data provider directly.This project guide is designed to showcase skills and does not represent actual work done for any company. It is intended to serve as a template for creating a portfolio project.

If you have any comments, suggestions, or inquiries about this document or the Industry Job Package, please contact: a2i.academiatoindustry@gmail.com

How to use these project ideas

1. Choose a Project:
   Select any project idea that interests you or aligns with your career goals.
2. Acquire the Data:
   Download the suggested dataset or find a similar one that fits the project's requirements.
3. Understand the Project Scope:
   Carefully read through the project description and objectives. Make sure you understand what can be accomplished with the data and what skills you'll be demonstrating.
4. Follow the Project Structure:
   Use the step-by-step guides provided. These steps can be adapted to any of the projects:
   a. Data loading and exploration
   b. Preprocessing and cleaning
   c. Exploratory Data Analysis (EDA)
   d. Feature engineering
   e. Model development and training
   f. Evaluation and optimization
   g. (Optional) Deployment
5. Customize Your Approach:
   Feel free to add your own ideas or techniques to make the project unique. This is your opportunity to showcase your creativity and problem-solving skills.
6. Document Your Process:
   Keep detailed notes of your methodology, challenges faced, and solutions implemented. This documentation will be valuable when presenting your project.
7. (Optional) Deployment:
   While deploying every project is not necessary, consider implementing at least one deployment to demonstrate your full-stack capabilities.

8. Showcase Your Work:
9. Create a GitHub repository for each completed project.
10. Consider presenting your projects on a free portfolio website like datascienceportfol.io.
11. Prepare a brief presentation or write-up summarizing your project, methodology, and key findings.

Remember, the goal is to demonstrate your skills and thought process. Even if you don't complete every aspect of a project, what you learn and how you approach problems are valuable experiences to discuss in interviews.

Project 1: E-commerce Customer Behavior Analysis and Recommendation System

Dataset: Online Retail Dataset (UCI Machine Learning Repository)

Step-by-step guide:

1. Data Loading and Initial Exploration
   a. Import necessary libraries (pandas, numpy, matplotlib, seaborn)
   b. Load the dataset using pd.read_csv()
   c. Display basic information about the dataset using df.info() and df.describe()
2. Data Cleaning and Preprocessing
   a. Handle missing values
   b. Remove duplicates
   c. Convert data types (e.g., convert 'InvoiceDate' to datetime)
   d. Create derived features (e.g., total purchase amount, customer lifetime value)
3. Exploratory Data Analysis (EDA)
   a. Analyze sales trends over time using time series plots
   b. Identify top-selling products and countries

    c. Visualize customer segmentation based on RFM (Recency, Frequency, Monetary) analysis
    d. Investigate correlations between variables

4. Feature Engineering
    a. Create RFM features
    b. Encode categorical variables
    c. Normalize numerical features
5. Customer Segmentation
    a. Implement K-means clustering on RFM features
    b. Visualize clusters using PCA or t-SNE
    c. Interpret and label customer segments
6. Product Recommendation System
    a. Create a user-item matrix
    b. Implement collaborative filtering using Surprise library
    c. Train and evaluate the model using cross-validation
7. Model Evaluation
    a. Use metrics like RMSE and MAE for the recommendation system
    b. Evaluate clustering quality using silhouette score
8. Deployment Options
    a. Flask Web Application:
    a. Create a simple Flask app to serve recommendations
    b. Design a user interface for inputting customer data
    c. Deploy on Heroku or AWS Elastic Beanstalk
9. Streamlit Dashboard:
    a. Develop an interactive dashboard using Streamlit
    b. Include visualizations from EDA and model insights
    c. Deploy on Streamlit Sharing
10. RESTful API with FastAPI:
    a. Build a RESTful API using FastAPI
    b. Implement endpoints for customer segmentation and recommendations
    c. Deploy on Google Cloud Run or Azure Container Instances
11. Jupyter Notebook on Google Colab:

- Create an interactive notebook with widgets
- Share the notebook publicly for easy access and reproducibility

Project 2: Predictive Maintenance for Machines

Data: Machine Predictive Maintenance Classification

Project Description:
Develop a machine learning model to predict potential failures in industrial equipment based on sensor data and operational parameters. This project utilizes a synthetic dataset that mimics real-world predictive maintenance scenarios, focusing on early fault detection to minimize downtime and maintenance costs.

Key Components:

1. Data Exploration and Preprocessing:
    a. Analyze the dataset, which includes features like air temperature, process temperature, rotational speed, torque, and tool wear.
    b. Handle the class imbalance present in the target variable (812 positive cases out of 10,000 samples).
    c. Investigate the relationship between failure types and other features.
2. Feature Engineering:
    a. Create relevant features from the 'Date' column to capture temporal patterns.
    b. Normalize numerical features and encode categorical variables (Type and Failure Type).
3. Model Development:
    a. Implement and compare various classification algorithms, including Random Forest, Gradient Boosting, and Support Vector Machines.

    b. Use techniques like SMOTE or class weighting to address the imbalanced nature of the dataset.
    c. Perform hyperparameter tuning using cross-validation to optimize model performance.

4. Time-based Analysis:
    a. Explore time series aspects of the data, particularly focusing on how tool wear and other parameters change over time.
    b. Implement sliding window techniques to capture temporal dependencies in the data.
5. Anomaly Detection:
    a. Develop anomaly detection models to identify unusual patterns in sensor readings that may indicate impending failures.
    b. Compare traditional anomaly detection methods with machine learning-based approaches.
6. Model Evaluation:
    a. Assess model performance using metrics suitable for imbalanced datasets, such as precision, recall, F1-score, and ROC-AUC.
    b. Analyze feature importance to understand which factors contribute most to failure predictions.
7. Interpretability and Visualization:
    a. Create a dashboard using tools like Plotly or Streamlit to visualize:
    b. Predicted failure probabilities for each machine
    c. Historical trends of key parameters
    d. Feature importance and their impact on failure prediction
    e. Implement model interpretability techniques (e.g., SHAP values) to explain individual predictions.
8. Deployment Strategy:
    a. Design an API that could integrate the predictive model into an existing industrial monitoring system.
    b. Outline a plan for continuous model monitoring and retraining to adapt to changing machine conditions over time.

This project will demonstrate your ability to handle imbalanced datasets, develop predictive models for industrial applications, and create actionable insights from machine learning predictions. It showcases skills in data preprocessing, advanced classification techniques, time series analysis, and data visualization, all crucial for a data scientist in the manufacturing and maintenance sectors.

Project 3: Telecom Customer Churn Prediction

Data:
https://www.kaggle.com/datasets/blastchar/telco-customer-churn/data

Project Description:
Develop a machine learning model to predict customer churn for a telecommunications company. This project will utilize a real-world dataset containing information about 7,043 customers, including their demographics, service subscriptions, account information, and churn status.

Key Components:

1. Data Exploration and Preprocessing:
    a. Analyze the dataset, which includes features like customer demographics, service plans, contract details, and payment information.
    b. Handle the class imbalance in the target variable 'Churn' (26.54% churn rate).
    c. Investigate relationships between various features and customer churn.
2. Feature Engineering:
    a. Create relevant features from existing data, such as total charges to tenure ratio.
    b. Encode categorical variables (e.g., InternetService, Contract type) appropriately.

    c. Normalize numerical features like 'TotalCharges' and 'MonthlyCharges'.

3. Model Development:
    a. Implement and compare various classification algorithms, including Logistic Regression, Decision Trees, Random Forest, and Gradient Boosting.
    b. Use techniques like SMOTE or class weighting to address the moderate class imbalance.
    c. Perform hyperparameter tuning using cross-validation to optimize model performance.
4. Customer Segmentation:
    a. Conduct clustering analysis to identify distinct customer segments based on their characteristics and behaviors.
    b. Analyze churn patterns within different segments to tailor retention strategies.
5. Model Evaluation:
    a. Assess model performance using metrics suitable for imbalanced datasets, such as precision, recall, F1-score, and ROC-AUC.
    b. Analyze feature importance to understand which factors contribute most to churn prediction.
6. Interpretability and Visualization:
    a. Create a dashboard using tools like Plotly or Streamlit to visualize:
    b. Predicted churn probabilities for each customer segment
    c. Key factors influencing churn decisions
    d. Historical trends of churn rates
    e. Implement model interpretability techniques (e.g., SHAP values) to explain individual predictions.
7. Actionable Insights and Recommendations:
    a. Based on the model's predictions and feature importance, develop targeted retention strategies for high-risk customers.
    b. Propose improvements to service offerings or customer experience based on the identified churn factors.

    c. Design a framework for ongoing monitoring of churn risk and the effectiveness of retention initiatives.

8. Deployment Strategy:
    a. Outline a plan for integrating the churn prediction model into the company's existing CRM system.
    b. Propose a workflow for regularly updating the model with new customer data to maintain its accuracy over time.

This project will demonstrate your ability to handle real-world telecom data, develop predictive models for customer behavior, and translate analytical insights into actionable business strategies. It showcases skills in data preprocessing, advanced classification techniques, customer segmentation, and the practical application of machine learning in a business context, all of which are crucial for a data scientist in the telecommunications industry.

Project 4: Price Optimization for E-commerce Platform

Data: https://www.kaggle.com/datasets/arashnic/e-product-pricing

Project Description:
Develop a dynamic pricing model for an online electronics marketplace to maximize revenue and competitiveness. This project utilizes a comprehensive dataset of over 7,000 electronic products with pricing information across various retailers, providing a rich foundation for analyzing market trends and optimizing pricing strategies.

Key Components:

1. Data Exploration and Preprocessing:
    a. Analyze the dataset, which includes features like product name, brand, category, retailer, and pricing information.
    b. Handle missing values and outliers in the pricing data.

    c. Investigate pricing patterns across different product categories and retailers.

2. Feature Engineering:
    a. Create relevant features such as price-to-MSRP ratio, discount percentage, and relative price position within category.
    b. Extract temporal features from the 'date_added' and 'date_updated' columns to capture seasonality and trends.
    c. Develop competitor pricing indices for each product category.
3. Market Analysis:
    a. Conduct time series analysis to identify pricing trends and seasonality across product categories.
    b. Perform regression analysis to understand factors influencing product prices.
    c. Analyze demand elasticity for different product types and price ranges.
4. Dynamic Pricing Model Development:
    a. Implement machine learning algorithms (e.g., Random Forest, XGBoost) to predict optimal pricing based on product attributes and market conditions.
    b. Develop a reinforcement learning model (e.g., Q-learning or Deep Q-Network) to optimize pricing strategies in a simulated market environment.
    c. Incorporate competitor pricing and estimated demand into the model's decision-making process.
5. Pricing Strategy Optimization:
    a. Design different pricing strategies (e.g., cost-plus, value-based, competitor-based) and evaluate their performance using the developed models.
    b. Implement a multi-armed bandit algorithm to dynamically adjust pricing strategies based on real-time performance.
6. A/B Testing Framework:

   a. Design an A/B testing methodology to evaluate the effectiveness of different pricing models in a simulated e-commerce environment.
   b. Develop metrics to measure the impact of pricing changes on revenue, sales volume, and market share.

7. Visualization and Reporting:
   a. Create an interactive dashboard using tools like Plotly or Streamlit to visualize:
   b. Price trends and comparisons across retailers
   c. Model predictions and recommended pricing adjustments
   d. Performance metrics of different pricing strategies
8. Deployment Strategy:
   a. Outline an implementation plan for integrating the pricing model into an e-commerce platform.
   b. Develop an API that could provide real-time pricing recommendations based on current market data.
   c. Propose a system for continuous monitoring and updating of the pricing model to adapt to market changes.

This project will demonstrate your ability to analyze complex e-commerce data, develop sophisticated pricing models, and create data-driven strategies for competitive pricing. It showcases skills in time series analysis, machine learning, reinforcement learning, and the practical application of data science in e-commerce, all of which are highly valuable for a data scientist in the retail and technology sectors.

Project 5: Multi-Class Vehicle Classification for Online Auto Marketplace

Dataset: Stanford Cars Dataset

Project Description:

Develop an advanced image classification model to automatically categorize and tag vehicle images for an online auto marketplace. This project will utilize the Stanford Cars Dataset, offering a diverse range of car makes, models, and years for fine-grained classification.

Key Steps:

1. Data Preprocessing:
    a. Load and explore the dataset using Python libraries like Pandas and Matplotlib
    b. Analyze class distribution and handle any imbalances
    c. Resize and normalize images for consistent input to the model
2. Model Development:
    a. Implement transfer learning using pre-trained CNNs (e.g., ResNet50, VGG16) via TensorFlow or PyTorch
    b. Fine-tune the model on the Stanford Cars Dataset
    c. Experiment with different architectures and hyperparameters to optimize performance
3. Data Augmentation:
    a. Apply techniques like rotation, flipping, and color jittering to increase dataset diversity
    b. Implement custom augmentation strategies tailored to vehicle images
4. Training and Evaluation:
    a. Train the model using appropriate loss functions and optimizers
    b. Monitor training progress and implement early stopping to prevent overfitting
    c. Evaluate model performance using metrics such as accuracy, precision, recall, and F1-score
5. API Development:
    a. Create a Flask or FastAPI-based RESTful API for the trained model
    b. Implement endpoints for image upload, classification, and retrieval of results
    c. Design a simple front-end interface for testing the API

6. Deployment and Integration:
   a. Deploy the API on a cloud platform (e.g., Heroku, AWS, or Google Cloud)
   b. Develop a mock-up of how the model could be integrated into an online auto marketplace