List of Projects for DSA 5105 - 2410

TOC

README

Welcome to the proposed list of student projects for this semester. This document outlines various real-life projects sourced from industry partners, providing an opportunity to apply your skills to practical problems. Here are some key points to keep in mind:

1. **Project List:** The list of projects is fixed, though we might add new ones over time. Some content may be adjusted for accuracy or clarity as needed.

2. **Data Sourcing:** Most projects require you to source your own data or generate synthetic data. While I will try to provide some additional datasets to assist you, the primary goal is for you to learn how to source and manage data independently. Each project includes suggestions for potential datasets to use.

3. **Optional Questions**: You are not required to complete all optional questions within a project. You can choose which optional questions to tackle and how many to address. Completing optional questions can help improve your grade by demonstrating a deeper understanding and additional effort.

4. **Industry Projects**: The projects are based on real-life scenarios proposed by various companies. This semester, the business questions are clearly stated to provide better guidance and focus.

5. **Grading**: Your grade will be based on the completion and quality of your work, with additional credit given for addressing optional questions. Projects are evaluated on their thoroughness, creativity, and how well you can apply theoretical knowledge to practical situations.

Feel free to reach out if you have any questions or need further guidance. Good luck, and I look forward to seeing your innovative solutions and analyses!

Automated ESG Data Extraction and Performance Analysis

Project Context:

Environmental, Social, and Governance (ESG) reporting has become increasingly crucial for businesses and investors. However, the process of extracting relevant ESG information from unstructured reports is often time-consuming and labor-intensive. Additionally, evaluating ESG performance across companies and industries remains challenging due to the lack of standardized, easily comparable data.

This project aims to develop an innovative system that automates the extraction of ESG information from unstructured reports and provides a comprehensive analysis of ESG performance within selected industries. By leveraging advanced natural language processing (NLP) techniques and data analysis, the project seeks to streamline the ESG data extraction process, improve data quality, and offer valuable insights into corporate sustainability practices.

Scenario:
You are a team of 8 data scientists joining a cross-functional team of ESG analysts, financial experts, and sustainability professionals working to enhance ESG data extraction and analysis capabilities.

The team asks for your help with the following business questions:

Collaborative Phase (All Team Members):

Data Collection and Preprocessing:
1. Identify and collect a diverse set of ESG reports from companies within a selected industry (e.g., healthcare, technology, finance).
2. Gather relevant ESG frameworks and standards, including the core ESG metrics suggested by the Singapore Exchange (SGX).
3. Create a labeled dataset for training and testing the NLP models by manually annotating a subset of the collected reports.
4. Preprocess the collected reports, including text cleaning, tokenization, and formatting for NLP tasks.
5. Document all data sources, preprocessing steps, and annotation guidelines.

After completing the collaborative phase, split into two subgroups to address the following business questions:

Subgroup A: Algorithm Development for ESG Information Extraction

1. How can we develop an effective NLP algorithm to automatically extract targeted ESG information from unstructured reports?
   - Explore and compare different NLP techniques (e.g., named entity recognition, topic modeling, sentiment analysis) for ESG data extraction.
   - Develop a multi-stage pipeline for identifying, categorizing, and structuring ESG-related data.
   - Implement techniques to handle industry-specific terminology and context.

2. How can we ensure the accuracy and reliability of the extracted ESG information?
   - Design and implement a validation system to cross-check extracted information against known ESG frameworks and standards.
   - Develop confidence scoring mechanisms for extracted data points.
   - Create a human-in-the-loop system for reviewing and correcting algorithm outputs.

3. How can we make the algorithm adaptable to evolving ESG reporting standards and company-specific reporting styles?
   - Implement transfer learning techniques to adapt the model to new industries or reporting frameworks.
   - Develop a system for continuous learning and improvement based on user feedback and new data.

Subgroup B: ESG Performance Evaluation and Analysis

1. How can we effectively evaluate ESG performance using the structured dataset obtained from the algorithm?
   - Develop a scoring system aligned with the core ESG metrics suggested by the Singapore Exchange (SGX).
   - Create industry-specific benchmarks for ESG performance.
   - Implement techniques to handle missing data and ensure fair comparisons across companies.

2. What are the key trends and patterns in ESG performance within the selected industry?
   - Conduct a comprehensive analysis of ESG scores across different companies and sub-sectors.
   - Identify common strengths and weaknesses in ESG practices within the industry.
   - Analyze the correlation between ESG performance and financial metrics.

3. How can we present ESG performance insights in a clear and actionable manner for various stakeholders?
   - Develop interactive visualizations and dashboards to showcase ESG performance trends.
   - Create customizable reports tailored to different stakeholder needs (e.g., investors, regulators, company management).

- Implement a system for generating automated ESG performance summaries and recommendations.

Optional Bonus Questions (for higher grades) - answer any or all to boost your score:

Subgroup A: Algorithm Development for ESG Information Extraction

1. How can we incorporate multi-lingual capabilities into the ESG information extraction algorithm?
   - Develop techniques for extracting ESG information from reports in multiple languages.
   - Implement a system for aligning ESG concepts across different languages and reporting cultures.

2. Can we develop a system to detect potential "greenwashing" or misleading ESG claims in company reports?
   - Implement advanced NLP techniques to identify discrepancies between quantitative data and qualitative statements.
   - Develop a model to compare company claims against industry benchmarks and external ESG assessments.

3. How can we extend the algorithm to extract ESG information from alternative data sources, such as news articles, social media, or regulatory filings?
   - Develop techniques for integrating and cross-validating ESG information from multiple source types.
   - Implement a system for real-time ESG monitoring and alert generation based on external data sources.

Subgroup B: ESG Performance Evaluation and Analysis

1. Can we develop a predictive model to forecast future ESG performance based on historical data and external factors?
   - Implement machine learning techniques to predict ESG scores and identify potential areas of improvement.
   - Incorporate external factors (e.g., regulatory changes, market trends) into the predictive model.

2. How can we quantify the financial impact of ESG performance on company valuation and risk profile?
   - Develop a model to analyze the relationship between ESG scores and financial metrics (e.g., stock price, volatility, cost of capital).
   - Create a risk assessment framework that incorporates ESG factors alongside traditional financial risk measures.

3. Can we implement a system for generating personalized ESG investment strategies based on user preferences and risk tolerance?
   - Develop an algorithm for portfolio optimization that balances financial returns with ESG performance.
   - Create a user interface for investors to customize their ESG priorities and receive tailored investment recommendations.

Collaborative Deliverable:

As a team, synthesize your findings and recommendations into a comprehensive strategy for enhancing ESG data extraction and performance analysis.

Your final deliverables should include:

1. A 10-minute video presentation for CxO senior stakeholders.
2. A slide deck (8-12 slides) supporting the video presentation.
3. [Optional] An interactive dashboard showcasing your results and key ESG metrics.

A Git repository containing:

1. Production-ready Python code:
   - Modular Python scripts for data preprocessing, NLP model development, ESG scoring, and analysis
   - A main.py file that orchestrates the entire data pipeline and analysis process
   - A config.py file for all configuration parameters
   - A utils.py file for utility functions used across multiple scripts
   - SQL scripts for data storage and retrieval (stored in a 'sql' directory)
   - A simple API (using Flask or FastAPI) to serve model predictions and key insights
   - A requirements.txt file listing all dependencies
   - Comprehensive docstrings for all functions, classes, and modules
   - [Optional] Unit tests for all critical functions
   - [Optional] A logging system for tracking the execution of the code

2. Docker-related files:
   - A Dockerfile to containerize the application
   - [Optional] A docker-compose.yml file if multiple services are required

3. Documentation:
   - A README.md file with:
     - Project overview
     - Instructions for setting up the environment and running the code
     - Description of the repository structure
     - Data sources and any necessary data preparation steps
     - Instructions for building and running the Docker container(s)
     - API documentation (endpoints, request/response formats)

- [Optional] API documentation using Swagger/OpenAPI specification
- A data dictionary explaining all variables used in the analysis

Optional Deliverables (for higher grades) - You don't need to do them all:

1. Advanced NLP Techniques:
   - Implement a transformer-based model (e.g., BERT, GPT) fine-tuned for ESG information extraction.
   - Develop a zero-shot learning approach for adapting to new ESG categories without additional training data.
   - Create an ensemble model combining multiple NLP techniques for improved accuracy.

2. Enhanced Visualization:
   - Develop an interactive, real-time dashboard for ESG performance monitoring using tools like D3.js or Plotly.
   - Create a network visualization of ESG relationships and impacts using tools like NetworkX and Cytoscape.
   - Implement a geospatial visualization of ESG performance across different regions and countries.

3. Advanced Software Engineering:
   - Implement a microservices architecture for different components of the system (e.g., data ingestion, NLP processing, scoring, API).
   - Set up a CI/CD pipeline using tools like Jenkins or GitLab CI.
   - Implement automated data quality checks and model performance monitoring.

4. Big Data Processing:
   - Set up a data pipeline using Apache Airflow for orchestrating the ESG data extraction and analysis process.
   - Implement distributed processing capabilities using Apache Spark for handling large volumes of ESG reports.
   - Design a data lake architecture for storing and processing structured and unstructured ESG data.

5. Advanced API and Web Application:
   - Develop a full-fledged web application with user authentication and role-based access control for ESG analysts and investors.
   - Implement real-time updates using WebSockets for live ESG performance monitoring.
   - Create a mobile app version of the ESG dashboard using a framework like React Native or Flutter.

6. Extended Documentation and Testing:
   - Develop comprehensive API documentation using tools like Swagger UI.
   - Implement integration tests and end-to-end tests in addition to unit tests.

- Create a detailed technical design document outlining the system architecture and data flow for ESG information extraction and analysis.

7. Advanced Containerization and Deployment:
   - Implement a Kubernetes deployment for scalability and easier management of the ESG analysis system.
   - Set up monitoring and logging using tools like Prometheus and ELK stack for tracking system performance and ESG data processing.
   - Implement blue-green deployment strategy for zero-downtime updates to the ESG analysis platform.

8. Ethics and Privacy Enhancements:
   - Develop a comprehensive data anonymization and privacy protection strategy for handling sensitive ESG information.
   - Implement federated learning techniques for collaborating on ESG models without sharing raw company data.
   - Create an ethics review process for new ESG analysis features and methodologies.

9. Business Intelligence and Strategy:
   - Develop a set of KPIs for tracking the effectiveness and impact of the ESG analysis system.
   - Conduct a cost-benefit analysis of implementing the automated ESG extraction and analysis system.
   - Perform a competitive analysis comparing the proposed system to other ESG data providers and rating agencies.

Instructions:

1. Each subgroup should create a 5-7 slide mini-presentation addressing their specific business questions.
2. The full group should then collaborate to create the main 8-12 slide presentation that synthesizes the key findings and recommendations from both subgroups.
3. Record a 10-minute video presentation suitable for CxO senior stakeholders, using the main slide deck as visual support.
4. Create a Wiki for your project.
5. Develop production-ready Python code, SQL scripts, and an API. Set up a Git repository with all code, documentation, and non-sensitive data files. Ensure it's well-organized, follows PEP 8 style guidelines, and includes clear instructions for use.
6. Create a Dockerfile and docker-compose.yml (if necessary) to containerize your application.

Grading:

Your assignment will be judged according to:

- The analytical approach and clarity of your graphs, tables, visualizations,

- The data decisions you made and reproducibility of the analysis,
- Strength of recommendations, prioritizations, and rationale behind those,
- The narrative of your presentation and ability to effectively communicate to non-technical stakeholders,
- Quality and organization of the production-ready Python code,
- Effective use of SQL for data storage and retrieval,
- Design and implementation of the API,
- Proper containerization of the application using Docker,
- Comprehensive documentation and testing of the code,
- The effectiveness of your group collaboration and integration of different aspects of the analysis.
- Implementation and quality of optional deliverables (for higher grades)

About Data:

For this project, you will need to work with ESG reports and related data. Here are some potential data sources:

1. Corporate Sustainability Reports:
   - Description: Annual sustainability or ESG reports published by companies.
   - Sources: Company websites, sustainability reporting databases

2. ESG Ratings and Scores:
   - Description: ESG ratings provided by various agencies.
   - Sources: MSCI ESG Ratings, Sustainalytics, Bloomberg ESG Data

3. Singapore Exchange (SGX) ESG Metrics:
   - Description: Core ESG metrics suggested by SGX for listed companies.
   - Source: SGX website and regulatory filings

4. Global Reporting Initiative (GRI) Standards:
   - Description: Widely used sustainability reporting framework.
   - Source: GRI website

5. CDP (formerly Carbon Disclosure Project) Data:
   - Description: Environmental impact and climate change-related disclosures.
   - Source: CDP website

For creating a labeled dataset for NLP model training:

Design an annotation scheme based on key ESG categories and metrics. Manually annotate a subset of collected reports, ensuring proper documentation of the annotation process and guidelines. If possible, have multiple annotators work on the same documents to calculate inter-annotator agreement and ensure consistency.

Instructions for Data Use:

1. Choose a specific industry or sector for your analysis to ensure comparability of ESG practices.
2. Collect ESG reports and related data for a representative sample of companies within the chosen industry.
3. Ensure proper data cleaning, preprocessing, and structuring for both the NLP tasks and performance analysis.
4. Clearly document any assumptions made or data transformations performed in your final report.

Additional Considerations:

- Pay special attention to the consistency and comparability of ESG data across different companies and reporting years.
- When working with ESG reports, be mindful of potential biases in company disclosures and the need for external validation.
- Consider the evolving nature of ESG reporting standards and ensure your system can adapt to changes in metrics and priorities.

Bonus / Extra Credit Opportunity:

After completing the main project, teams can earn extra points by creating a Minimum Viable Product (MVP) for an interactive ESG analysis platform:

1. Design and implement a basic web interface for ESG data exploration and company comparison.
2. Incorporate at least one AI-powered feature (e.g., natural language query system for ESG data, automated ESG report summarization).
3. Create a small-scale prototype that demonstrates how the interactive system could enhance ESG analysis and decision-making.
4. Provide a brief report (2-3 pages) outlining:
   - The key features of your MVP
   - How it addresses common challenges in ESG data analysis
   - Potential benefits for investors, analysts, and companies
   - Implementation challenges and how you addressed them
   - Ethical considerations and data privacy aspects of your interactive system
5. Include a demo video of your MVP in action.

Greenhouse Gas Emissions Calculator Application for Singapore's Office Buildings

Project Context:

As climate change concerns escalate, organizations are increasingly focused on measuring and reducing their greenhouse gas (GHG) emissions. However, many companies, especially those without specialized knowledge in carbon auditing, face challenges in accurately quantifying their GHG emissions. This is particularly true for office buildings in Singapore, where energy consumption and associated emissions can be significant.

This project aims to develop a user-friendly, standalone application for GHG emissions calculation tailored for Singapore's office buildings. The tool will enable companies to easily quantify their annual GHG emissions, track performance over time, and identify opportunities for energy savings and carbon reduction. By providing two calculation methods and incorporating local and international emission factors, the project seeks to make GHG emissions tracking accessible to a wide range of users while maintaining accuracy and reliability.

Scenario:
You are a team of 8 data scientists joining a cross-functional team of environmental engineers, software developers, and sustainability experts working to develop and implement the GHG emissions calculator application for Singapore's office buildings.

The team asks for your help with the following business questions:

Collaborative Phase (All Team Members):

Data Collection and Preprocessing:

1. Identify and collect relevant datasets for GHG emissions calculations, including:
   - Energy consumption data for typical office buildings in Singapore
   - Transportation data for estimating commuting emissions
   - Water and waste management data
   - Local and international emission factors
2. Gather information on Singapore's building standards and energy efficiency regulations.
3. Collect sample activity data from a diverse set of office buildings for testing and validation.
4. Preprocess and clean the collected datasets, ensuring consistency and compatibility.
5. Document all data sources, preprocessing steps, and assumptions made.

After completing the collaborative phase, split into two subgroups to address the following business questions:

Subgroup A: GHG Emissions Calculation Methodology and Implementation

1. How can we develop accurate and reliable methods for GHG emissions calculations tailored to Singapore's office buildings?
   - Design and implement the "activity data-based calculations" method, incorporating local and international emission factors.
   - Develop the "estimation calculations" method for scenarios where detailed activity data is unavailable.
   - Create a system for automatic selection and updating of appropriate emission factors.

2. How can we ensure the accuracy and reliability of the emissions calculations across different types of office buildings?
   - Develop a validation system to cross-check calculated emissions against known benchmarks and industry standards.
   - Implement sensitivity analysis to identify key factors influencing emissions calculations.
   - Design a system for continuous improvement of calculation methodologies based on user feedback and new data.

3. How can we incorporate spatial analysis to estimate average commuting emissions for employees?
   - Develop a model to estimate commuting emissions based on office location and transportation infrastructure.
   - Implement geospatial analysis techniques to account for variations in commuting patterns across different areas of Singapore.
   - Create a system for updating commuting emissions estimates based on changes in transportation infrastructure or work patterns.

Subgroup B: User Interface, Data Visualization, and Benchmarking

1. How can we design a user-friendly interface that allows non-experts to easily input data and understand their GHG emissions?
   - Develop an intuitive desktop application interface for data input and results display.
   - Create interactive guides and tooltips to assist users in understanding and inputting required data.
   - Implement data validation and error checking to ensure data quality and completeness.

2. What are the most effective ways to visualize GHG emissions data for different stakeholders?
   - Design interactive dashboards within the application to display emissions data, trends, and comparisons.
   - Develop customizable reports for different user needs (e.g., management summaries, detailed breakdowns for sustainability teams).
   - Create visualizations that highlight the impact of different emissions sources and potential areas for reduction.

3. How can we implement an effective benchmarking system to compare emissions across different office buildings?
   - Develop a methodology for calculating and comparing emissions intensity across buildings.
   - Create a system for anonymized peer comparison and industry benchmarking.
   - Implement features to track and visualize progress towards emissions reduction goals.

Optional Bonus Questions (for higher grades) - answer any or all to boost your score:

Subgroup A: GHG Emissions Calculation Methodology and Implementation

1. How can we incorporate machine learning techniques to improve the accuracy of emissions estimations?
   - Develop ML models to predict emissions based on building characteristics and partial activity data.
   - Implement anomaly detection algorithms to identify potential errors in user-input data or unusual emissions patterns.

2. Can we develop a system to automatically estimate the uncertainty in emissions calculations?
   - Implement Monte Carlo simulation techniques to quantify uncertainty in emissions estimates.
   - Develop a methodology for communicating uncertainty to users in an understandable way.

3. How can we extend the calculator to account for scope 3 emissions related to office building operations?
   - Develop methodologies for estimating emissions from employee business travel, procurement, and waste disposal.
   - Create a system for tracking and allocating emissions from shared spaces or multi-tenant buildings.

Subgroup B: User Interface, Data Visualization, and Benchmarking

1. Can we implement a feature for generating customized emission reduction recommendations based on a building's specific characteristics and emissions profile?
   - Develop an AI-powered recommendation engine that suggests targeted emission reduction strategies.
   - Create a system for estimating the potential impact and cost-effectiveness of different reduction measures.

2. How can we incorporate data import features for various file formats and building management systems?
   - Develop integrations with common file formats (CSV, Excel) and building management system data exports.
   - Implement data validation and error handling for imported data.

3. Can we create a feature for scenario modeling and forecasting future emissions?

- Develop a system for users to model different scenarios (e.g., energy efficiency upgrades, occupancy changes).
- Implement forecasting algorithms to project future emissions based on historical data and planned changes.

Collaborative Deliverable:

As a team, synthesize your findings and recommendations into a comprehensive strategy for implementing the GHG emissions calculator application for Singapore's office buildings.

Your final deliverables should include:

1. A 10-minute video presentation for senior stakeholders in Singapore's building and environmental sectors.
2. A slide deck (8-12 slides) supporting the video presentation.
3. [Optional] A prototype or demo version of the GHG emissions calculator application.

A Git repository containing:

1. Production-ready Python code:
   - Modular Python scripts for emissions calculations, data processing, and analysis
   - A main.py file that orchestrates the entire calculation and analysis process
   - A config.py file for all configuration parameters
   - A utils.py file for utility functions used across multiple scripts
   - A desktop application (using a framework like PyQt or Tkinter) to serve as the user interface for the emissions calculator
   - A requirements.txt file listing all dependencies
   - Comprehensive docstrings for all functions, classes, and modules
   - [Optional] Unit tests for all critical functions
   - [Optional] A logging system for tracking the execution of the code

2. Packaging and distribution files:
   - Setup files for creating an executable or installable package of the application
   - [Optional] Scripts for automating the build and distribution process

3. Documentation:
   - A README.md file with:
     - Project overview
     - Instructions for setting up the development environment and running the code
     - Description of the repository structure
     - Data sources and any necessary data preparation steps
     - Instructions for building and distributing the application
   - A user manual for the GHG emissions calculator application
   - A data dictionary explaining all variables used in the emissions calculations

Optional Deliverables (for higher grades) - You don't need to do them all:

1. Advanced Emissions Modeling:
   - Implement a machine learning model for predicting future emissions based on historical data and building characteristics.
   - Develop a system for scenario analysis to help users understand the potential impact of different emission reduction strategies.
   - Create a module for lifecycle emissions analysis of building materials and equipment.

2. Enhanced Visualization:
   - Develop interactive, customizable charts and graphs for emissions data visualization.
   - Create a 3D visualization of building emissions using a graphics library like OpenGL or DirectX.
   - Implement a geospatial visualization of emissions across different office buildings in Singapore.

3. Advanced Software Engineering:
   - Implement a plugin architecture to allow for easy extension of the application's capabilities.
   - Set up a CI/CD pipeline using tools like Jenkins or GitLab CI for automated testing and building of the application.
   - Implement automated data quality checks and model performance monitoring.

4. Data Management and Processing:
   - Develop a local database system for storing and managing emissions data over time.
   - Implement data import/export features for various file formats and databases.
   - Design an efficient data structure for handling large volumes of emissions-related data.

5. Advanced User Interface:
   - Develop a customizable dashboard system allowing users to create personalized views of their emissions data.
   - Implement a natural language interface for querying emissions data and generating reports.
   - Create an embedded tutorial system to guide new users through the application's features.

6. Extended Documentation and Testing:
   - Develop comprehensive API documentation for potential future integrations.
   - Implement integration tests and end-to-end tests in addition to unit tests.
   - Create a detailed technical design document outlining the system architecture and data flow for emissions calculations and analysis.

7. Optimization and Performance:
   - Implement multi-threading or multi-processing to improve calculation speed for large datasets.
   - Optimize memory usage for handling data from multiple buildings or long time periods.
   - Develop a caching system to improve application responsiveness.

8. Ethics and Privacy Enhancements:
   - Develop a comprehensive data anonymization and privacy protection strategy for handling sensitive building energy data.
   - Implement local encryption for stored data to ensure confidentiality.
   - Create an ethics review process for new features and methodologies related to emissions tracking and reduction recommendations.

9. Business Intelligence and Strategy:
   - Develop a set of KPIs for tracking the effectiveness and impact of emissions reduction strategies.
   - Conduct a cost-benefit analysis of implementing various emission reduction strategies across different building types.
   - Perform a comparative analysis of Singapore's office building emissions against international benchmarks.

Instructions:

1. Each subgroup should create a 5-7 slide mini-presentation addressing their specific business questions.
2. The full group should then collaborate to create the main 8-12 slide presentation that synthesizes the key findings and recommendations from both subgroups.
3. Record a 10-minute video presentation suitable for senior stakeholders in Singapore's building and environmental sectors, using the main slide deck as visual support.
4. Create a Wiki for your project.
5. Develop production-ready Python code and a standalone application. Set up a Git repository with all code, documentation, and non-sensitive data files. Ensure it's well-organized, follows PEP 8 style guidelines, and includes clear instructions for use.
6. Create necessary files for packaging and distributing the application.