Predictive Data Analytics Report

1. Data description and research question

The **unified country-year panel (from 1990 to 2020)** dataset combines indicators of **agro-food activities**, **demographic characteristics** and the major outcome variable (i.e. **total_emission**). Following pre-processing, the data comprise of some **1,200 observations** and **31 to 35 predictors** for **197 countries** over a time span of **30 years**. The input parameters are classified into **three main groups**:

- **Emission drivers in the agro-food sector** — itemised in terms of **crop residues, rice paddy production, fertiliser application, manure treatment, food transport, processing and storing**;
- **Population & urbanisation indicators** – which included **total, rural and urban population numbers**, and derived variables such as the **urban population ratio (UrbanPop_ratio)**;
- **Environmental covariates** – in particular the **average temperature** and other **climatic factors**.

Taken together, these three **industrial**, **agricultural**, and **demographic dimensions** offer a uniquely varied view of how **global food systems influence the climate**. This makes the data particularly appropriate for the module's focus on **complex, "raw" multivariate data**. In particular, **data quality issues** consisted of up to **30% missing values** in some of the fields, as well as **"streaks of phantom zeros"** in country-level time series, as explained in the **cleaning log**.

Research question

Can total CO_2 emissions of a country be accurately predicted from its agro-food activities and population characteristics?

2. Data preparation and cleaning

Uncovering Structure in Data through Visualization

We started the analysis by using the **country-year 1990–2020 dataset** and, before performing the imputation steps, we created a **categorical Region variable**. This category followed a classification of countries sharing similar **geographic location, climate conditions, and economic levels**. Adapting regional structure served **three main purposes**:

- **Improved visual analysis** – it allowed our ad-hoc plots to portray true regional emissions (e.g. Africa vs Europe).
- **Local context imputation** – by aggregating countries regionally, we ensured missing values were imputed in a context-aware manner, using neighbouring environmental and socio-economic similarities.
- **Model efficiency and interpretability** – region labels were used as **low-cardinality categorical features**, which facilitated model interpretation and kept computational burden minimal.

Once **regional clusters** were determined, the **four-stage cleaning process** was applied:

Missing Value Audit & Phantom Zero Indicators

A preliminary isnull() check indicated that as much as **30% of values** were missing in some agro-food variables. Pattern scans revealed **long sequences of zeros**, suggesting non-reporting rather than true zero values. A **confidence flag** was assigned to each entry to differentiate imputed from original values.

Two-Stage Imputation

- **Stage 1: Interpolation** – A first step applied **linear interpolation** to fill brief missing periods in a country's timeline, preserving temporal continuity.
- **Stage 2: KNN Imputation** – Remaining missing or flagged values were filled using **K-Nearest Neighbours (k=5)** within a **standardised feature space** that included the Region variable. **KNN was preferred** over global

mean/median to preserve local relationships, and over k-means due to the data's non-convex, high-dimensional nature.

Zero-Variance and Missing Features Pruning

Columns with **zero standard deviation** (e.g., Savanna fires in some regions) were **removed** to avoid incorporating noise into the model.

Feature Scaling

All numerical variables were **standardised using StandardScaler (mean = 0, std = 1)**. This step was essential as both **distance-based algorithms (e.g., KNN, SVM)** and **deep learning models** require features to be on comparable scales. Otherwise, large-valued features (e.g., population) would overpower small-valued ones (e.g., fertiliser use).

3. Exploratory data analysis

Exploratory data analysis (EDA) began with **statistical and graphical profiling** of the cleaned dataset, which included around **1,200 country-year entries**.

- A **global line chart time series** showed that **agro-food related CO_2 emissions** steadily increased from approximately **11 million kt in 1990 to more than 17 million kt by 2020**. This **monotonic rise** indicated a key pattern that any predictive model must capture.
- **Regional bar plots for 2020** demonstrated that **Asia and Africa dominate current emissions**, while **Europe's emissions have plateaued**, emphasizing the importance of accounting for **regional effects**.
- **Distribution visualisations** (e.g., **boxplots** and **log-scale histograms**) revealed a **heavily right-skewed distribution with high variance**, indicating the influence of a **small number of high-emitting countries** and the need for **outlier-resilient models**.

- A **correlation heatmap** identified **strong multicollinearity** among variables such as **fertiliser use, manure management, food transport, and total population**, all of which were **highly correlated (≥ 0.8)** with **total_emission**. In contrast, **temperature had almost no correlation (r ≈ 0.02)**. This suggested **redundancy among predictors** and motivated **dimensionality reduction**.

Why We Ran PCA as Part of EDA

Principal Component Analysis (PCA) was conducted during the EDA stage on **standardised numerical features**, enabling immediate use of **principal components for visualisation and clustering**. The following key benefits emerged:

1. **Reduced noise and multicollinearity**
    a. The original **35 interrelated variables** were simplified. A **scree plot** showed that the **first 5–6 components captured 75–80% of total variance**, reducing computational costs for algorithms like **SVM and k-means**.

2. **Improved 2D visualisation**
    a. A **PC1 vs. PC2 biplot** presented a clear view of countries and feature contributions. **Fertiliser, manure, and processing loaded onto PC1**, while **urbanisation indicators were prominent on PC2**—a level of clarity impossible in the full 35-dimensional space.
3. **Identification of natural clusters**
    a. The **elbow method suggested k = 4**, and **k-means clustering** on principal components produced distinct **high-, mid-, and low-emission groups** that aligned with the predefined **Region variable**.
4. **Informed feature selection**
    a. **Component loading plots** revealed that features like **food processing, fertiliser production, and population size** dominated PC1, while variables such as **savanna and humid-forest fires** had

minimal effect and could be **removed or down-weighted**. This directly guided the **Random Forest feature set**.

5. **Early evaluation of dimensionality trade-offs**
    a. Since PCA was embedded in EDA, it allowed us to compare model performance using **raw features vs. PCA features**. Results showed **Random Forest experienced only a minor drop (~6% RMSE, 0.0005 R^2)**, while **SVM performance declined significantly (~38% RMSE)**. This insight helped determine when to use compressed features (for deep learning) and when to retain original features (for tree-based models).

In summary, including **PCA during the EDA phase** transformed a complex, multicollinear dataset into a **smaller, orthogonal feature space** that:

- **Enabled intuitive visualisations and clustering**,
- **Exposed dominant sources of variation**, and
- **Provided a flexible input space** for downstream machine and deep learning models.

This strategic integration was **instrumental in refining the modelling strategy**, as it allowed **direct comparisons between raw and reduced feature sets** early in the process.

4. Machine learning prediction

Support-Vector Machine

The cleaned dataset was split *by country* into 60 % training, 20 % validation and 20 % test folds preserving comparability with the tree experiment. Because SVMs are highly scale-sensitive, every numeric predictor was first standardised to zero mean and unit variance. A grid search over the regularisation parameter C (0.1 – 100) and kernel width γ (1/p – 10/p) was conducted with five-fold cross-validation on the training data; the optimum lay near $C = 10$ and $\gamma \approx 2/p$. The resulting model used just 194 support vectors—about 4.6 % of the 1 200 records—indicating a well-regularised fit.

Performance-On the unseen test partition it posted **R^2 = 0.9928 and RMSE = 18 563 kt**, while the training R^2 was 0.9979, leaving a mere 0.005 generalisation gap and demonstrating that the margin-maximisation framework had contained over-fitting ML_RANDOM FOREST.

Interpretation- Notably, when the predictor matrix was compressed to six principal components beforehand, test RMSE worsened to 25 700 kt and R^2 dropped to 0.9863, underscoring the model's dependence on a rich, unreduced feature space

Limitations.

SVM is highly sensitive to scaling (hence the mandatory standardisation step) and to the richness of the feature space; any aggressive dimensionality reduction degrades its precision. Kernel outputs are also harder to translate into concrete policy levers than tree-based importance scores. Nevertheless, the raw-feature SVM provides a robust, almost-as-accurate alternative to Random Forest, confirming that the agro-food and demographic predictors carry enough nonlinear signal to support multiple modelling paradigms.

5. Deep learning prediction

Autoencoder+MLP

The autoencoder + MLP pipeline was introduced to learn a data-driven 32-dimensional latent space that removes multicollinearity and noise more flexibly than manual PCA, yet still captures the nonlinear agro-food/population patterns driving emissions. Feeding this compact code into a tiny regressor trimmed model size, yielded $R^2 \approx 0.96$, and produced a visually interpretable latent map of high- vs. low-emitting countries that can be reused in future tasks.

Implementation-To see whether a learned latent representation could rival manual PCA, a deep autoencoder first compressed the ~30 predictors into a 32-dimensional bottleneck (encoder: Dense-128 → Dropout 0.3 → Dense-64 → Dropout 0.3 → Dense-32). After convergence, the decoder was discarded and the

32-dim code fed into a small regressor (Dense-32 → linear 1). The same Adam optimiser, batch size and Early-Stopping regime as the ANN were used, with dropout providing extra regularisation. Latent quality was inspected with a t-SNE plot that showed distinct clusters separating high-emission industrial nations from low-emission agrarian ones DL_ANNDL_ANN.

Performance- On the held-out test set the hybrid model achieved RMSE ≈ 23 000–24 000 kt and $R^2 = 0.9634$, producing a train-test R^2 gap of 0.011—slightly wider than the plain ANN but still within acceptable bounds. Points in the actual-vs-predicted scatter lay close to the diagonal, and the residual distribution peaked sharply at zero, although long tails revealed a bit more variance for country-level outliers. While the autoencoder trails the Random Forest and ANN numerically, it adds value by delivering a compact, interpretable latent space that could be reused in downstream tasks or visual analytics.

Limitations

The DL_Autoencoder+MLP architecture, while offering a compressed and interpretable latent representation of the data, comes with increased model complexity and tuning demands. The multi-stage design introduces more trainable parameters and requires careful balancing of dropout, bottleneck size, and early stopping to avoid overfitting. Although it generalizes well, its test R^2 (≈0.9634) lags behind simpler alternatives like Random Forest and even the basic MLP, indicating that the gain in latent structure comes at the cost of reduced predictive accuracy. Additionally, the slightly wider train-test R^2 gap and residual scatter for high-emission countries suggest some variance leakage, and the added computational footprint—though manageable—makes the model less lightweight than single-pass regressors

6) ML Performance evaluation and comparison of methods (Random Forest, Decision Tree, XGBoost, SVM)

- **Accuracy**
  Random Forest: $R^2 = 0.9958$, RMSE = 13,419 kt

XGBoost: $R^2 = 0.9916$, RMSE = 17,028 kt
SVM: $R^2 = 0.9928$, RMSE = 18,563 kt
Decision Tree: $R^2 = 0.9643$, RMSE = 31,082 kt — Random Forest leads in precision, with XGBoost and SVM close behind; Decision Tree underperforms on unseen data due to overfitting.

- **Generalisation**
  Train-test R^2 gaps are minimal for Random Forest (0.0024), XGBoost (~0.0047), and SVM (0.0050), confirming solid generalisation. Decision Tree shows a wider performance drop-off, highlighting its sensitivity to training variance and lack of regularisation mechanisms. SVM's performance dips more noticeably when dimensionality is reduced (e.g., PCA), while tree-based models maintain stability.
- **Interpretability & Scale**
  Decision Tree is most interpretable due to its simple, rule-based structure but fails to capture higher-order interactions. Random Forest enhances interpretability through aggregated feature importance and handles multicollinearity well. XGBoost offers faster training and competitive accuracy, though its iterative boosting structure slightly compromises transparency. SVM performs well on raw features but becomes harder to interpret and more sensitive to feature scaling, especially with large datasets. In terms of scalability, Random Forest and XGBoost are both efficient, with XGBoost offering greater memory efficiency, while SVM lags on high-dimensional data.
- **Verdict**
  All four models support the research hypothesis, with Random Forest delivering the strongest accuracy, robustness, and interpretability—making it the top choice for policy-relevant forecasting. XGBoost is a high-performing alternative, particularly suited for fast, large-scale modeling. SVM is mathematically robust and nearly matches tree ensembles in accuracy but requires more careful tuning and lacks actionable transparency. Decision Tree provides pedagogical value and interpretability but falls short in predictive power on complex, high-dimensional data.

DL Performance evaluation and comparison of methods (MLP, Autoencoder+MLP, Deep Neural Network, LSTM)

- **Accuracy**
  MLP: $R^2 = 0.9894$ | RMSE ≈ 23,505 kt
  Autoencoder + MLP: $R^2 = 0.9634$ | RMSE ≈ 23,000–24,000 kt
  Deep Neural Network (DNN): $R^2 = 0.9667$ | RMSE ≈ 40,870 kt
  LSTM with Embeddings: $R^2 = 0.9835$ | RMSE ≈ 40,115 kt
  -While the MLP achieves the highest predictive accuracy among the tested architectures, the LSTM offers strong results for temporal patterns, and the autoencoder introduces valuable latent insights.
- **Generalisation**
  The MLP maintains a narrow train-test R^2 gap (≈ 0.006), confirming robust generalization. Autoencoder + MLP sees a slightly wider drop (≈ 0.011), reflecting its deeper architecture. LSTM generalises exceptionally well, showing minimal overfitting despite handling sequential data. The DNN, however, presents signs of mild overfit, as indicated by its higher test RMSE despite a respectable R^2.
- **Latent Structure & Learning Behaviour**
  The autoencoder captures a compact, 32-dimensional representation of high- and low-emission countries, visualised via t-SNE. This adds value for interpretability and potential multi-task extensions. LSTM stands out for temporal awareness, capturing lagged interactions across years, which static models miss. The plain MLP and DNN models, while less interpretable, excel at direct regression tasks due to their straightforward feed-forward structure.
- **Compute & Scaling**
  All models were trained on CPUs without GPU acceleration. The MLP and autoencoder scale well for this data size. LSTM and DNN, while deeper, did not encounter performance bottlenecks on ≈1,200 samples but may require GPU support for future extensions or larger-scale temporal modelling.

- **Verdict**
  For pure predictive accuracy on flat data, MLP remains the best performer. The Autoencoder + MLP variant sacrifices a small amount of accuracy in exchange for dimensional compression and latent-space interpretability. LSTM is highly promising for datasets with time-sequential structure and delivers strong generalization, while the DNN provides an intermediary option with deeper abstraction but slightly higher error. Together, these models confirm that deep learning architectures can capture the nonlinear emission patterns embedded in agro-food and demographic signals.

7. Discussion of the findings

According to results, **all four models can successfully predict national CO_2 emissions** using agro-food activity and population data with very good performance. A **Random Forest model** is the **best performing model (R^2 = 0.9958; RMSE ≈ 13 kt)** followed by **SVM (R^2 = 0.9928)**. For a **two-layer ANN**, we still obtain a high R^2 of **0.989**, and the **autoencoder-MLP hybrid** delivers an acceptable value of **0.963**. In all models, the **most influential predictors** consistently include **manure management, fertiliser usage, food transport, and total population** – clear, **policy-relevant variables**.

A **closer analysis of residuals** reveals further insight: **RF and ANN show slightly increased prediction error** for the **largest emitters** (suggesting **heteroscedasticity**), while **SVM under-predicts** in these extreme cases. The **autoencoder model loses some accuracy** due to a **latent space** that doesn't fully capture high-, mid-, and low-emission groupings. This suggests that **one-size-fits-all models work generally**, but **adaptive network structures** might be needed for outlier or temporal evolution scenarios.

Yet there are **important limitations** to consider. First, the **dataset ends in 2020**, and shifting conditions—such as **post-pandemic behaviors or climate policies**—could require **model retraining**. Second, **country-level aggregation may mask within-country disparities**, especially in large or diverse nations, potentially **diluting emission hotspots**. Lastly, **systematic under-reporting** in source inventories may **carry over into model predictions**, which statistical tools alone cannot correct.

In conclusion, the findings demonstrate that a **well-calibrated Random Forest model** offers **near-agency level predictive accuracy**. Meanwhile, **SVM and deep learning approaches** provide **valuable perspectives on data dimensionality and latent structure**. Achieving this level of performance will require **regular model updates**, **integration of emerging features** (like **regenerative agriculture siting**), and possibly **ensemble-based hybrid strategies**. Most importantly, the **strong R^2 values** across all models confirm that **agro-food and demographic variables are not just correlated but are reliable surrogates for national CO_2 emissions**—a finding that upholds the central hypothesis and aligns with **Learning Outcome 2's** emphasis on **reflective, context-aware model evaluation**.

8) Data Management Plan and Author Contribution statement

All four authors—Shawaiz Ahmed, Siva Shanker, Aditya Khachar, and Preyanka Chander—collectively conceived the study, formulated the research question, located and documented the dataset. Shawaiz Ahmed and Siva Shanker led the data-preparation and cleaning workflow, whereas Aditya Khachar and Preyanka Chander headed the exploratory data analysis and Principal-Component Analysis that reshaped the feature space. Throughout the project, every member reviewed one another's code, exchanged interim reports, and critically discussed results and limitations. Consequently, each author made a substantial technical contribution and participated in the critical interpretation of findings, fully satisfying the two authorship-contribution requirements.

9) Reference

Data source:- **https://www.gigasheet.com/sample-data/agri-food-co2-emission-dataset---forecasting-ml**

KNN Imputation – **https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html**

Scikit-learn PCA – **https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html**

K-Means with PCA – **https://towardsdatascience.com/k-means-clustering-introduction-to-machine-learning-algorithms-347e6b5d0c65**

Random Forests in Scikit-learn – https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

TensorFlow MLP Tutorial – https://www.tensorflow.org/tutorials/keras/classification

DMP:- Attached in Appendix