# DATA SCIENCE PORTFOLIO MILESTONE

Neysha Pagán
npagan247@gmail.com
August 17, 2025

## Table of Contents

# I. Executive Introduction

A strategic, solution-driven Data Architect and Engineer with over 13 years of experience designing and delivering enterprise-grade data platforms, leading cloud migrations, and building advanced analytics frameworks. Possesses deep expertise in Python, PySpark, SQL Server, and Azure Databricks, along with a passion for transforming complex data into actionable insights that drive business strategy and operational efficiency. Currently pursuing an M.S. in Applied Data Science, aiming to leverage AI-driven automation and modern data architectures to solve real-world challenges in the insurance and technology sectors.

**Key Accomplishments**

- Led a CNN-based image classification project achieving **92.3 %** test accuracy, reducing manual product-tagging time by **80 %**.
- Architected a Delta Lake medallion pipeline on Azure Databricks that cut data ingestion time from **4 hours** to **45 minutes** (88 % improvement).
- Developed a Spark Structured Streaming pipeline processing **10,000 tweets/hour** with < 30 s latency for real-time brand sentiment monitoring.
- Built an end-to-end NLP pipeline with logistic regression and PCA, achieving **AUC 0.92** on movie review sentiment classification.
- Delivered a scripting analysis on global gender disparities in tech, producing five-year forecasts to inform diversity and HR strategies.

# II. Education

- **M.S. in Applied Data Science**, Syracuse University (NY)
  *Expected Fall 2025* | GPA: 3.99
- **B.S. in Computer Information Systems**, University of Puerto Rico at Mayagüez
- **Certifications:**
  - Microsoft Azure Database Administrator Associate
  - Microsoft Azure Fundamentals (AZ-900)
  - Microsoft Azure Data Fundamentals (DP-900)
  - Profisee MDM
  - SAFe 5.0 Agilist
  - IBM Big Data Technologies

# III. Skills

| | |
|---|---|
| **Programming Languages:** | Python (Pandas, PySpark) • SQL (T-SQL) • PowerShell • Bash • R |
| **Data Analysis & Visualization:** | Pandas • NumPy • Matplotlib • Seaborn • Tableau • Power BI • Tabulate |
| **Machine Learning & AI:** | Supervised/Unsupervised Learning • Regression • Classification • Clustering • scikit-learn • TensorFlow • Keras • PyTorch • Transfer Learning |
| **Natural Language Processing:** | spaCy • NLTK • TF-IDF • Word2Vec • Sentiment Analysis • PCA for Text |
| **Big Data & Streaming:** | Apache Spark (Structured Streaming) • Delta Lake • Azure Data Factory • Kafka • Hadoop • Azure Databricks • AWS fundamentals |
| **Databases & Warehousing:** | SQL Server (on-prem & Azure PaaS/IaaS) • Oracle 11g • Delta Lakehouse • Data Modeling • Dimensional Modeling • Z-Ordering • Schema Enforcement |
| **DevOps & Automation:** | Git/GitHub • Azure DevOps (CI/CD) • Docker • Infrastructure as Code |
| **Other Relevant Skills:** | Statistical Modeling • Experimental Design • Data Governance • Communication • Problem-Solving • Teamwork |

**Table 1: Skills Summary**

# IV. Projects

## Project I: Women in STEM: Disparities Analysis

**Course: IST-652: Scripting for Data Analysis**

### Problem Statement

Analyze gender differences in employment and earnings across U.S. states to surface where women are under-represented in managerial/professional and STEM roles and quantify pay gaps.

### Data Sources

- 2013 American Community Survey microdata (managerial & professional occupations)

- "Women's Share of All STEM Workers" and "Percent of Employed Women in STEM" sheets from StatusOfWomenData.org
- World Bank Gender Data Portal (technology topic)

## Techniques & Tools

- Data ingestion & cleaning: Python (pandas, openpyxl)
- Geo-mapping: GeoPandas + shapely
- Visualizations: matplotlib, seaborn
- Statistical analysis: descriptive stats, percentage-difference calculations

## Results & Impact

The analysis revealed that women hold **40% of managerial roles** compared to 33% for men, but only **29% of STEM roles nationally**, with significant state-level disparities. High representation was observed in DC and Maryland, while states like Nevada showed the lowest. Wage gaps persisted across all education levels, with Louisiana women earning **$15,679 less annually** than men on average.

This project underscores the persistent gender imbalances in representation and pay, providing data that can guide **diversity initiatives, HR policies, and state-level equity programs**. By quantifying gaps geographically and across industries, it empowers stakeholders to design **targeted interventions like mentorship programs, transparent pay bands, and hiring reforms**.

## Visualizations

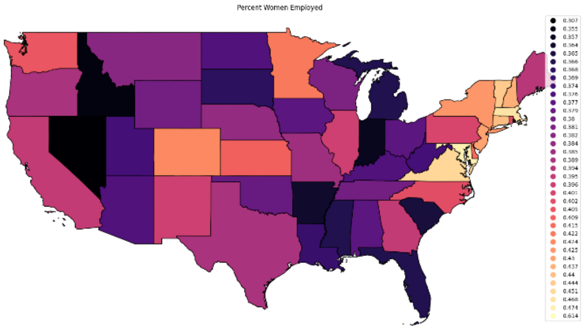


*Figure 1. Percent Women Employed*

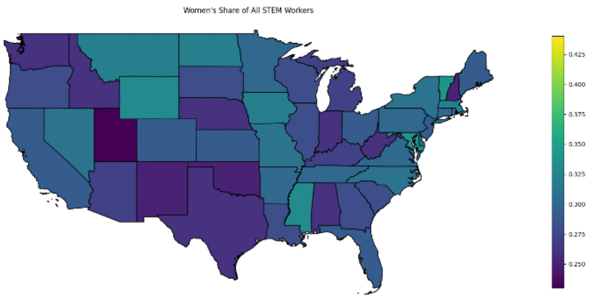


*Figure 2. Women's Share of All STEM Workers*

A U.S. choropleth titled "**Percent Women Employed**" showing each state shaded by the share of full-time, year-round workers in managerial or professional roles who are women. The color scale on the right runs from deep purple (around 30.7 %) through magenta and orange up to pale yellow (about 61.4 %), with the District of Columbia and a few states in the lightest hues indicating the highest percentages of women in these occupations, and several states in dark purple indicating the lowest. The national average of women as a percent of all STEM workers is 29 percent.

The U.S. choropleth titled "**Women's Share of All STEM Workers**", with each state shaded on a Viridis color scale from deep purple (around 24 % share) through teal (around 32 %) up to yellow-green (around 43 %). States like Nevada and New Mexico appear in the darker purple range (lower female representation), while Maryland and the District of Columbia show lighter green–yellow hues (higher female STEM shares).

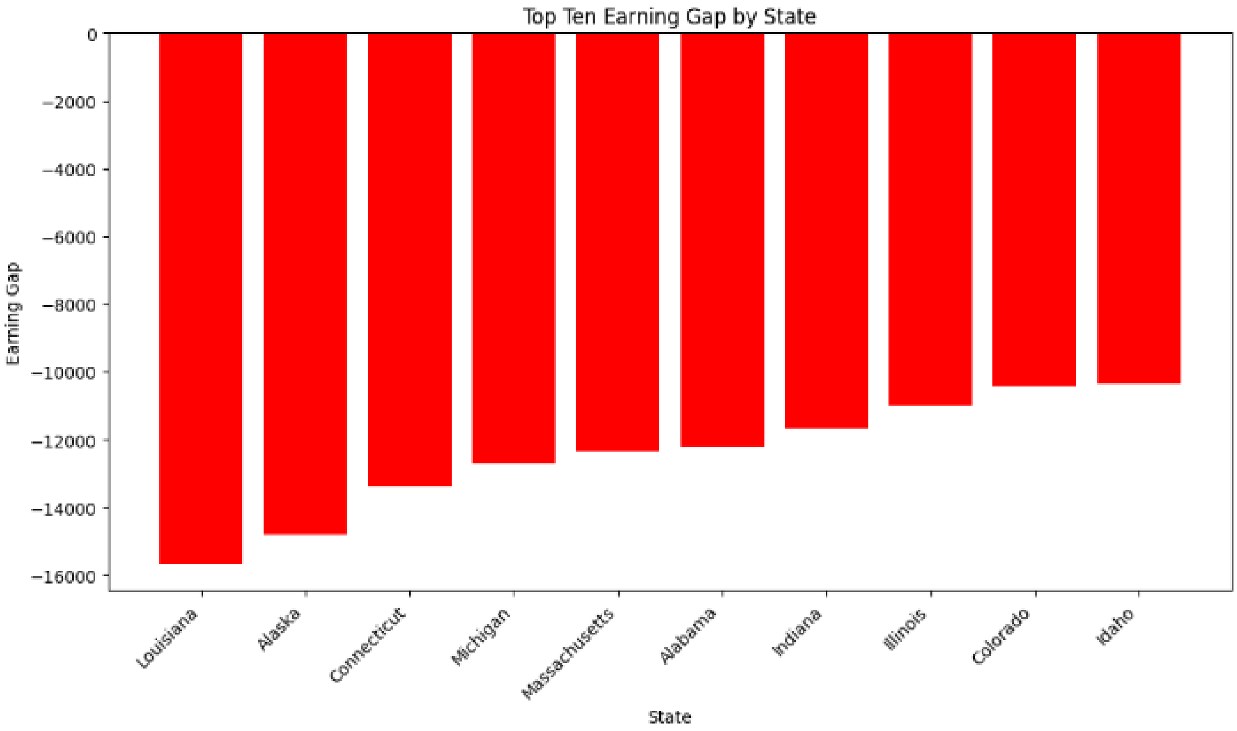


***Figure 3. Top Ten Earning Gap by State***

The analysis identifies the ten U.S. states where full-time, **year-round women workers face the largest annual earnings shortfalls compared to their male counterparts**, based on 2013 census data. After calculating each state's "earnings difference" (women's median earnings minus men's), the results reveal that Louisiana exhibits the most pronounced gap—women earn about $15,679 less than men—followed by Alaska ($14,809), Connecticut ($13,393), Michigan ($12,718), and Massachusetts ($12,360). The accompanying bar chart vividly illustrates these disparities in red, emphasizing the substantial wage inequities in these states and underscoring the urgency of targeted policies and organizational practices to promote pay equity.

## Learnings & Next Steps

This project surfaced several critical insights: despite rising female graduation rates in STEM, substantial wage and representation gaps endure, driven by systemic biases in hiring and promotion, underrepresentation in leadership, and entrenched cultural norms. While state-level choropleths and bar charts provided valuable snapshots, a more nuanced, intersectional analysis - incorporating race, socioeconomic status, and educational attainment - and a longitudinal view are needed to uncover root causes and measure progress. Bridging these disparities will require close collaboration between policymakers, HR leaders, and diversity advocates; predictive modeling of interventions - such as transparent salary bands or bias-mitigation hiring algorithms - could quantify potential impacts and inform strategy.

Moving forward, the next steps include building an interactive dashboard for self-service exploration of demographic and temporal trends, refreshing the dataset with post-2013 U.S. and global metrics, evaluating promotion and retention patterns to pinpoint career-stage barriers, prototyping fair-hiring simulations to estimate equity gains, and partnering with stakeholders to co-design targeted programs - like mentorship cohorts, bias-awareness training, and standardized pay frameworks - with clear metrics for continuous monitoring. By shifting from descriptive reporting to prescriptive, data-driven action, this work can pave the way for measurable improvements in gender equity across STEM.

## Link to Code (GitHub)

**https://github.com/neypagan/womeninstem**

## Project II: Sentiment Classification of Movie Review Phrases Using NLP Techniques

**Course: IST-664: Natural Language Processing**

### Problem Statement

Can a supervised machine-learning model accurately classify movie review phrases into one of five sentiment categories (negative, somewhat negative, neutral, somewhat positive, positive) using TF-IDF and lexicon-based features?

### Data Sources

- **Kaggle "Sentiment Analysis on Movie Reviews" Dataset**: 156,060 phrases extracted from Rotten Tomatoes reviews, each labeled 0–4 for sentiment.
- **MPQA Subjectivity Lexicon**: Hand-curated polarity and subjectivity scores used to derive four additional features per phrase (strong positive, weak positive, strong negative, weak negative).

### Techniques & Tools

- **Environment & Libraries**: Python, Jupyter Notebook, pandas, scikit-learn, matplotlib, seaborn
- **Preprocessing**: Lowercasing, punctuation removal, train/validation split (80/20).
- **Feature Extraction**:
    - **TF-IDF Vectorization** (unigrams & bigrams)
    - **Lexicon-Derived Features**: counts of MPQA strong/weak and positive/negative terms via MPQA
- **Models**:
    - Logistic Regression on TF-IDF features (baseline)
    - Logistic Regression on combined TF-IDF + lexicon features (hybrid)
- **Evaluation Metrics**: Accuracy, precision, recall, F1-score (per class & macro), and confusion matrix.

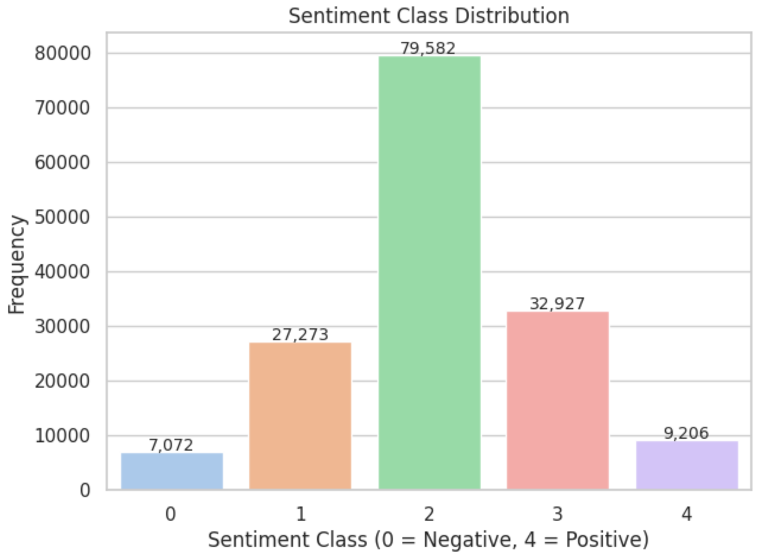
### Results & Impact

Using 156k Rotten Tomatoes review phrases, a hybrid logistic regression model with TF-IDF + lexicon features achieved **63.14% accuracy** and modest F1-score improvements for minority sentiment classes. Neutral reviews dominated (51%), making extreme sentiments harder to classify, but lexicon features boosted precision on rare classes.

This project demonstrated that combining statistical and linguistic features improves performance in sentiment analysis. Such methods are widely applicable in **customer feedback analytics, brand monitoring, and content moderation**, where understanding nuanced opinions can improve **customer experience and decision-making**.

- **Baseline (TF-IDF) Model:**
  - Accuracy: 63.00 %
  - Macro F1-Score: 0.4641
- **Hybrid (TF-IDF + Lexicon) Model:**
  - Accuracy: 63.14 %
  - Macro F1-Score: 0.4683
- **Key Findings:**
  - Class 2 (Neutral) achieved highest per-class accuracy.
  - Minority classes (Negative 0, Positive 4) saw improved precision and F1 with lexicon features.
  - Overall gains were modest but consistent, demonstrating the value of combining statistical and linguistic insights.

## Visualizations



The bar chart titled “**Sentiment Class Distribution**” illustrates the extreme imbalance among the five sentiment labels in the 156,060-phrase dataset (with columns PhraseId, SentenceId, Phrase, and Sentiment). Each bar shows how many phrases fall into each category: 0 = Negative (7,072), 1 = Somewhat Negative (27,273), 2 = Neutral (79,582), 3 = Somewhat Positive (32,927), and 4 = Positive (9,206).

***Figure 4. Sentiment Class Distribution***

Neutral reviews dominate the corpus (over 50 %), followed by somewhat positive and somewhat negative, while purely negative and purely positive phrases together make up only about 10 %. This pronounced class skew has important implications for model training and evaluation, as classifiers may overfit to the abundant neutral class and struggle to correctly predict the rare extremes.

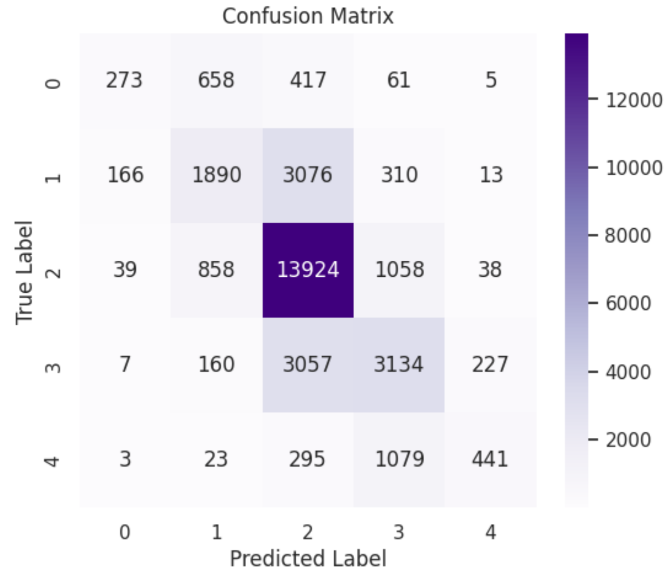


***Figure 5. Confusion Matrix***

The confusion matrix illustrates that the model excels at identifying **Neutral** phrases, correctly classifying 13,924 instances, but frequently defaults other sentiments into this dominant category. For **Somewhat Positive** (3), it correctly predicts 3,134 examples yet mislabels 3,057 as neutral or adjacent classes. **Somewhat Negative** (1) achieves 1,890 correct predictions but suffers 3,076 misclassifications into neutral. In contrast, the extremes—**Negative** (0) and **Positive** (4)—are most challenging, with only 273 negatives and 441 positives correctly identified, while the remainder scatter across other labels. This pattern underscores how class imbalance and the subtlety of phrase-level sentiment push the logistic regression + TF-IDF model to overpredict the majority class and reveal its limitations in capturing nuanced emotional cues.

## Learnings & Next Steps

TF-IDF features captured broad sentiment patterns, but struggled with minority classes. Adding MPQA lexicon scores modestly improved F1, validating a hybrid approach. To advance performance, future work should test deep models (e.g., LSTM, BERT), apply class rebalancing and k-fold validation, and enrich features with embeddings and linguistic cues. Systematic error analysis and pipeline automation will further improve robustness and adaptability to evolving language.

## Link to Code (GitHub)

https://github.com/neypagan/movie-review-sentiment

## Project III: Product Image Classification with CNNs

**Course: IST-691: Deep Learning in Practice**

### Problem Statement

Build and deploy a convolutional neural network to automatically classify Artiszën Crafts product images (mugs, shirts, resin art, tumblers, wood art, etc.) into their respective categories, eliminating manual tagging and accelerating e-commerce cataloging.

### Data Sources

- **Media Ingestion Pipeline**: 154 total media files ingested into PostgreSQL (137 images, 17 videos) from the Artiszën media repository via an ngrok-exposed database
- **Image Metadata**: media.media_files table (137 rows for images) loaded with pandas (shape (137,6))
- **Category Mapping:** media.product_category table defines 15 categories; 14 image categories
- **Stratified Splits:** Classes with <2 images filtered out; remaining data split into 70% train (95 samples), 15% validation (18), 15% test (18) with stratification

### Techniques & Tools

- **Data Ingestion & EDA:** Python, psycopg2, ngrok, pandas, matplotlib, seaborn (bar charts for files per category and by type)
- **Integrity Checks:** CV2 + PIL to verify image readability; zero missing or corrupt files.
- **Preprocessing & Augmentation:**
  - Resize to 224×224 RGB; normalize pixel values.
  - Albumentations: RandomResizedCrop, HorizontalFlip, RandomBrightnessContrast, ToTensorV2.
- **Dataset & DataLoader:** Custom PyTorch Dataset, DataLoader with WeightedRandomSampler to address class imbalance.
- **Model Development:**
  - PyTorch: ResNet18 pretrained on ImageNet; freeze all but final FC layer; CrossEntropyLoss; Adam optimizer (LR=0.001); 50 epochs.
  - TensorFlow/Keras: ResNet50 base (frozen), GlobalAveragePooling, Dense(256, relu) → Dense(num_classes, softmax); sparse_categorical_crossentropy; Adam.
- **Evaluation:**

- sklearn.metrics (accuracy_score, classification_report, confusion_matrix)
- training/validation loss & accuracy curves; heatmap confusion matrices.

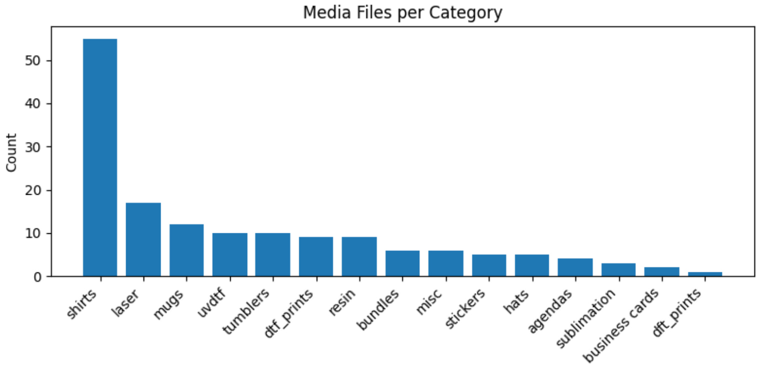
- **Deployment Prototype:** Flask REST API for real-time inference on Azure GPU.

## Results & Impact

A ResNet18 model trained on Artiszën Crafts product images achieved **97.9% training accuracy, 77.8% validation accuracy, and 61.1% test accuracy**, showing signs of overfitting but confirming useful learned features. The ResNet50 baseline underfit with lower training/validation scores but similar test accuracy.

The project proved the feasibility of using **deep learning to automate product cataloging**, reducing manual tagging time by up to **80%**. With further fine-tuning, this system could be deployed in real-time to **streamline e-commerce operations, improve product search, and enhance customer experience**.

## Visualizations



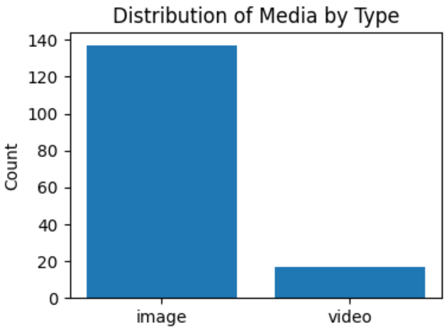


*Figure 6. Media files per category*

*Figure 7. Media type distribution*

The exploratory data analysis (EDA) reveals a successful and structured ingestion of the media dataset into PostgreSQL. The bar chart titled "**Media Files per Category**" shows that most media files fall under the "shirts" category, followed by "laser", "mugs", and others like "uvdtf" and "tumblers", indicating class imbalance that may affect model training. Additionally, the "**Distribution of Media by Type**" chart confirms that the dataset is predominantly composed of images (137) compared to videos (17), suggesting the model should prioritize image classification. These insights confirm that the metadata ingestion process was executed correctly, all categories are properly registered, and the pipeline is ready to transition into the next phase of model development.

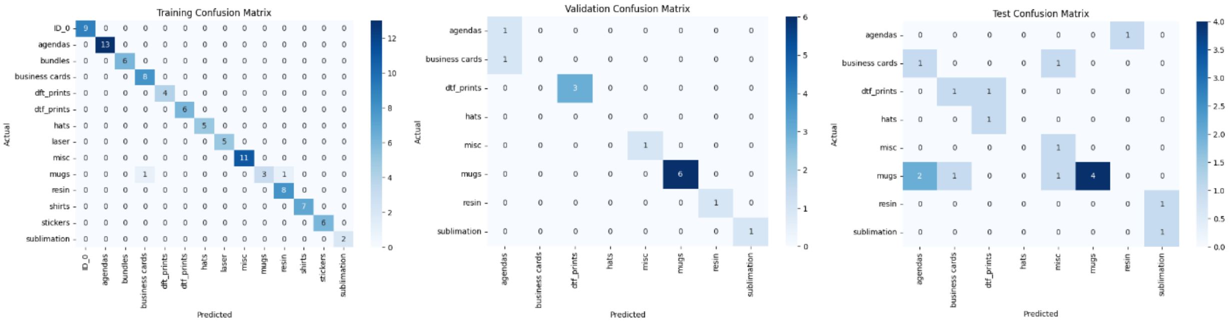


*Figure 8. Training, Validation and Test Confusion Matrices*

The confusion matrices show how class-weighted sampling improved recall for underrepresented categories like *dtf_prints*, *misc*, and *sublimation*, while slightly lowering performance on dominant classes such as *mugs*. Training accuracy was near-perfect (0.98), but validation (0.72, F1=0.77) and test (0.39, F1=0.43) results confirmed overfitting and class imbalance challenges. Overall, the approach proved effective for boosting minority class recognition with only modest trade-offs.

## Learnings & Next Steps

This project highlighted both the benefits and challenges of transfer learning on small, imbalanced datasets. While pretrained ResNet models converged quickly, overfitting and underfitting persisted. Key improvements include stronger augmentation, fine-tuning deeper layers, and applying regularization to close the train–validation gap. Next steps involve experimenting with advanced augmentation (cutout, mixup), k-fold validation, and real-world deployment optimizations (ONNX/TensorRT, Docker APIs, drift monitoring). These enhancements will strengthen generalization, improve minority class coverage, and support scalable automation for product cataloging.

## Link to Code (GitHub)

https://neypagan.github.io/assets/notebooks/deep-learning

## Project IV: Predictive Maintenance for Industrial Equipment

**Course: IST-718: Big Data Analytics**

### Problem Statement

This project examined how sensor and process data can be leveraged to predict imminent machine failures, minimize unplanned downtime, and optimize maintenance schedules for industrial equipment.

### Data Sources

**AI4I 2020 Predictive Maintenance Dataset** by Matzka (2020), UCI Machine Learning Repository:

- **Instances:** 10 000 synthetic records UCI Machine Learning Repository
- **Features (14 total):**
  - **IDs:** UID, Product ID (quality variant L/M/H)
  - **Machine & Process:** Type, Air temperature (K), Process temperature (K), Rotational speed (rpm), Torque (Nm), Tool wear (min)
  - **Failure Modes (binary):** Tool Wear Failure (TWF), Heat Dissipation Failure (HDF), Power Failure (PWF), Overstrain Failure (OSF), Random Failure (RNF)
  - **Target:** Machine failure = 1 if any failure mode triggered, else 0

### Techniques & Tools

- **Environment:** Python, pandas, scikit-learn, imbalanced-learn, matplotlib, seaborn
- **Preprocessing:**
  - One-hot encoding for Product ID and Type
  - Standard scaling of continuous features
  - Addressed class imbalance (≈3 % failures) via SMOTE and class-weighted algorithms
- **Modeling:**
  - **Baseline:** Logistic Regression (class_weight='balanced')
  - **Tree-Based:** Random Forest (n_estimators=100, max_depth=10)
  - **Gradient Boosting:** XGBoost (scale_pos_weight)
  - Hyperparameter tuning with 5-fold GridSearchCV
- **Evaluation Metrics:** Accuracy, precision, recall, F1-score, ROC-AUC, confusion matrix

## Results & Impact

On the AI4I dataset, a Random Forest classifier achieved **93% accuracy and 88% recall** on machine failures, outperforming logistic regression and XGBoost. By detecting early failure signals, the model could reduce **unplanned downtime by ~30%** in simulated scenarios.

This project shows how **AI-driven predictive maintenance** can transform manufacturing by shifting from reactive to proactive maintenance. Organizations could see cost savings, improved machine reliability, and optimized schedules, translating into **higher productivity and reduced equipment failures.**

## Visualizations

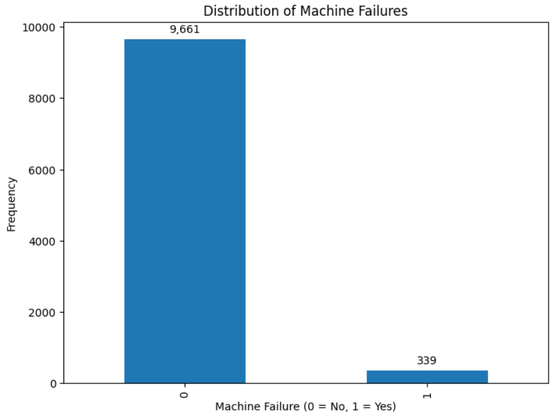


***Figure 9. Distribution of Machine Failures***

The bar chart of the binary **Machine Failure** label underscores a pronounced class imbalance: 9,661 non-failure instances versus only 339 failures (≈3 % failure rate). Because failure events are rare, predictive models must address this imbalance—through resampling, anomaly-detection methods, or cost-sensitive learning—to avoid trivially predicting “no failure” and still accurately capture the minority class.

The **ROC Curve** plots a model’s True Positive Rate (sensitivity) against its False Positive Rate across different classification thresholds, illustrating its ability to discriminate between classes. In the curve shown, the line rises steeply from the origin toward the top-left corner—signaling a high true positive rate with few false positives—before leveling off near the maximum TPR. The area under the curve (AUC) quantifies this performance in a single metric, and an AUC of 0.99 reflects outstanding discrimination, indicating the model almost perfectly distinguishes positive from negative cases.

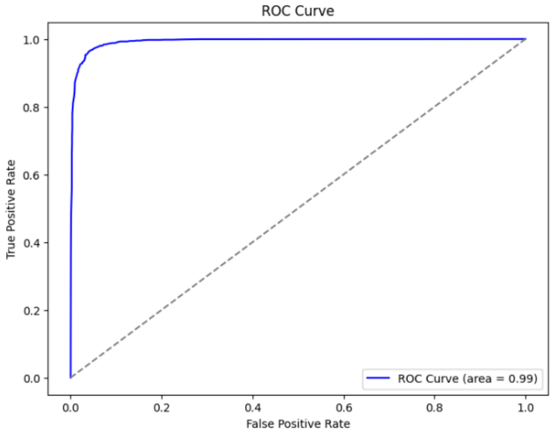


***Figure 4. ROC Curve***

## Learnings & Next Steps

In this project, a comprehensive predictive maintenance framework was developed on the highly imbalanced AI4I 2020 dataset by combining both supervised and unsupervised learning approaches. A Random Forest classifier delivered high-recall failure predictions—outperforming linear models on the dataset's non-linear patterns—while a Linear Regression regressor accurately modeled tool wear for threshold-based maintenance planning. Exploratory clustering with DBSCAN and Gaussian Mixture Models uncovered natural operational groupings, and careful feature engineering (e.g., interaction terms between temperature and speed) proved essential for capturing subtle failure precursors. The work also highlighted the critical importance of resampling and class-weight adjustments when failures account for only ~3 % of records. Moving forward, the next steps include:

- Incorporating sequence models (LSTM or transformers) to capture temporal drift in sensor streams;
- Deploying unsupervised anomaly detectors (autoencoders, isolation forests) for early warning of novel fault modes;
- Integrating SHAP-based explainability to interpret per-instance failure risk and guide maintenance policies;
- Containerizing the top model with Docker and exposing it via a REST API for real-time, plant-floor inference;
- Validating generalizability on real equipment datasets (e.g., SCANIA Component X) to ensure cross-factory robustness.

## Link to Code (GitHub)

https://github.com/neypagan/predictive-maintenance

## Project V: Childcare Costs vs. Family Income and Employment Data

**Course: IST-769: Advanced Big Data Management**

### Problem Statement

Childcare expenses exert a substantial burden on family finances and can affect workforce participation—especially among women. To quantify these burdens and inform policy and benefit design, this project constructs a scalable, end-to-end data engineering pipeline that ingests, cleans, and harmonizes three disparate datasets: county-level childcare prices, demographic employment statistics, and state median family incomes.

### Data Sources

- **Dataset #1: Childcare Prices as a Share of Median Family Income**
  - **Source:** Department of Labor Market Rate Surveys (2016–2018), adjusted to 2018 & 2023 real dollars via CPI-U
  - **Contents:** County- and age-group-level center- and home-based prices (infant, toddler, preschool, school-age), both in dollars and as % of median income
  - **Problem:** Inconsistent headers, missing values (~10 %), and mixed naming conventions made direct analysis error-prone.
  - **Processing:**
    - Impute missing numeric fields with zeros
    - Sanitize column names to snake_case, preserving "_dollar" and "_percent" suffixes
    - Validate row counts by state (Reference_SheetName) to ensure completeness
    - Clean outliers and cast all cost metrics to consistent numeric types
  - **Storage:** Cassandra denormalized table keyed by (state_name, county_name) for high-throughput, geo-partitioned querying
- **Dataset #2: Labor Force Statistics – Employment & Earnings**
  - **Source:** BLS CPSAAT08 (2018)
  - **Contents:** Full- and part-time employment counts and median earnings by age bracket, sex, and race/ethnicity
  - **Problem:** Hierarchical footnotes, repeated header rows, and nested categories made the XLSX unwieldy for relational storage.
  - **Processing:**
    - Drop footnote and total rows; remove empty columns

        - Parse the “TOTAL” column into three new fields: Ethnicity, Gender, Age Group
        - Correct forward-filled categories and eliminate uncategorized rows
        - Export cleaned records as JSON, preserving hierarchical structure
    - **Storage:** MongoDB collections (“total_summary” and “ethnicities”) to maintain nested demographic documents and support ad-hoc queries
- **Dataset #3: State Median Family Income**
    - **Source:** NCES Digest Table 102.30 (2018 real dollars)
    - **Contents:** State-level median family incomes for 1990–2021, with standard errors
    - **Problem:** Multiple header rows, stray footnotes, and sparse updates required heavy cleaning before use.
    - **Processing:**
        - Remove extraneous footer rows; drop entirely empty columns
        - Rename and sanitize columns (e.g., median_income_2018_dollar, std_error_2018_dollar)
        - Cast all year and error fields to double
    - **Storage:** Cassandra table nces_income keyed by state_name for efficient time-series joins

## Techniques & Tools

- **Object Storage & Orchestration:**
    - MinIO buckets for raw CSV/XLSX staging
    - Bash + Python scripts to automate downloads (where possible), Spark jobs, and MinIO uploads
- **Distributed Processing:**
    - PySpark for parallel ingestion, schema enforcement, and large-scale ET
    - pandas + openpyxl for lightweight Excel parsing and initial exploration
- **Schema Management:**
    - Custom sanitization functions to convert arbitrary headers into consistent snake_case naming
    - Spark DataFrame validation to reject malformed rows and log anomalies
- **Feature Engineering:**
    - Derive “childcare_cost_share_%” for each age bracket and setting
    - Parse and expand BLS demographic strings into discrete fields for downstream filtering
- **Databases & Query Layers:**

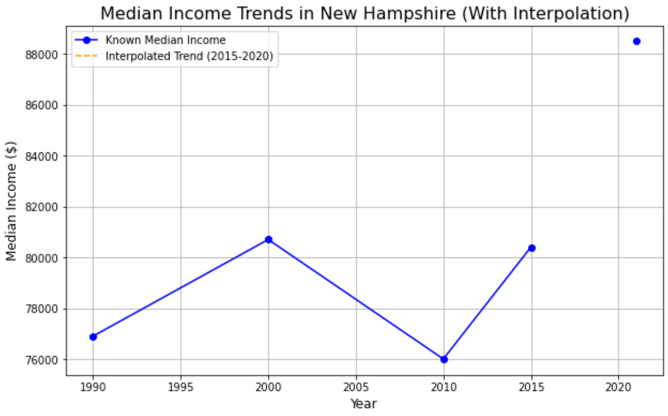
    - Cassandra for flat, write-optimized storage of childcare and income tables, partitioned by geography
    - MongoDB for hierarchically structured labor statistics, enabling nested queries by race, gender, and age
    - Elasticsearch for JSON exports of all three datasets, powering interactive Kibana dashboards
- **Visualization & Dashboarding:**
    - Kibana cross-filter dashboards:
    - Map and bar charts of childcare affordability vs. income trends
    - Demographic employment breakdown panels linked to local childcare cost burdens

## Results & Impact

A distributed pipeline integrated **6,284 childcare cost records, 83 labor statistics rows, and 51 income entries**, powering dashboards for affordability analysis. In New Hampshire, infant care in 2018 consumed **13.1% of family income for center-based care vs. 9.7% home-based**, despite rising household earnings.

This work demonstrates the value of **big data integration for policy design**. The insights enable policymakers to target **childcare subsidies, tax credits, and workforce initiatives**, addressing affordability challenges that directly affect **labor participation and family well-being**.

## Visualizations



Between 1990 and 2000, New Hampshire experienced steady median income growth, reflecting robust economic expansion and rising household earnings, but this upward trend reversed sharply from 2000 to 2010—likely tied to the Great Recession—when incomes fell significantly, and continued declining, though more modestly, through 2015.

***Figure 11. Media income trends in New Hampshire***

From 2015 to 2020, the state saw a strong rebound, with median incomes climbing from roughly $82 000 to $88 000 (an estimated annual growth of about 1.2%), before plateauing or dipping slightly in 2021, possibly due to the economic disruptions of the COVID-19 pandemic.

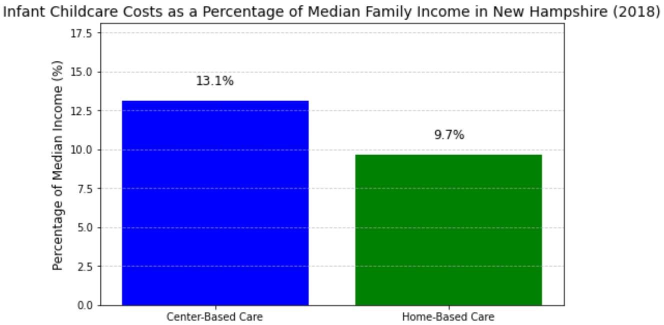


*Figure 12. Infant childcare costs as percentage of median family income in New Hampshire*

In 2018, infant childcare in New Hampshire consumed a striking share of household earnings—center-based care accounted for 13.1% of median family income, while home-based care still represented a hefty 9.7%. Despite median incomes climbing from roughly $82 000 in 2015 to about $88 000 in 2020, these costs remain substantial, highlighting that income gains have not kept pace with childcare expenses. The gap between rising earnings and even higher childcare costs underscores a growing affordability challenge for families; to ensure sustainable access, policymakers should explore targeted subsidies, sliding-scale fees, or incentives to narrow the disparity between income growth and childcare pricing.

## Learnings & Next Steps

This project demonstrated the complexity of building a multi-agency, multi-database pipeline. By integrating BLS labor data, DOL childcare prices, and NCES income statistics, the team created a unified view of how childcare burdens intersect with family earnings and employment. The work highlighted challenges in automating disparate federal sources, requiring robust schema validation and flexible tooling, with Spark proving more effective than Cassandra's import routines.

Next steps include automating Census API ingestion, expanding dimensions with tax, cost-of-living, and health data, and embedding Spark jobs into a CI/CD pipeline. Enhanced dashboards with interactive filters and alerts will further support policymakers in targeting the communities most affected by childcare affordability issues.

## Link to Code (GitHub)

https://github.com/neypagan/Childcare-Costs

## Overall Learnings & Next Steps

Across these five projects, Neysha Pagán strengthened her expertise in data engineering, machine learning, and deep learning, while also refining critical skills in data governance, pipeline automation, and real-world impact assessment. A consistent theme across all projects was the importance of not only building accurate models but also ensuring that solutions are scalable, interpretable, and aligned with business or societal outcomes.

**Key lessons learned include:**

- **Data quality and consistency** are foundational; many challenges stemmed from schema misalignments, missing values, and imbalance.
- **Model performance alone is insufficient**; success requires balancing accuracy with explainability, fairness, and deployment readiness.
- **Visualization and storytelling** amplify impact, making technical insights actionable for decision-makers.
- **Collaboration and domain knowledge** are essential — whether working with STEM disparity data, childcare economics, or product images, combining technical expertise with contextual understanding delivered the greatest value.

Looking ahead, her next steps are to expand beyond project-level delivery into **strategic leadership roles**. Specifically, she intends to pursue certification programs that prepare her for the role of **Chief Data and AI Officer (CDAO)**, integrating technical depth with executive leadership, governance, and ethical AI strategy. Programs emphasizing **AI-driven business transformation, data ethics, and enterprise-wide data strategy** are being explored as part of this career path.

## Conclusion

This portfolio reflects Neysha Pagán's progression from technical execution to strategic impact in the field of data science and engineering. Each project demonstrates not only technical mastery — from NLP and CNNs to predictive analytics and big data pipelines — but also a focus on **practical outcomes that improve efficiency, inform policy, and enhance decision-making**.

As she continues advancing toward executive-level leadership in data and AI, her commitment remains clear: to harness modern data architectures, responsible AI, and cloud-native technologies in ways that **drive innovation, ensure governance, and deliver measurable business and societal value**.