

Syracuse University

School of Information Studies

# Master of Science

Applied Data Science

Secondary Core: Artificial Intelligence

## Portfolio Milestone

Benjamin Heindl

359536618

10 March 2024

[Syracuse MSDS Project Portfolio \(github.com\)](https://github.com/bheindl/Syracuse-MSDS-Project-Portfolio)



# Introduction

The Project Portfolio for the MS in Applied Data Science aims to demonstrate a student's abilities in several key areas: collecting, storing, and accessing diverse data effectively; generating valuable insights applicable to various domains using the complete data science process; creating predictive models and visualizations for actionable intelligence; coding in R and Python for these tasks; effectively communicating these insights to various stakeholders; and upholding ethical standards related to fairness, bias, transparency, and privacy in their work.

The program's courses involved the preparation of reports and presentations that showcased the competencies acquired, such as:

- IST 659 - Database Administration
- IST 652 - Scripting for Data Analysis
- IST 707 - Applied Machine Learning
- IST 664 - Natural Language Processing
- IST 691 - Deep Learning in Practice

# The Applied Data Science Program Learning Objectives

1. Collect, store, and access data by identifying and leveraging applicable technologies
2. Create actionable insight across a range of contexts (e.g. societal, business, political), using data and the full data science life cycle
3. Apply visualization and predictive models to help generate actionable insight
4. Use programming languages such as R and Python to support the generation of actionable insight
5. Communicate insights gained via visualization and analytics to a broad range of audiences (including project sponsors and technical team leads)
6. Apply ethics in the development, use and evaluation of data and predictive models (e.g., fairness, bias, transparency, privacy)



# IST 659 Database Administration Concepts & Database Management

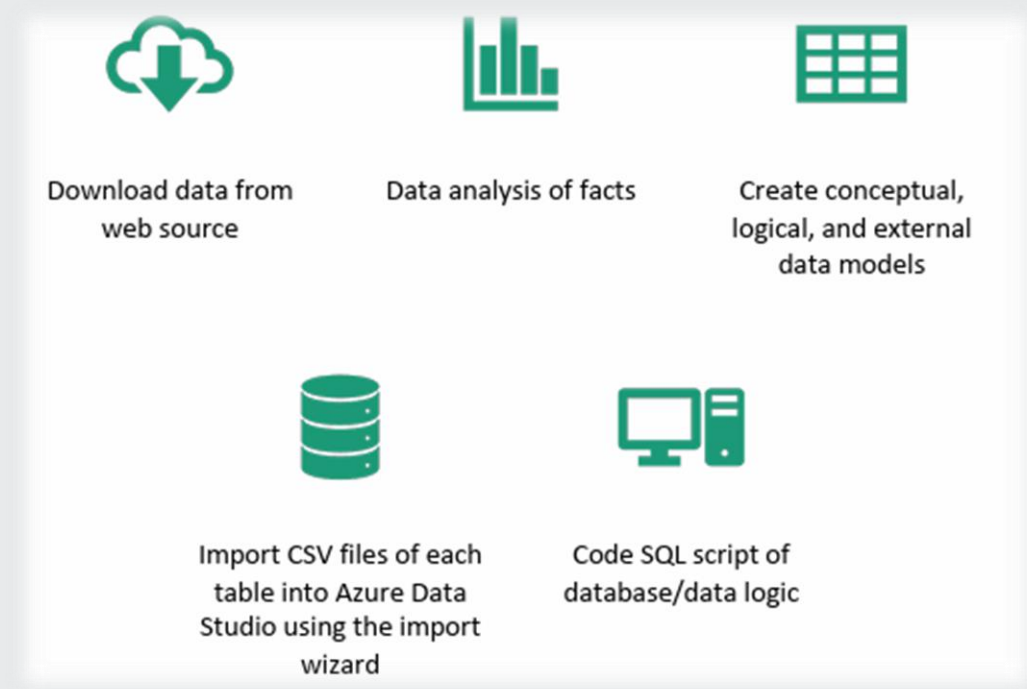
## PGA Golf Database



# IST 659: Database Administration

## *Project Overview*

- Database app for searching PGA tour golf tournament results from 2018-2022
  - Easy filtering by player, tournament, and/or date
  - Reliable information source for research and analysis
  - Ideal for tracking the progress of favorite players or creating custom reports



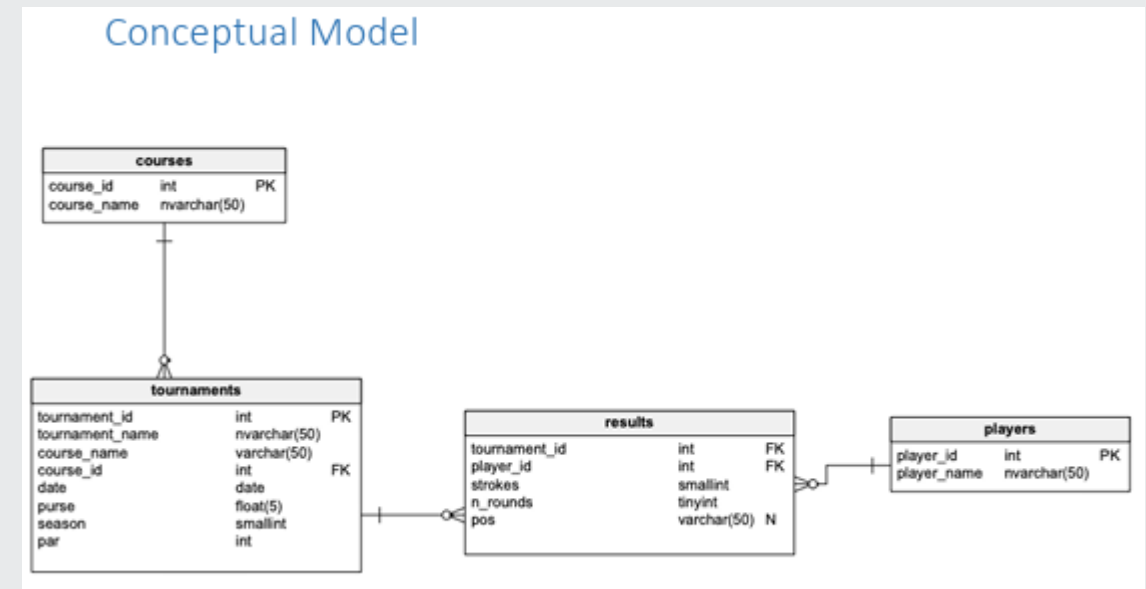
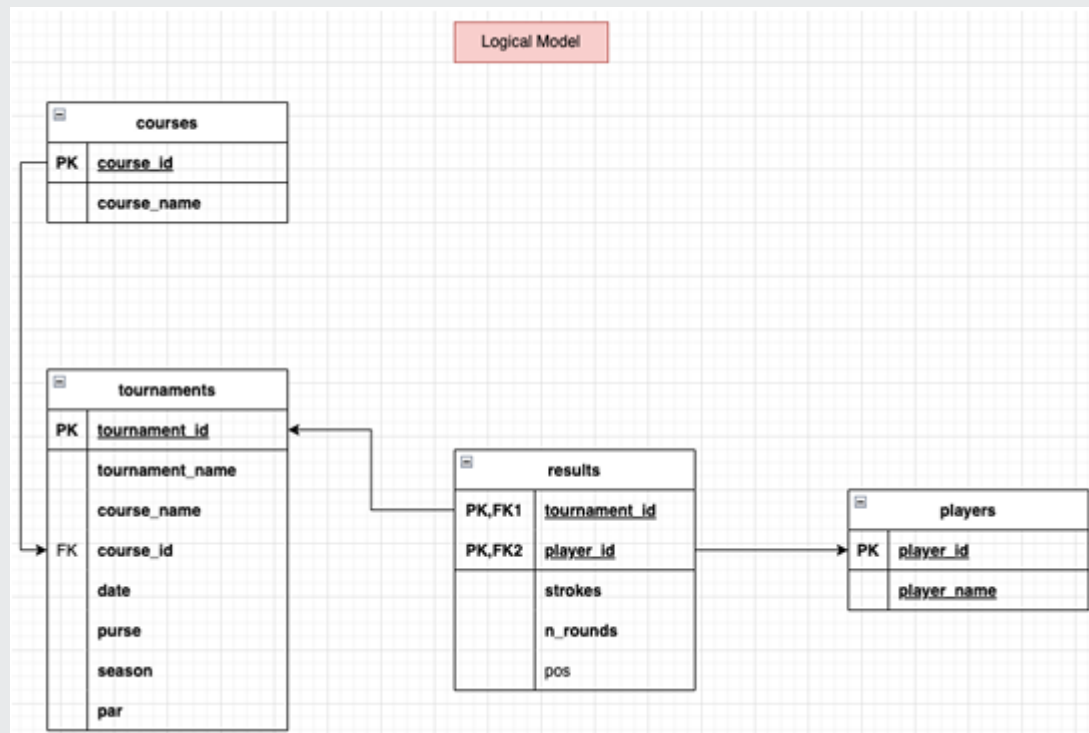
# IST 659: Database Administration

## *Modeling & Table Creation*

- Conceptual and Logical models were developed to organize the relationships between the various tables and fields of the database.
- Docker and Azure Data Studio were utilized to create and populate the tables.

# IST 659: Database Administration

## *Logical & Conceptual Models*





# IST 659: Database Administration

## Testing Accuracy of PGA Database


Database Query

```
62 -- 2021 Masters Results
63 SELECT p.player_name, r.strokes, r.n_rounds, r.pos
64 FROM players p
65 JOIN results r ON p.player_id = r.player_id
66 JOIN tournaments t ON r.tournament_id = t.tournament_id
67 WHERE t.tournament_name = 'Masters Tournament'
68 AND YEAR(t.date) = 2021
69 AND r.pos > 0 -- Exclude players with pos = 0
70 ORDER BY r.pos ASC;
71
```

Results Messages

	player_name	strokes	n_rounds	pos
1	Hideki Matsuyama	278	4	1
2	Will Zalatoris	279	4	2
3	Xander Schauffele	281	4	3
4	Jordan Spieth	281	4	3
5	Marc Leishman	282	4	5
6	Jon Rahm	282	4	5
7	Justin Rose	283	4	7
8	Patrick Reed	284	4	8
9	Corey Connors	284	4	8
10	Cameron Smith	285	4	10

Official PGA Results

 MASTERS							
POS	PLAYER	R1	R2	R3	R4	TOTAL SCORE	TOTAL PAR
1	H Matsuyama	69	71	65	73	278	-10
2	W Zalatoris	70	68	71	70	279	-9
T3	J Spieth	71	68	72	70	281	-7
T3	X Schauffele	72	69	68	72	281	-7
T5	J Rahm	72	72	72	66	282	-6
T5	M Leishman	72	67	70	73	282	-6
7	J Rose	65	72	72	74	283	-5
T8	P Reed	70	75	70	69	284	-4
T8	C Connors	73	69	68	74	284	-4
T10	C Smith	74	68	73	70	285	-3



# IST 659: Database Administration

## *Web App Platform*

- Developed a database app for searching PGA tour golf tournament results from 2018-2022.
- Allows filtering by player, tournament, or date for quick information retrieval.
- Tailored for golf fans and statisticians with complete PGA tour data coverage.
- Features a user-friendly interface with up-to-date, reliable information.
- Equipped with a two-way slider for selecting and refining results within a date range.
- Designed for simplicity and ease of use, ensuring efficient navigation and data access.



# IST 659: Database Administration

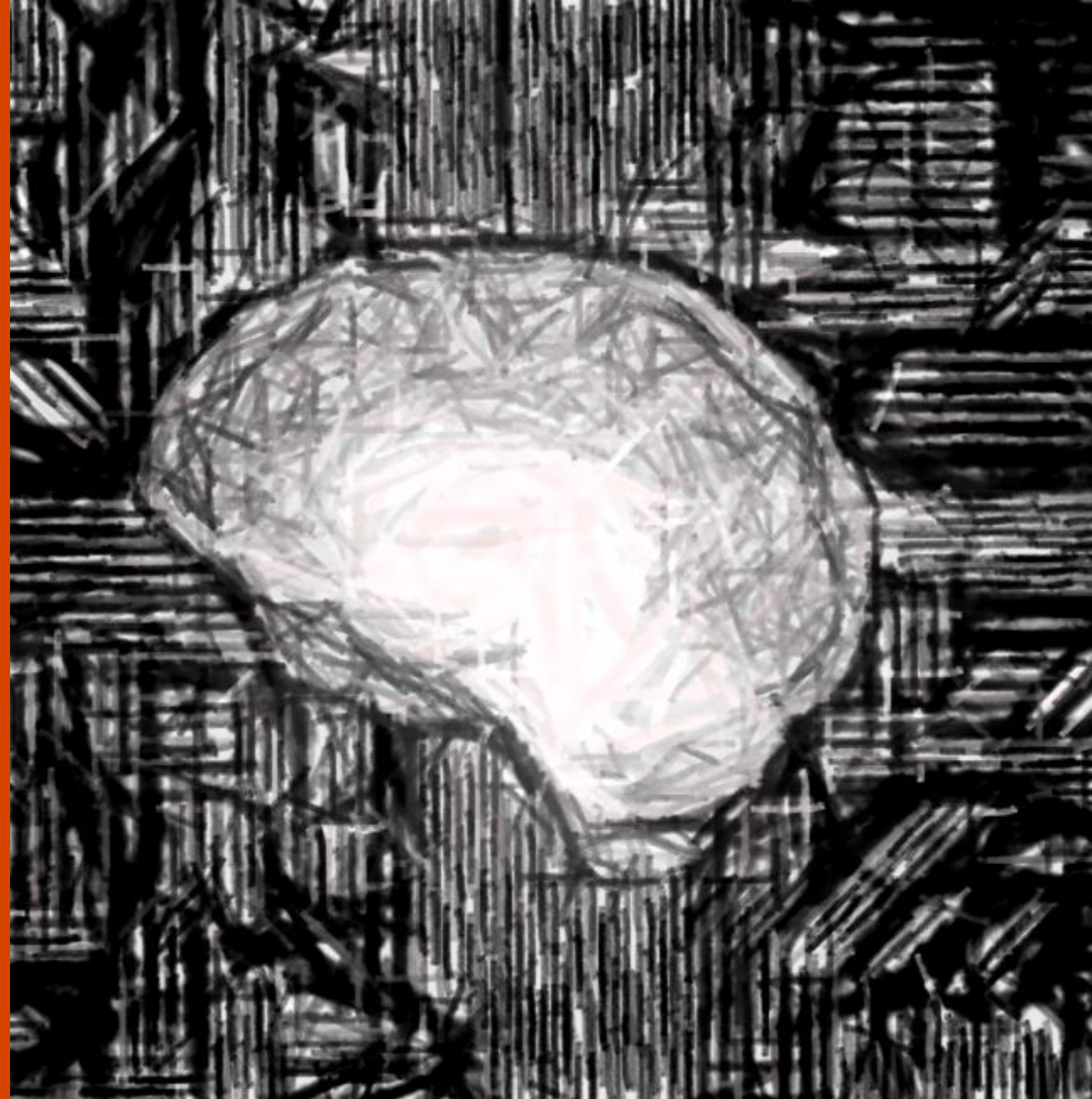
## *Reflection*

- Deepened understanding of data/database concepts and the development lifecycle.
- Learned to analyze business issues and devise relational database solutions.
- Improved design, implementation, and SQL querying skills through practical assignments.
- Employed knowledge from data analysis courses in app development.
- Provided a user-friendly app with extensive data, useful for player tracking or report generation.

# IST 707

## Applied Machine Learning

Multimodal Exploration of Mass Killings: A  
Machine Learning Approach



# IST 707: Data Analytics

## *Key Analytic Questions - KAQS*

- How can narrative summaries within the mass killings dataset be transformed into a tokenized structure?
- Using the TF-IDF weighted data structure, how well do machine learning methods perform in classifying the type of killing?
- Of the models considered in the analysis, which models perform best?
- How does rebalancing the multi-class outcome (“type”) change model performance?



# IST 707: Data Analytics

## *Analytic Process*

- **Business understanding**
  - Background, business objectives, overall project plan - integrate project requirements with coursework
  - Use of R, multiple packages (notably caret)
- **Data understanding**
  - Collection of data, initial exploratory data analysis (EDA)
  - Data quality - identify missing values, iterate throughout all phases
- **Data preparation**
  - Data cleansing - missing values, analysis and removal as appropriate
  - Central focus involved text mining methods: creation of a document-term matrix from narrative elements, creation of appropriate data structure and integration with the core dataset to support the modeling process
- **Modeling**
  - Classification trees, recursive partitioning, C5, support vector machines, random forest and k-means clustering

# IST 707: Data Analytics

## *Exploratory Data Analysis*

Figure 1. Data Summary – Incidents Dataset

	Descriptions	Value
1	Sample size (nrow)	553
2	No. of variables (ncol)	17
3	No. of numeric/interger variables	7
4	No. of factor variables	0
5	No. of text variables	10
6	No. of logical variables	0
7	No. of identifier variables	5
8	No. of date variables	0
9	No. of zero variance variables (uniform)	0
10	% of variables having complete cases	88.24% (15)
11	% of variables having >0% and <50% missing cases	5.88% (1)
12	% of variables having >=50% and <90% missing cases	5.88% (1)
13	% of variables having >=90% missing cases	0% (0)

Figure 2. Data Summary – Document-Term Matrix, Incidents Dataset

	Descriptions	Value
1	Sample size (nrow)	553
2	No. of variables (ncol)	168
3	No. of numeric/interger variables	168
4	No. of factor variables	0
5	No. of text variables	0
6	No. of logical variables	0
7	No. of identifier variables	1
8	No. of date variables	0
9	No. of zero variance variables (uniform)	0
10	% of variables having complete cases	100% (168)
11	% of variables having >0% and <50% missing cases	0% (0)
12	% of variables having >=50% and <90% missing cases	0% (0)
13	% of variables having >=90% missing cases	0% (0)

# IST 707: Data Analytics

## Text Analysis Preprocessing

Figure 3. Code Snippet: Text Analysis Preprocessing

```
## create corpus; convert the narrative field within the incidents dataset to the corpus; conduct the following preprocessing steps
## 1. convert all text to lower case, 2. remove punctuation, and 3. remove stopwords
incident_corpus <- Corpus(VectorSource(incidents2$narrative))
incident_corpus <- tm_map(incident_corpus, PlainTextDocument)
incident_corpus <- tm_map(incident_corpus, tolower)
incident_corpus <- tm_map(incident_corpus, removePunctuation)
incident_corpus <- tm_map(incident_corpus, removeWords(stopwords("english")))

## Two distinct approaches here - TF and TFIDF weighting; first is TF
## create the document term matrix w TF weighting; extract frequently occurring words (target roughly 170 words for the analysis)
## create dataframe of sparse matrix, one word per column
dtm <- DocumentTermMatrix(incident_corpus)
notSparse <- removeSparseTerms(dtm, 0.975)
finalWords <- as.data.frame(as.matrix(notSparse), stringsAsFactors = FALSE)
head(finalWords)

## create index column; check dimensions of data frame; view column names; examine subset and check summary of one of the terms
## conduct EDA
finalWords2 <- cbind(index = 1:nrow(finalWords), finalWords)
dim(finalWords2)
ExpData(finalWords2, type=1)
ExpData(finalWords2, type=2)
colnames(finalWords2)
finalWords2[148:158, 26:29]
summary(finalWords2$fire)
```

Figure 4. Tokens (Words) Extracted from Incidents Dataset

```
> colnames(fw3)
```

[1] "type"	"fire"	"gunman"	"killed"	"opened"	"police"	"apartment"	"children"
[9] "died"	"fatally"	"girlfriend"	"inside"	"later"	"shot"	"three"	"man"
[17] "night"	"one"	"two"	"women"	"friends"	"hone"	"life"	"took"
[25] "wife"	"according"	"arrested"	"authorities"	"charged"	"connection"	"days"	"family"
[33] "five"	"house"	"killing"	"murders"	"neighbors"	"rifle"	"several"	"shooting"
[41] "went"	"allegedly"	"fired"	"injuring"	"parents"	"people"	"residence"	"four"
[49] "others"	"party"	"victims"	"assailant"	"injured"	"adults"	"woman"	"back"
[57] "death"	"entered"	"outside"	"sons"	"committing"	"day"	"also"	"child"
[65] "dead"	"dispute"	"found"	"reportedly"	"responding"	"men"	"murdersuicide"	"another"
[73] "call"	"counts"	"discovered"	"father"	"murder"	"called"	"members"	"drove"
[81] "related"	"seven"	"shootings"	"standoff"	"believe"	"six"	"sister"	"survived"
[89] "eight"	"domestic"	"history"	"mother"	"violence"	"suicide"	"lee"	"bodies"
[97] "suspect"	"stabbed"	"wounded"	"case"	"charges"	"exgirlfriend"	"order"	"time"
[105] "investigators"	"county"	"incident"	"including"	"officer"	"drug"	"james"	"remains"
[113] "fourth"	"hospital"	"robbery"	"say"	"scene"	"vehicle"	"left"	"awaiting"
[121] "trial"	"handgun"	"set"	"daughter"	"daughters"	"returned"	"dropped"	"injuries"
[129] "officers"	"michael"	"guilty"	"parole"	"pleaded"	"sentence"	"without"	"committed"
[137] "mental"	"said"	"young"	"believed"	"unsolved"	"ended"	"convicted"	"prison"
[145] "sentenced"	"eligibility"	"ages"	"former"	"son"	"received"	"brother"	"used"
[153] "prior"	"months"	"car"	"years"	"given"	"earlier"	"estranged"	"kids"
[161] "told"	"relatives"	"sentences"	"kill"	"boyfriend"	"couple"	"serving"	"slayings"

# IST 707: Data Analytics

## Analytic Process – Exploration of Machine Learning Methods Using TF-IDF Weighted Tokens

Figure 5. Factor Levels for the Multi-Class Outcome

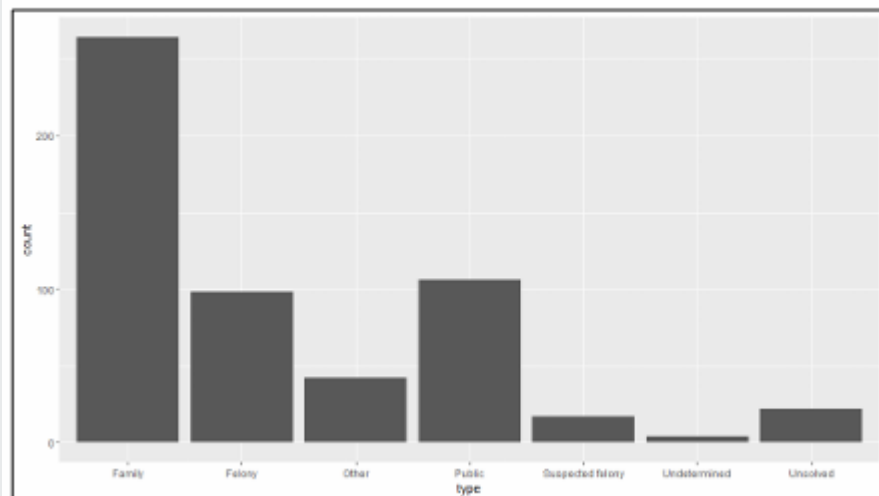


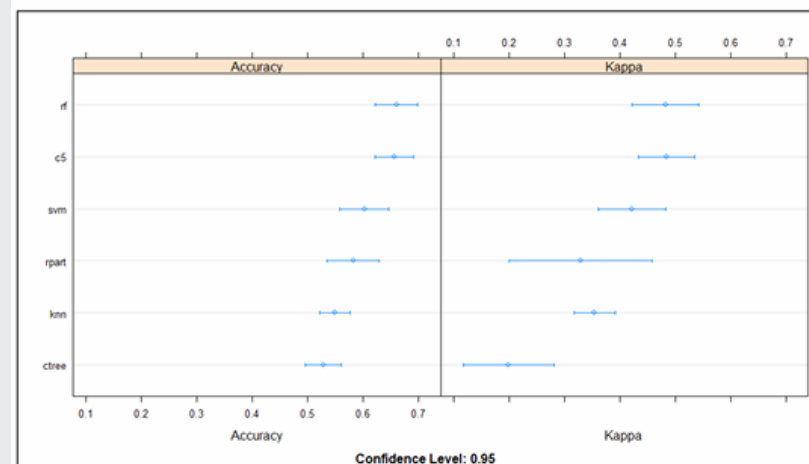
Figure 6. Accuracy and Kappa Statistics by Model

```
Call:
summary.resamples(object = results)

Models: ctree, rpart, c5, svm, knn, rf
Number of resamples: 10
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
Accuracy							
ctree	0.4629630	0.4888001	0.5456349	0.5286604	0.5599415	0.5849057	0
rpart	0.4814815	0.5507519	0.5857700	0.5821597	0.6284722	0.6792453	0
c5	0.5555556	0.6315789	0.6666667	0.6565030	0.6755952	0.7358491	0
svm	0.5178571	0.5555556	0.5862573	0.6027475	0.6578164	0.6851852	0
knn	0.4814815	0.5231481	0.5584795	0.5493028	0.5815364	0.5964912	0
rf	0.5925926	0.6266447	0.6522989	0.6607141	0.7083333	0.7358491	0
Kappa							
ctree	0.01136364	0.1161165	0.2097977	0.1988840	0.2706449	0.4065934	0
rpart	0.00000000	0.3345313	0.3906120	0.3288896	0.4277723	0.5142857	0
c5	0.32919255	0.4608912	0.4786608	0.4842673	0.5197922	0.5927552	0
svm	0.30990415	0.3466647	0.4063818	0.4218467	0.5071457	0.5248447	0
knn	0.26694717	0.3199713	0.3735661	0.3542111	0.3809188	0.4307425	0
rf	0.33926585	0.4495966	0.4680049	0.4824585	0.5490586	0.5998094	0

Figure 7. Accuracy and Kappa Statistics and 95% CIs by Model





# IST 707: Data Analytics

## *Reflection*

- Text mining with machine learning models links narratives to mass killings outcomes; integrated data frames use tokenization and weighting.
- Data wrangling ensures outcome variable balance, enhancing model accuracy; pair-wise model performance comparisons offer unique insights.
- Resampling methods are crucial for initial analysis and should be refined to handle data imbalances in future work.
- Inclusion of random forests and other algorithms with text mining addresses vital social issues; iterative and incremental work unites course data mining aspects.

# IST 652

## Scripting for Data Analysis

World Happiness Analysis:  
Investigating Socio-economic and  
Demographic Influences



# IST 652: Scripting for Data Analysis

## *Key Analytic Questions - KAQS*

- How do social, economic, and demographic factors affect a nation's happiness ranking?
- What data can show us more about the components of a country's overall satisfaction and happiness?
- The goal is to dive deep into the factors that contribute to a nation's perceived happiness and understand how different aspects of a country's statistics can influence its happiness ranking.

# IST 652: Scripting for Data Analysis

## *Methodology*

1. Setup and Data Merging: Setting up the environment and merging the World Happiness Report data with the countries' additional statistics.
2. Data Preprocessing: Cleaning and preprocessing the data for analysis, handling missing values, and ensuring consistent data types.
3. Exploratory Data Analysis: Providing an overview of the data
4. Factor Analysis: Investigating which socio-economic, demographic, and other factors have the strongest relationship with a country's happiness ranking.
5. Regional Comparisons: Comparing and contrasting happiness scores and key influencing factors among different world regions.
6. Predictive Modeling: Building a regression model to predict happiness scores based on selected factors.
7. Conclusion and Future Recommendations: Summarizing findings, discussing implications for policy-makers and leaders, and suggesting areas for future research.



# IST 652: Scripting for Data Analysis

## *Setup, Data Merging and Preprocessing*

- **Environment Preparation**
  - Establish an environment using Python and essential libraries (e.g., Pandas, NumPy).
  - Ensure seamless execution of analysis tasks.
  - Employ Jupyter Notebooks for interactive exploration and documentation.
- **Data Cleaning and Preprocessing**
  - Clean and standardize data variables to ensure consistency.
  - Handle missing values by employing appropriate techniques (e.g., imputation, removal).
  - Verify and enhance data quality to prevent bias in analysis.
- **Data Merging and Integration**
  - Merge WHR data with country-specific statistics.
  - Utilize shared identifiers (e.g., country names, codes) to facilitate accurate connections.
  - Harness Pandas' merging capabilities to create a consolidated dataset.

# IST 652: Scripting for Data Analysis

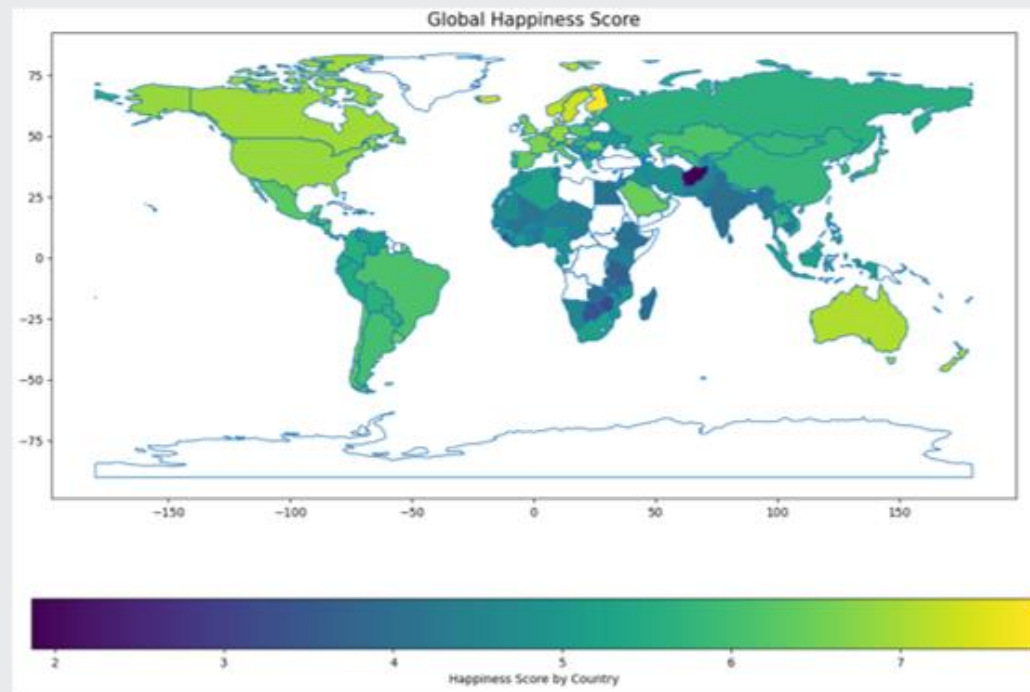
## *Ranking of the Top 5 World's Happiest and Least Happy Countries*

### Top 5 Happiest Countries

1. Finland
2. Denmark
3. Iceland
4. Israel
5. The Netherlands

### Bottom 5 Happiest Countries

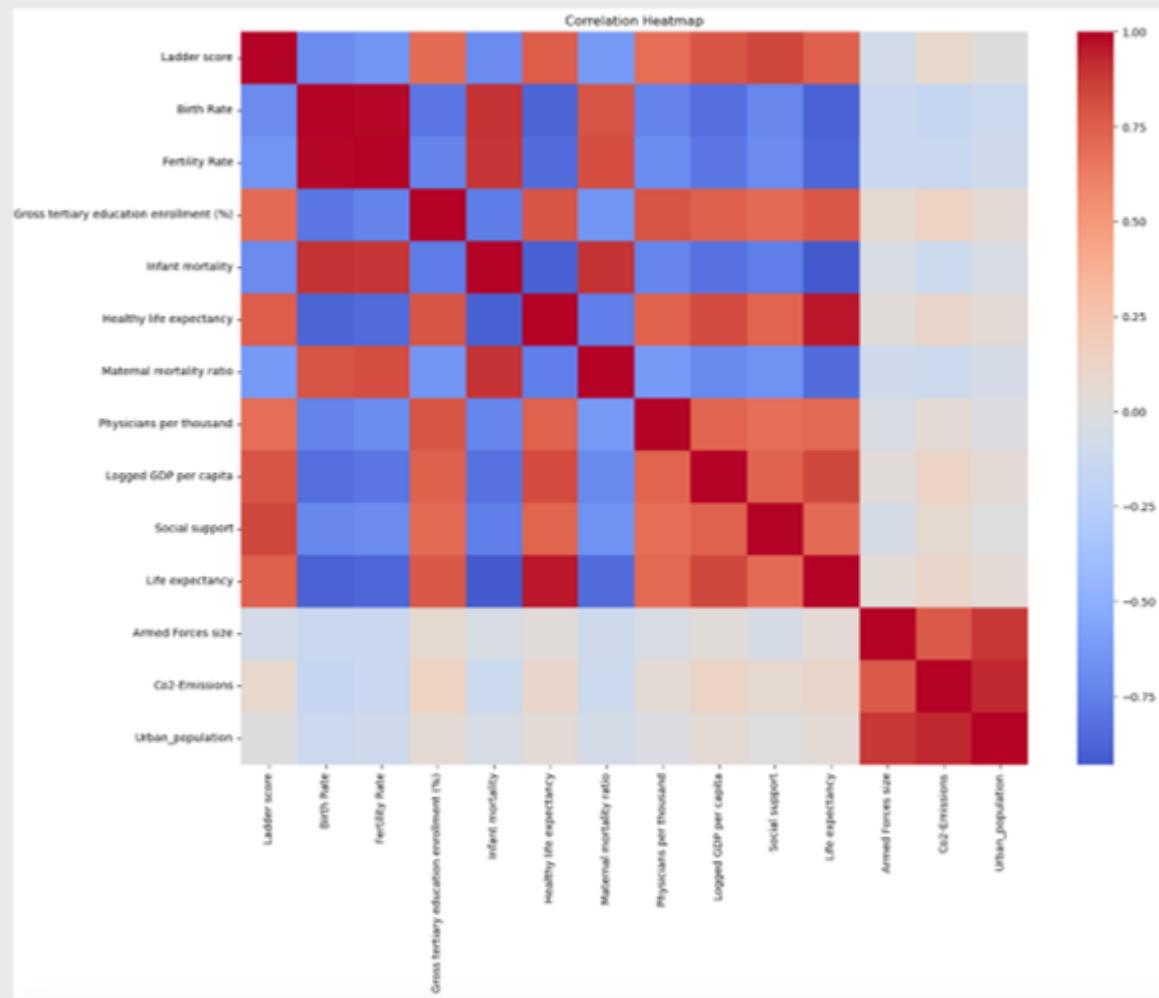
123. Botswana
124. Zimbabwe
125. Sierra Leone
126. Lebanon
127. Afghanistan



# IST 652: Scripting for Data Analysis

## *Factor Analysis*

- Correlation Heatmap
  - Strong Positive Correlations with Ladder Score: Logged GDP per Capita, Social Support, Healthy life expectancy
  - Strong Negative Correlations with Ladder Score: Birth Rate, Infant Mortality, Fertility Rate



# IST 652: Scripting for Data Analysis

## *Reflection*

- Our research analyzed how socio-economic and demographic factors relate to country happiness ranking.
- Findings show that GDP per capita, social support, and individual freedom all play a role in a nation's happiness ranking. These elements reveal the importance of social and political aspects of well-being.
- The regression model explains 83.4% of the "Ladder Score" variability, with significant predictors being "Logged GDP per capita," "Social Support," and "Freedom to make life choices."



# IST 691

## Deep Learning in Practice

### Essay Classification Model



# IST 691: Deep Learning in Practice

## *Goal*

- Build a highly accurate model for automated essay classification, with applications in educational assessment and AI evaluation.
- Classify essays as human or AI authored
- Enhance understanding of LLM capabilities

# IST 691: Deep Learning in Practice

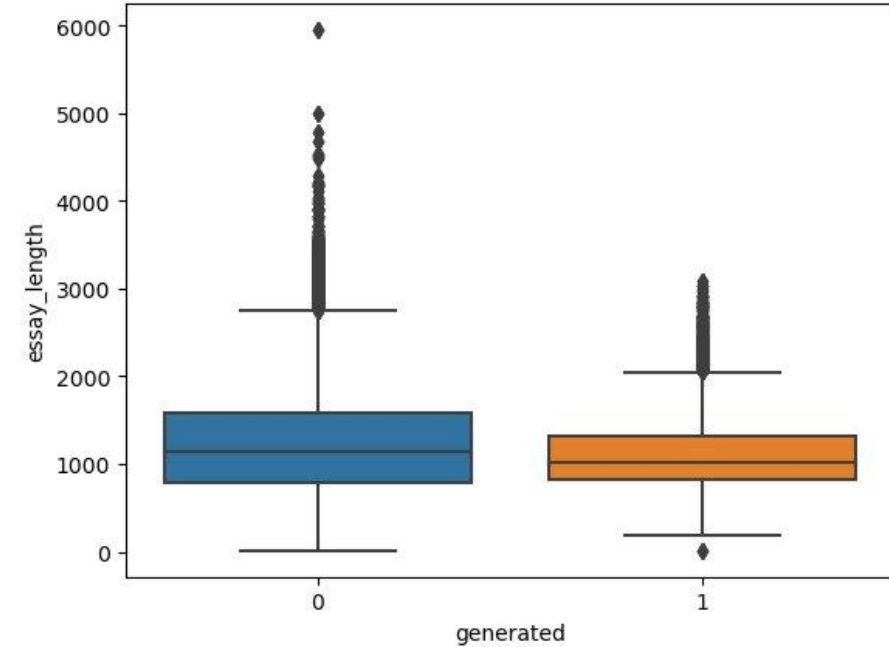
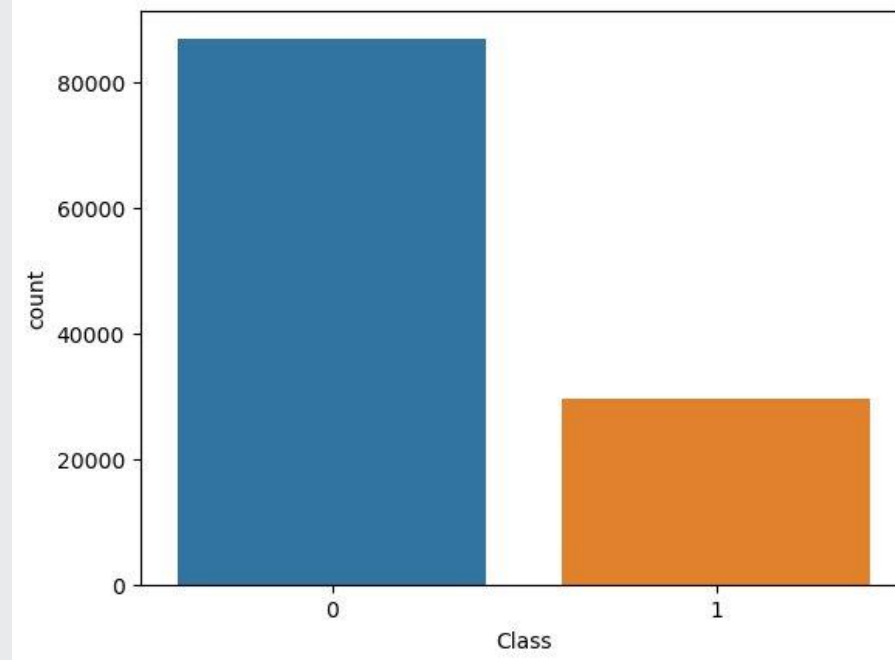
## *Data Preprocessing*

- Merged Kaggle and Dr. Cat datasets into one dataset
- Ensured equal number of human and LLM essays
- Studied essay length distribution
- Changed labels to 0 for human, 1 for LLM

# IST 691: Deep Learning in Practice

## *Balanced Classes*

- Introducing the Dr. Cat datasets provided more balance between AI-generated and human essays, but there are still clearly gaps between the two.

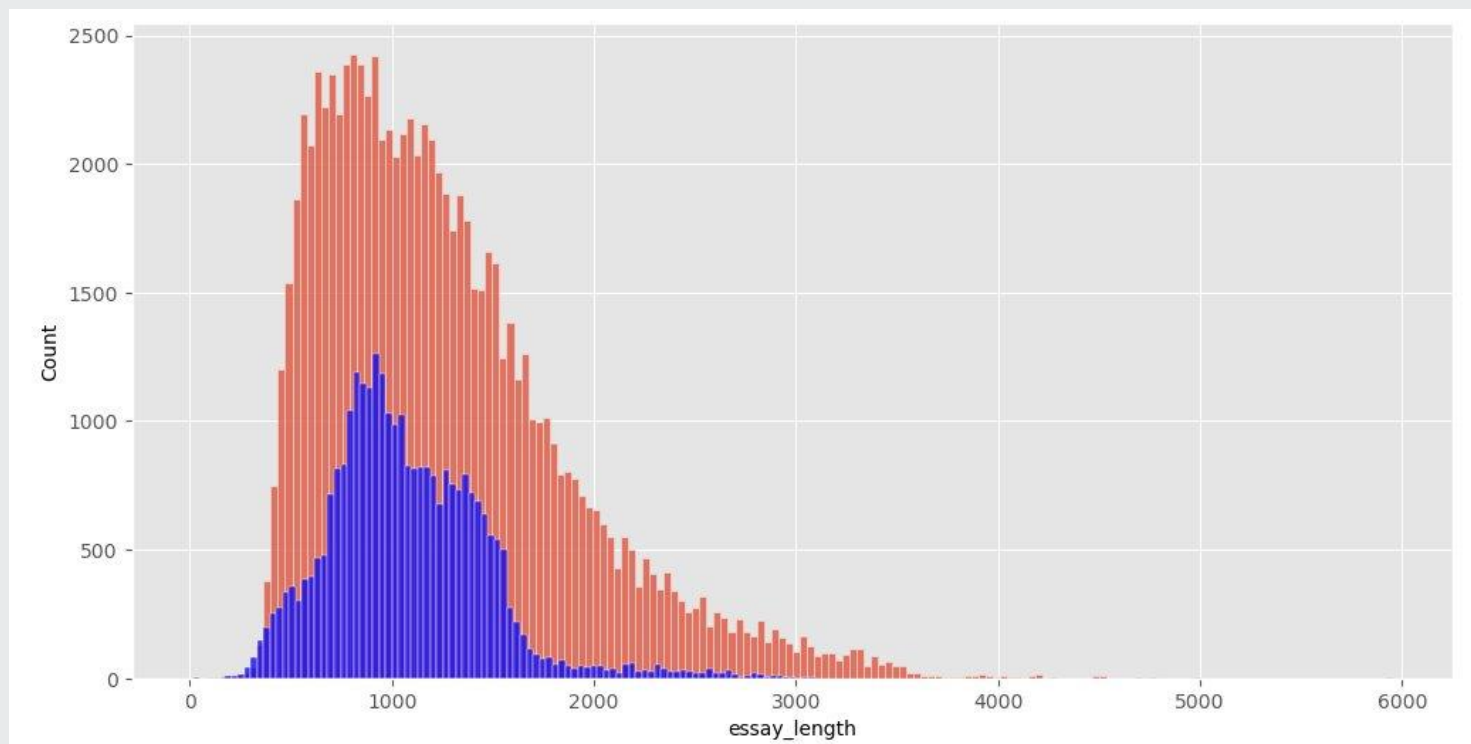




# IST 691: Deep Learning in Practice

## *Analyzing Essay Length*

- There were lower counts of AI-Generated essays, and they tended to be shorter.



# IST 691: Deep Learning in Practice

## *Model Development*

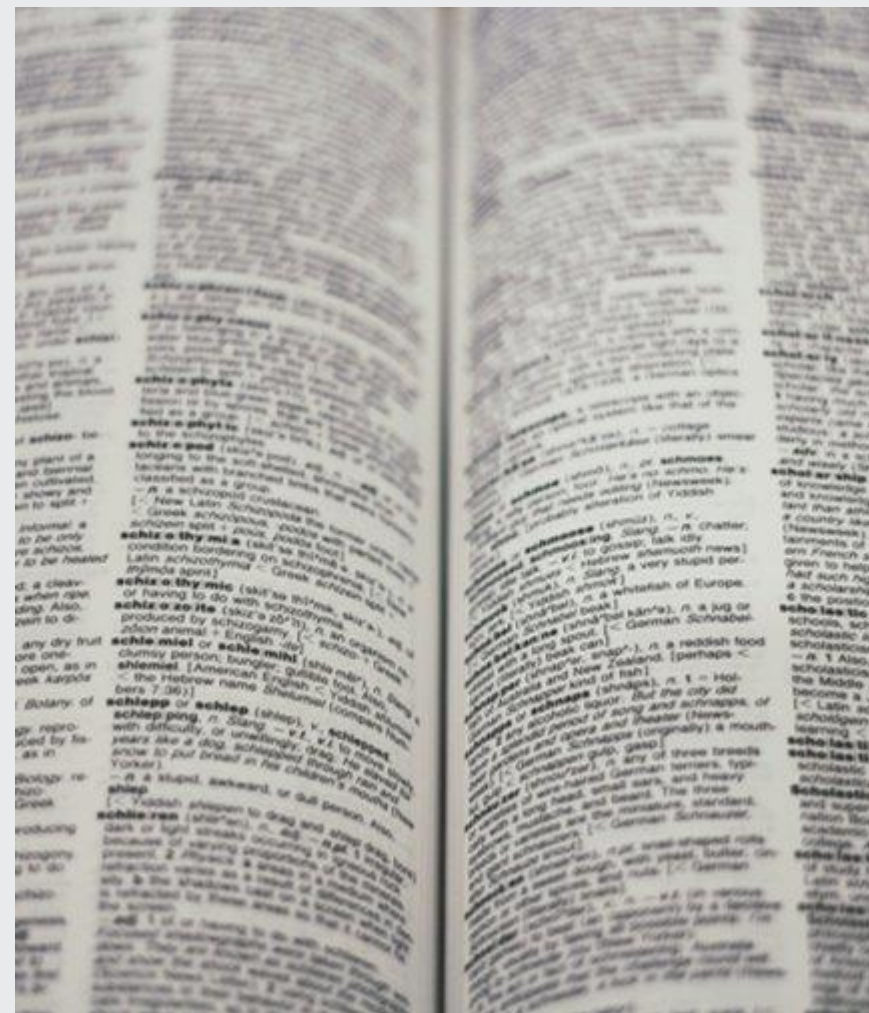
- The GRU model uses GloVe word embeddings to understand the meaning and context of words in the essays. This enhances the model's natural language processing capabilities for accurate essay classification.



# IST 691: Deep Learning in Practice

## Why GRU?

- Sequential Data Handling
- Capturing Context
- Feature Extraction
- Adaptability
- Performance on Large Datasets



# IST 691: Deep Learning in Practice

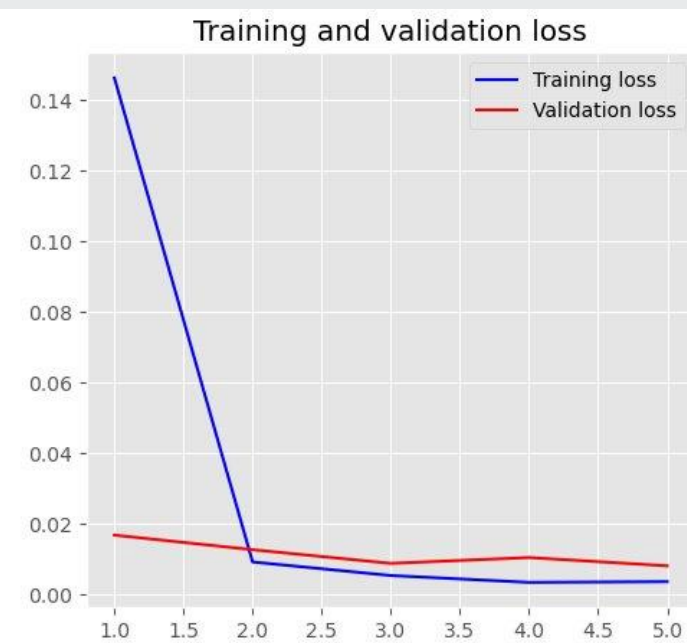
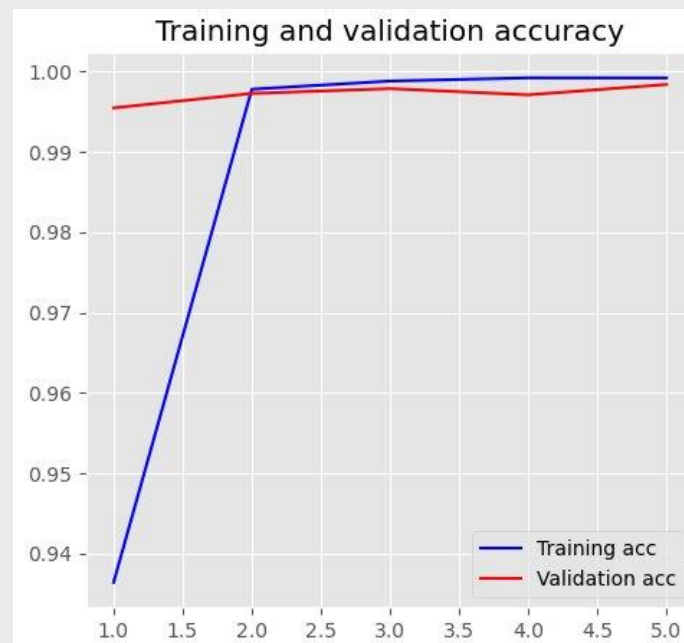
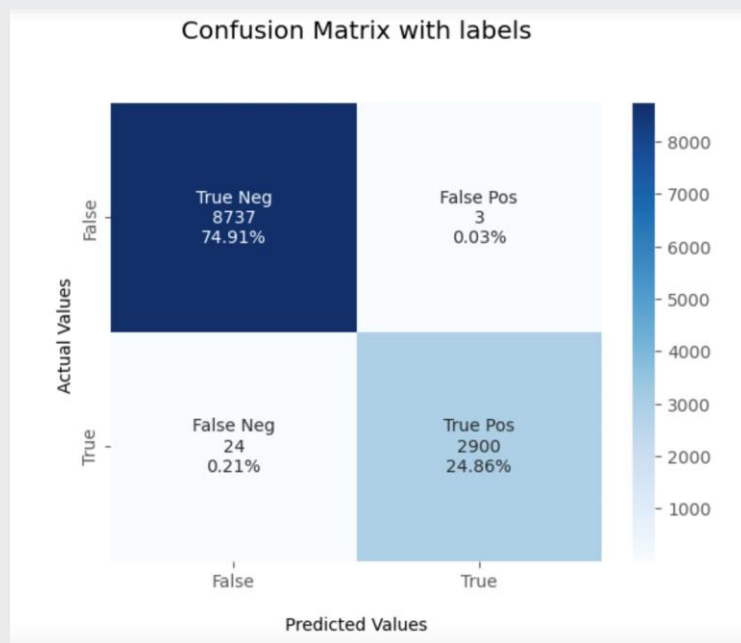
## *Model Training*

- The model demonstrates high accuracy on both training and validation data, indicating its reliability for essay classification.
- Model trained on 80% of data, validated on 10%, and tested on 10%.
- Achieved high accuracy with 99.98% on training data and 99.84% on validation data.



# IST 691: Deep Learning in Practice

## Results



# IST 691: Deep Learning in Practice

## *Hyperparameter Tuning*

- Tuning Process
  - Used Keras Tuner to find optimal hyperparameters like units in dense layer and learning rate
- Optimization Results
  - Optimal dense layer units: 448, optimal learning rate: 0.001

# IST 691: Deep Learning in Practice

## *Final Model Training*

- Train Final Model
  - Train the final model using the optimal hyperparameters and model architecture determined during hyperparameter tuning.
- Select Best Epoch
  - Evaluate the model after each training epoch and select the epoch that results in the highest validation accuracy.

# IST 691: Deep Learning in Practice

## *Key Insights and Future Directions*

- The project successfully developed a highly accurate essay classification model with promising applications, and future work will focus on optimizations.
- The model has potential real-world applications like automated grading of student essays or evaluating large language models.
- High accuracy model for essay classification
  - Developed a model with training accuracy of 99.96% and validation accuracy of 99.79% for classifying essays.



# Conclusion

- Syracuse University's Information Studies School equips students with skills in data collection, management, and analysis, utilizing a range of data science techniques for actionable insights.
- The program enhances a multifaceted approach to tackle both structured and unstructured data challenges of growing complexity.
- Strategies developed within the program enhance organizational efficiency.
- Emphasis on transparency, reproducibility, and ethics in data management to maintain an organization's analytical integrity.
- Graduates of the Applied Data Science program are prepared to address diverse problems and effectively communicate findings to stakeholders and business experts.



Thank You!

Template from:

<https://www.syracuse.edu/about/brand/visual-identity/powerpoint-templates/>