Student Name: …

ADTA 5340: Discovery and Learning with Big Data

**Final Project Report**

# PART I:

**Strategy for Implementing Machine Learning in Future Tech**

***1. Company Overview:***

The Company Name is Future Tech. Its industry is Information Technology and Software Development. It is medium-sized, with a workforce of around 500 employees. It offers software solutions for various industries, including CRM systems, and cybersecurity software

***2. Competitive Landscape:***

Competitors include major tech companies offering similar software solutions with machine learning capabilities.

Their competitors leverage machine learning for automation and security enhancements.

***3. Objectives:***

1. Increase product offerings with machine learning capabilities to stay ahead of competitors.
2. Improve internal processes, such as software development and customer support, using machine learning algorithms to surpass the competitors.

***4. Strategy Overview:***

The technology selection is to adopt an approach using open-source frameworks like TensorFlow and Scikit-learn.

It's focus Areas being :

1. Product Enhancement: Integrate machine learning algorithms into existing software products for predictive analytics and recommendation systems.
2. Internal Operations in the company: Develop AI-driven tools for code review automation, and customer sentiment analysis.
3. Customer Support: Implement chatbots and natural language processing (NLP) for better personalized customer interactions.

Human Capital Development:

1. Conduct training programs and workshops for software engineers and data scientists on machine learning techniques and best practices.

***5. Implementation Plan:***

Phase 1: Needs Assessment (2 weeks)

1. Identify areas for machine learning integration based on customer feedback.
2. Evaluate existing data infrastructure and identify data sources for model training.
3. Define project scope, objectives, and success criteria.

Phase 2: Model Development (6 weeks)

1. Develop models to help in some specific use cases (e.g., predictive maintenance).
2. Collaborate with domain experts and stakeholders to refine models and validate results from the model.
3. Ensure data load and user scalability and performance optimization for production deployment.

Phase 3: Integration and Testing (4 weeks)

1. Integrate machine learning models to software products and company-internal tools.
2. Conduct serious testing to check model accuracy and performance in real-world scenarios.

Phase 4: Deployment and Monitoring (Ongoing)

1. Deploy models into production environments with monitoring capabilities for performance, data quality, and model drift.
2. Implement feedback loops and continuous improvement processes for retraining the model and updating it.
3. Establish metrics for tracking the impact of machine learning on product performance and customer satisfaction.

***6. Budget Allocation:***

1. Allocate resources for data acquisition, model development tools, cloud infrastructure (For example AWS), and talent acquisition.
2. Budget for ongoing maintenance, monitoring tools, and capacity expansion as machine learning initiatives scale.

***7. Risk Management:***

1. Address security concerns through robust encryption, access controls, and compliance with data regulations.
2. Implement model transparency measures to mitigate risks related to bias, fairness, and ethical AI practices.

***8. Communication and Stakeholder Engagement:***

1. Regularly communicate progress updates and milestones to the executive team, product managers, development teams, and other stakeholders.
2. Foster a culture of collaboration and knowledge sharing among teams working on machine learning projects.

# PART II: *# Report on IoT Agriculture Dataset*

The name of this Dataset is IoT Agriculture Dataset 2024.Downloaded from Kaggle. The link to the dataset is <https://www.kaggle.com/datasets/wisam1985/iot-agriculture-2024>

The dataset is composed of 14 columns, namely :

1. date (datetime64): The date and time the measurements were recorded.
2. temperature (int64): The recorded temperature in degrees Celsius.
3. humidity (int64): The percentage of humidity in the environment.
4. water\_level (int64): The water level as a percentage.
5. N (int64): The nitrogen level in the soil, scaled from 0 to 255.
6. P (int64): The phosphorus level in the soil, scaled from 0 to 255.
7. K (int64): The potassium level in the soil, scaled from 0 to 255.
8. Fan\_actuator\_OFF (float64): Indicator for the fan actuator if it is off (0 or 1).
9. Fan\_actuator\_ON (float64): Indicator for the fan actuator if it is on (0 or 1).
10. Watering\_plant\_pump\_OFF (float64): Indicator for the plant watering pump if it is off (0 or 1).
11. Watering\_plant\_pump\_ON (float64): Indicator for the plant watering pump if it is on (0 or 1).
12. Water\_pump\_actuator\_OFF (float64): Indicator for the water pump actuator if it is off (0 or 1).
13. Water\_pump\_actuator\_ON (float64): Indicator for the water pump actuator if it is on (0 or 1).

***Quality of the data:***

***Missing Values:***

The dataset of 37922 records is mostly clean with only one attribute having missing values. The date attribute has 2 missing values.

***Handling Missing Values***

Since date is the only column with missing values, and is a datetime64 type, the values are unique. Hence they cannot be imputed using mean or mode methods like numerical attributes. The approach taken to solve this was identifying missing values based on the surrounding data points. This was done by taking the average time difference between adjacent dates(timestamps) and filling the values.

***Summary of Machine Learning Project***

The dataset can be used to predict the status of actuators in a smart greenhouse based on environment conditions and soil-nutrient levels. The target variables being

1. Fan\_actuator\_OFF
2. Fan\_actuator\_ON
3. Watering\_plant\_pump\_OFF
4. Watering\_plant\_pump\_ON
5. Water\_pump\_actuator\_OFF
6. Water\_pump\_actuator\_ON

# PART III:

***1. Preprocessing the Dataset***

The dataset requires preprocessing for several reasons:

1. Handling missing values: The jupyter script first checks if there are missing values and drops rows with missing values, ensuring the dataset is clean.
2. Handling abnormal values: The jupyter script first checks if there are abnormal values For example ‘3+’ and preprocesses the data by replacing these values with other numeric counterparts.
3. One-hot encoding: The 'Sex' column is one-hot encoded to convert categorical data into numerical format for modeling.

***2. Model Selection***

Linear Regression model has been selected for predicting the age of abalones.

This model used is Linear Regression. This is because

1. Linear Regression takes/assumes a linear relationship between the features and target variable, which is a reasonable assumption for some datasets.
2. The model is also simple, easy to understand and easy to interpret.

***3. Building the Model***

The model was constructed using the Linear Regression algorithm.

It aims to find the best linear relationship between the features X and the target variable y..

***4. Training the Model***

The model is trained with the training data, to learn patterns and relationships between the features (X) and the target variable (y).

It uses model.fit(X\_train\_scaled, y\_train) to train the Linear Regression model.

***5. Making up Two New Records***

Two new data-records with the attribute values are created to demonstrate how the model predicts the age from the records..

The attribute names include Length, Diameter, Height, etc.

***6. Predicting the Age of Abalones***

Explanation: The model predicts the age of abalones for the new records based on the trained Linear Regression model.

***7. Evaluating the Model***

Explanation: The model is evaluated using metrics like Mean Squared Error, Accuracy, and R-squared on the test set to assess its performance and predictive capability.

1. The Mean Squared Error is : 4.891232447128581
2. The Accuracy: 0.5481628137889262
3. The R-squared (Coefficient of Determination): 0.5481628137889262

***8. Interpretation of Prediction Results***

1. The MSE (4.89) indicates that the squared difference between the actual ages of abalones and the predicted ages by the model is around 4.89 years.
2. The R-squared value of 0.54/0.55 (or 55%) shows that the model explains 55% of the variance in the age of abalones based on the features included in the dataset.

# PART IV:

## EDA: Data Discovery

The dataset name is Adult Salary. The dataset has 48841 records, and 15 columns. The dataset features include, Age, Employment type,Education and Education\_num, which is the highest level of education in numerical form. Other features include marital, Occupation, relationship, race, sex, Capital gain and loss, Weekly\_hours, number of hours worked per week, Country of origin and lastly Income.

Data Types of the features include : int64 for Age,Fnlwgt,EducationNum, Capital\_gain, Capital\_loss and Weekly-hours. The rest of the features are objects.

## EDA: Pre-Processing Data

The dataset has no null values. However, there are instances where empty fields are represented by ?. There are 1049 instances, where ? is in the dataset across several features. To remove these null values, I opted to replace the character “?” with NaN. I also dropped rows with empty values. This ensures that clean data is used in training the model.

Categorical variables were encoded using the LabelEncoder pre-processing import. For each categorical column, label encoder was applied to transform into numerals/numerical values.

Features to be used were selected based on their importance to the target variable y(Income feature).

Chosen features include Age, employment\_type,Education, Marital\_status,Occupation,Relationship,Race, Sex,Capital gains and Losses, weekly hours and lastly Country.

## Model Planning

The model selected is Random Forest Classifier.

Random Forest has been chosen due to its ability to handle categorical features, non-linear relationships, and provide good predictive performance.

In addition to this, RandomForest has several advantages

1. Ensemble learning- This helps to reduce overfitting, and generalization performance.
2. It is also robust to outliers and noise in the data, making it resilient to potential data anomalies.

## Model Building

The model was constructed with the following parameters:

1. Number of Estimators=100
2. Random State=42 (To ensure consistent results across different runs of the model).

Standard Scaler was applied to the features before training the model. This step ensures that all features are on the same scale, which is important for many machine learning algorithms.

The dataset was split into training and testing sets using 80/20 split ratio. I.e 80% of the data for training the model. 20% reserved for evaluating the model`s performance on unseen data.

## Reporting Results

Supervision learning used to train the Random Forest classifier model. The model`s performance was evaluated using the accuracy and cross validation scores.

The model achieved a testing accuracy of approximately 51.35%, indicating its ability to predict income levels correctly on unseen data.

The model`s cross-validation scores across 10 folds are :

[0.515, 0.504, 0.519, 0.515, 0.506, 0.513, 0.508, 0.516, 0.503, 0.514]

The mean accuracy across these folds is approximately 51.13%. This shows the model`s consistent performance across different subsets.

# PART V:

## EDA: Data Discovery

The dataset name is car\_evaluation. The dataset has 1728 records, and 7 columns. Initially, the dataset had no column names, so I had to allocate feature names to the columns. The dataset features include, Price, Maintenance, Doors, Passengers, Luggage, Safety and Evaluation.

Data Types of the features are objects.

## EDA: Pre-Processing Data

The dataset has no null values. This was determined by calculating the sum of missing values. I have dropped missing values and duplicate values. The cleaned dataset still has 1728 records and 7 columns.

Categorical variables were encoded using the LabelEncoder pre-processing import. For each categorical column, label encoder was applied to transform into numerals/numerical values.

## Model Planning

The model used is KMeans Clustering.

KMeans is suitable to identify natural groupings within the dataset as it is able to split the data into several distinct clusters. The number of clusters was set to 4.

## Model Building

The model was constructed with 4 clusters and a random state of 42. Dimensionality reduction was conducted using PCA inorder to reduce the number of features while retaining most of the important information. PCA also transforms original into a lower dimensional space allowing us to visualize the data in a scatter plot.

## Reporting Results

The KMeans clustering model successfully identified 4 cluster centers showing distinct characteristics for price, maintenance, doors,Passengers, luggage and safety features.

The KMeans clustering model predicted the cluster for the new records as [1 1], assigning both records to cluster 1.

The silhouette score for the model is approximately 0.115,indicating a moderate degree of separation between clusters and quality of the clustering.

# PART VI `Linear Regression vs. Decision Tree (CART) Regression`:

## EDA: Data Discovery

The dataset name is appointment\_dataset. The dataset`s link is here <http://www.kaggle.com/joniarroba/noshowappointments> .

The dataset has 110526 records, and 14 columns. Initially, the dataset had no column names, so I had to allocate feature names to the columns. The dataset features include, PatientId,AppointmentID,Gender,ScheduledDay, AppointmentDay,Age,Neighbourhood,Scholarship,Hipertension,Diabetes,Alcoholism,Handcap,SMS\_received,No-show.

Data Types of the features are as follows :

1. float64 - PatientId
2. Int64-AppointmentId,Age,Scholarship,Hypertension,Diabetes,Alcoholism,Handicap,SMS\_received
3. object- Gender,ScheduledDay,AppointmentDay,Neighbourhood,No-show.

## EDA: Pre-Processing Data

The dataset has no null values. This was determined by calculating the sum of missing values. I have dropped missing values and duplicate values. The cleaned dataset still has 110526 records and 14 columns.

I dropped unnecessary columns like PatientId, AppointmentID, Neighbourhood, as they don` t have any significant effect on the model.

Categorical variables were encoded using the LabelEncoder pre-processing import. For each categorical column, label encoder was applied to transform into numerals/numerical values.

I split the dataset into features(X) by dropping No\_show and target variable(No-show). The model will be used to try to determine if a patient will honor the appointment.

## Model Planning

Two models were used, LinearRegression and DecisionTreeRegressor. The goal was to compare appropriateness of model to the appointment dataset.

Linear regression predicts numerical values, in this case, the predictions are large numerical values. These predictions don't directly represent whether a patient will show up for their appointment or not.

Decision tree regression on the other hand predicts categorical values, often in the form of classes or categories

## Model Building

The DecisionTree Regressor model was constructed with a random state of 42.

R Squared scores of the both models were calculated after the data was fitted in the models.

## Reporting Results

The r squared score for Linear Regression is approximately 0.04, while for DecisinTree is approximately -0.69.

A new data set with two records was introduced. The date in the records was converted to Unix timestamp format for model compatibility. The features were encoded and fed into the models.

Predictions for Linear Regression is : [15730809.66850501 15730727.16734266], while predictions for Decision Tree Regression is [1,1].

Linear Regression predictions don't directly represent whether a patient will show up for their appointment or not.while Decision Tree predicts that in both cases the patient will show up for the appointment.

# PART VI Logistic Regression vs. K-Nearest Neighbors

## EDA: Data Discovery

The dataset name is appointment\_dataset. The dataset`s link is here <http://www.kaggle.com/joniarroba/noshowappointments> .

The dataset has 110526 records, and 14 columns. Initially, the dataset had no column names, so I had to allocate feature names to the columns. The dataset features include, PatientId,AppointmentID,Gender,ScheduledDay, AppointmentDay,Age,Neighbourhood,Scholarship,Hipertension,Diabetes,Alcoholism,Handcap,SMS\_received,No-show.

Data Types of the features are as follows :

1. float64 - PatientId
2. Int64-AppointmentId,Age,Scholarship,Hypertension,Diabetes,Alcoholism,Handicap,SMS\_received
3. object- Gender,ScheduledDay,AppointmentDay,Neighbourhood,No-show.

## EDA: Pre-Processing Data

The dataset has no null values. This was determined by calculating the sum of missing values. I have dropped missing values and duplicate values. The cleaned dataset still has 110526 records and 14 columns.

I dropped unnecessary columns like PatientId, AppointmentID, Neighbourhood, as they don` t have any significant effect on the model.

Categorical variables were encoded using the LabelEncoder pre-processing import. For each categorical column, label encoder was applied to transform into numerals/numerical values.

I split the dataset into features(X) by dropping No\_show and target variable(No-show). The model will be used to try to determine if a patient will honor the appointment.

## Model Planning

Two models were used, LogisticRegression and KNeighborsClassifier. The goal was to compare appropriateness of model to the appointment dataset.

Linear regression predicts numerical values, in this case, the predictions are large numerical values. These predictions don't directly represent whether a patient will show up for their appointment or not.

Decision tree regression on the other hand predicts categorical values, often in the form of classes or categories

## Model Building

The LogisticRegression model was constructed with a random state of 42, while KNeighborsClassifier was constructed with “n\_neighbors=3”.

R Squared scores of the both models were calculated after the data was fitted in the models.

## Reporting Results

The accuracy score for Linear Regression is approximately 0.796, while for KNeighborsClassifier is approximately 0.75.

A new data set with two records was introduced. The date in the records was converted to Unix timestamp format for model compatibility. The features were encoded and fed into the models.

Predictions for LogisticRegression is : [1 1], while predictions for KNeighborsClassifier is [0 0].

These predictions show the different approaches and outcomes of the two models. Logistic Regression is a linear model which estimates probabilities, making it suitable for binary classification tasks like predicting appointment show-ups. K-Nearest Neighbors, on the other hand, is a non-linear classifier that classifies data points based on their nearest neighbors, which may lead to different predictions compared to Logistic Regression.

After K-cross validation, with K being 10, the Mean accuracy for LogisticRegression is 0.79, while for K-nearest neighbor is 0.74.