**INTERNSHIP PROJECT REPORT**

**Task 2: Machine Learning with PySpark**

**Project Title: Predicting AI Impact on Jobs Using Decision Tree & Random Forest**

**Objective:**

The objective of this task was to apply machine learning techniques using PySpark to predict the **level of AI impact on various job roles** (Low, Moderate, or High), based on job-related features like salary, education, job openings, risk of automation, and more.

**Dataset Description:**

* **Name**: AI Impact on Jobs
* **Total Rows**: ~30,000
* **Target Column**: AI Impact Level
* **Feature Columns**:
  + Median Salary (USD)
  + Required Education
  + Experience Required (Years)
  + Job Openings (2024)
  + Projected Openings (2030)
  + Remote Work Ratio (%)
  + Automation Risk (%)
  + Gender Diversity (%)
  + Job Title
  + Industry

**Tools & Technologies Used:**

* Google Colab
* PySpark (Spark MLlib)
* Python
* Pandas (for CSV export)
* DecisionTreeClassifier
* RandomForestClassifier

**Step-by-Step Workflow:**

1. **Data Loading & Cleaning**
   * Loaded dataset using Google Colab
   * Checked missing values and data types
   * Cleaned and explored dataset
2. **Data Exploration**
   * Used .describe(), .select(), .distinct() to understand distributions
   * Found that some columns had many unique categories (e.g., Job Title)
3. **Data Preparation**
   * Applied StringIndexer to encode text columns (e.g., Job Title, Industry)
   * Used VectorAssembler to combine numeric and encoded columns into one features column
4. **Model Training**
   * Split dataset into 80% training and 20% testing
   * Trained both:
     + **Decision Tree Classifier**
     + **Random Forest Classifier**
5. **Evaluation**
   * Used MulticlassClassificationEvaluator to measure accuracy

**Results:**

| **Model** | **Accuracy** |
| --- | --- |
| Decision Tree | 33.46% |
| Random Forest | 32.32% |

Despite tuning and using Random Forest, the accuracy remained low due to:

* High-cardinality features (639 unique job titles)
* Imbalanced classes in the target column
* Overlapping feature values across classes

**Conclusion:**

Although the model accuracy was low, the project successfully demonstrated the **end-to-end machine learning pipeline using PySpark**, including:

* Data cleaning and encoding
* Feature engineering
* Model training and evaluation
* Identification of real-world challenges in AI-driven job predictions

This task improved my understanding of **PySpark MLlib**, working with big data, and handling challenges like class imbalance and categorical overfitting.