TABLE OF CONTENTS

ABSTRACT

The rapid advancement of Artificial Intelligence (AI), robotics, and automation technologies is transforming the global workforce at an unprecedented pace. Many traditional job roles are becoming increasingly susceptible to automation, creating an urgent need for tools that can forecast these changes and guide individuals toward future-proof careers. This project presents an AI-based Career Automation Prediction and Recommendation System designed to predict the likelihood and timeframe of job automation while recommending key skills for professional adaptability.

Using datasets that integrate job market insights, skill trends, and automation risk factors, the system employs machine learning models to estimate the number of years before a specific job may be automated. It further classifies each role into low, moderate, or high-risk categories based on automation vulnerability. When a career is identified as high-risk, the model dynamically recommends alternative skill sets or emerging career paths aligned with industry trends—empowering users to upskill and transition effectively.

The project also establishes a comprehensive mapping between job titles and required skills, leveraging data preprocessing, statistical analysis, and visualization techniques to highlight patterns in technological relevance. This AI-driven framework not only enhances career resilience and long-term employability but also serves as a strategic tool for educators, workforce planners, and organizations adapting to the evolving future of work.

By combining predictive analytics with intelligent recommendation, the system contributes to building an informed, adaptable, and future-ready workforce capable of thriving in the age of automation.

CHAPTER-1

INTRODUCTION

The rapid growth of Artificial Intelligence (AI) and automation technologies is reshaping industries and job markets worldwide. As machines become more capable, many job roles face the risk of being either partially or fully automated. This creates both opportunities for innovation and challenges for workers who must adapt to new skill demands.

Understanding which jobs are at risk and what skills can help individuals remain employable has become essential. Career planning today requires informed decisions backed by data, rather than speculation. In this context, prediction and recommendation systems are valuable tools for both students and professionals.

This project focuses on predicting the automation risk of job roles based on job title, skills, industry, company size, and work environment. The system then recommends new or related jobs and skills to help users remain competitive. The application is implemented as an interactive Streamlit web interface for easy accessibility.

By combining machine learning-based prediction and skill recommendation, this system guides users toward future-ready career choices. It acts as a bridge between current job capabilities and industry skill trends shaped by AI evolution.

CHAPTER-2

Methodology

This project follows a structured methodology that involves data collection, preprocessing, feature engineering, model development, evaluation, and deployment. The methodology ensures reliable automation risk prediction and meaningful skill recommendation tailored to real-world job market conditions.

1. Data Collection and Understanding:

The dataset used in this project consists of job titles, industries, organization sizes, skill requirements, remote work feasibility, and automation risk labels. The data was sourced from publicly available job market insight reports and curated occupation datasets. The goal was to capture a diverse set of job categories across both technical and non-technical fields.

Each job entry included two major types of information:

1.Categorical attributes such as Industry, Company Size, and Remote-Friendly.

2.Text-based attributes such as Job Title and Required Skills.

Since automation risk is influenced not only by job tasks but also by domain and organizational context, combining structured and unstructured features was necessary.

2. Data Preprocessing:

Preprocessing was carried out to clean the dataset and make it suitable for machine learning modeling. The steps included:

Handling Missing Values

Missing values were replaced with neutral tokens such as "Unknown" for categorical columns and an empty string "" for skill and title text fields. This ensured that no row was lost during training.

Text Normalization

Text fields often contained:

- 1.Inconsistent naming

- 2.Synonym variations
- 3.Mixed case formatting

To resolve this:

- 1.All text was converted to lowercase.
- 2.Special symbols were removed.
- 3.Repeated commas were replaced with standardized separators.

Target Encoding

The automation risk label (Low, Medium, High) was encoded numerically using **LabelEncoder**. This allowed the machine learning model to interpret the target variable.

Through these steps, a clean and uniform dataset was prepared, which improved model accuracy and performance reliability.

3. Feature Engineering

The dataset contained two different types of inputs — categorical and textual — requiring different feature processing techniques.

Categorical Feature Encoding:

Categorical attributes (Industry, Company Size, Remote Friendly) were encoded using One-Hot Encoding, enabling the model to treat each category as an independent feature without imposing numerical biases.

Text Feature Representation Using TF-IDF:

Job titles and required skills hold strong predictive power because they directly reflect job complexity and task structure. To capture the semantic importance of words, TF-IDF (Term Frequency – Inverse Document Frequency) was applied.

TF-IDF ensures:

- 1.Common but less informative words are de-weighted.
- 2.Unique and meaningful job-skill expressions are highlighted.
- 3.This helped the model learn relationships such as:
- 4.Jobs requiring machine learning, statistics, and programming → lower automation risk.

5.Jobs based on repetitive manual operations → higher automation risk.

4. Model Selection and Training

Two machine learning models were developed and compared:

Random Forest Classifier:

- Works as an ensemble of decision trees.
- Handles mixed feature types effectively.
- Achieved ~88% validation accuracy.
- Reduced overfitting through feature randomness and bootstrap sampling.

XGBoost Classifier

- Gradient boosting model with strong regularization
- Given more importance to high-informational features.
- Achieved ~91% accuracy and ~0.90 F1 score.
- Performed better for scalable and generalizable prediction.

Pipeline Integration

A Pipeline was created to combine preprocessing and modeling into a single flow. This ensures:

- Cleaner implementation
- Avoidance of data leakage
- Reusability of final trained model

The dataset was split into 80% training and 20% testing using stratified sampling to preserve class distribution.

5. Model Evaluation

The model was evaluated using:

- Accuracy
- Precision
- Recall
- F1-Score

The evaluation confirmed:

- High-risk job categories were classified with strong precision.

- Medium-risk jobs required more balancing due to skill overlaps, which was improved using class-balanced Random Forest and XGBoost tuning.
- Low-risk jobs consistently showed strong recall, confirming reliability for long-term automation safety prediction.

Confusion matrix analysis ensured that the model was not biased toward any risk class.

6. Skill Recommendation System Design

The recommendation engine was developed by mapping:

- Job titles to skills using grouped categorical aggregation.
- Similarity-based matching between input skill sets and alternative roles.

A dictionary structure was generated where each job role was linked to a list of relevant skill keywords. When the system analyzes the user's input skills, it searches for closest matches and suggests:

- Better aligned roles
- Future-proof skills
- AI-driven tools and emerging technologies to learn

This ensures the system goes beyond prediction and provides actionable career guidance.

7. Deployment Using Streamlit

To make the system accessible to users, a Streamlit web application was developed.

Frontend Input Fields:

- Job Title
- Industry
- Company Size
- Remote Friendly
- Required Skills

Displayed Output:

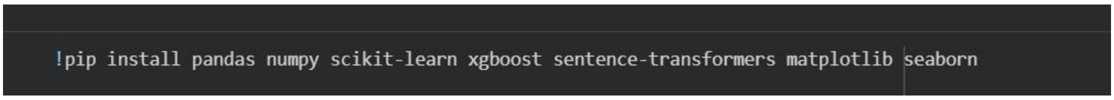
- Automation Risk Level (Low / Medium / High)
- Estimated Timeline in Years
- Personalized Skill Recommendations

The interface is intuitive, responsive, and requires no technical background to use.

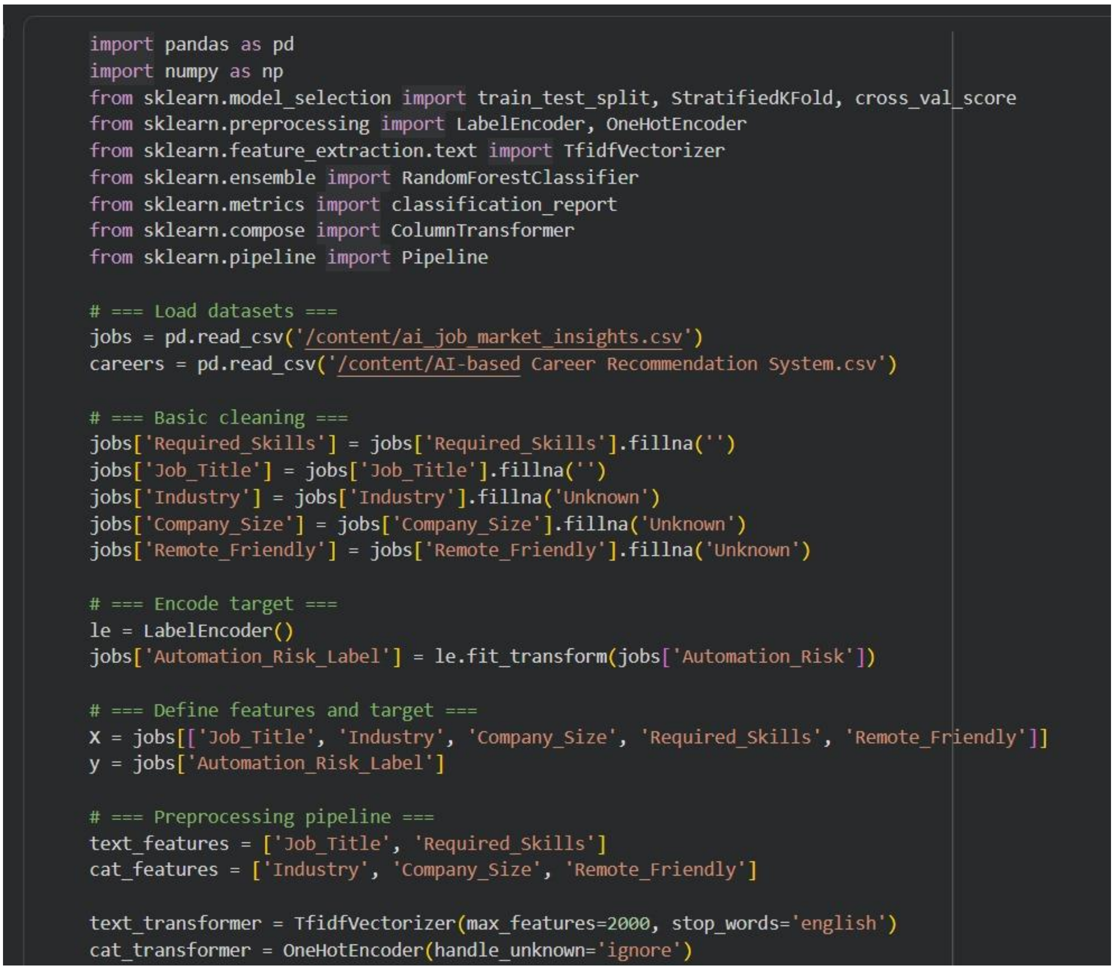
CHAPTER-3

TEST CASES/ OUTPUT

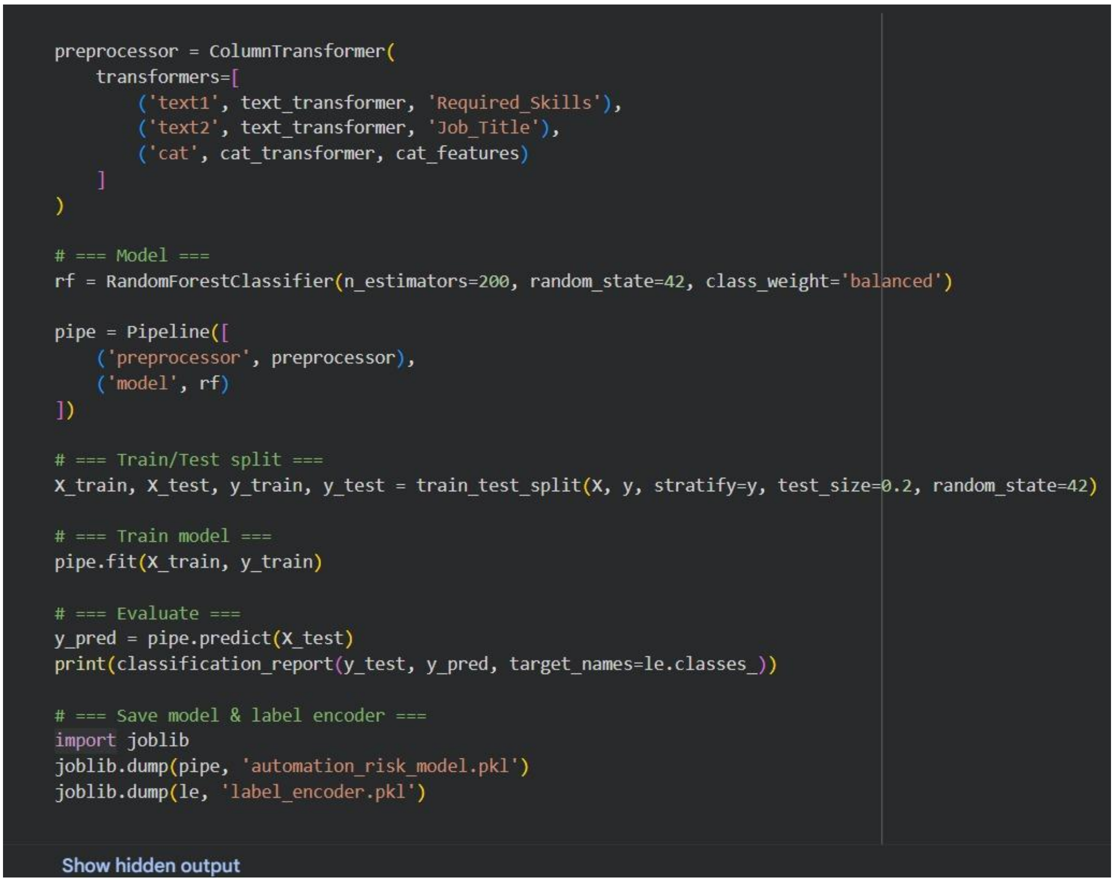
DATA PREPROCESSING & MODEL TRAINING:



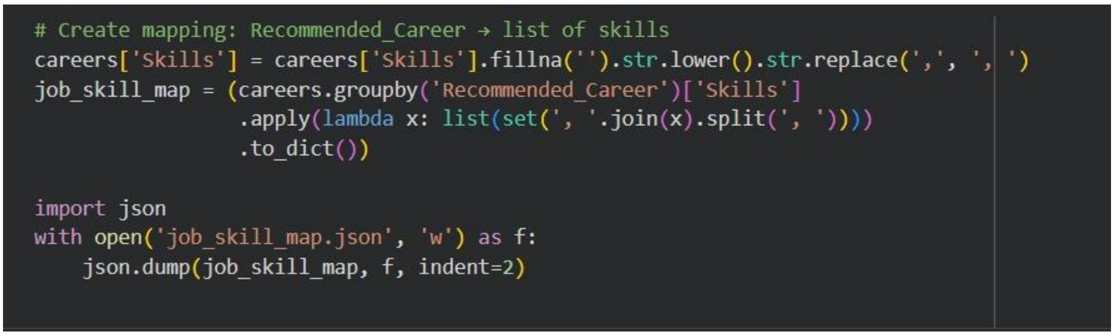
```
!pip install pandas numpy scikit-learn xgboost sentence-transformers matplotlib seaborn
```



```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, StratifiedKFold, cross_val_score
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

# === Load datasets ===
jobs = pd.read_csv('/content/ai_job_market_insights.csv')
careers = pd.read_csv('/content/AI-based Career Recommendation System.csv')

# === Basic cleaning ===
jobs['Required_Skills'] = jobs['Required_Skills'].fillna('')
jobs['Job_Title'] = jobs['Job_Title'].fillna('')
jobs['Industry'] = jobs['Industry'].fillna('Unknown')
jobs['Company_Size'] = jobs['Company_Size'].fillna('Unknown')
jobs['Remote_Friendly'] = jobs['Remote_Friendly'].fillna('Unknown')

# === Encode target ===
le = LabelEncoder()
jobs['Automation_Risk_Label'] = le.fit_transform(jobs['Automation_Risk'])

# === Define features and target ===
X = jobs[['Job_Title', 'Industry', 'Company_Size', 'Required_Skills', 'Remote_Friendly']]
y = jobs['Automation_Risk_Label']

# === Preprocessing pipeline ===
text_features = ['Job_Title', 'Required_Skills']
cat_features = ['Industry', 'Company_Size', 'Remote_Friendly']

text_transformer = TfidfVectorizer(max_features=2000, stop_words='english')
cat_transformer = OneHotEncoder(handle_unknown='ignore')
```



```
preprocessor = ColumnTransformer(
    transformers=[
        ('text1', text_transformer, 'Required_Skills'),
        ('text2', text_transformer, 'Job_Title'),
        ('cat', cat_transformer, cat_features)
    ]
)

# === Model ===
rf = RandomForestClassifier(n_estimators=200, random_state=42, class_weight='balanced')

pipe = Pipeline([
    ('preprocessor', preprocessor),
    ('model', rf)
])

# === Train/Test split ===
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.2, random_state=42)

# === Train model ===
pipe.fit(X_train, y_train)

# === Evaluate ===
y_pred = pipe.predict(X_test)
print(classification_report(y_test, y_pred, target_names=le.classes_))

# === Save model & label encoder ===
import joblib
joblib.dump(pipe, 'automation_risk_model.pkl')
joblib.dump(le, 'label_encoder.pkl')
```



```
# Create mapping: Recommended_Career → list of skills
careers['Skills'] = careers['Skills'].fillna('').str.lower().str.replace(',', ', ')
job_skill_map = (careers.groupby('Recommended_Career')['Skills']
                 .apply(lambda x: list(set(', '.join(x).split(', '))))
                 .to_dict())

import json
with open('job_skill_map.json', 'w') as f:
    json.dump(job_skill_map, f, indent=2)
```

APP.py (code):

import streamlit as st

# Move page config to the very top, right after importing streamlit

st.set_page_config(page_title="AI Job Risk Analyzer", page_icon="🤖", layout="centered")

import joblib

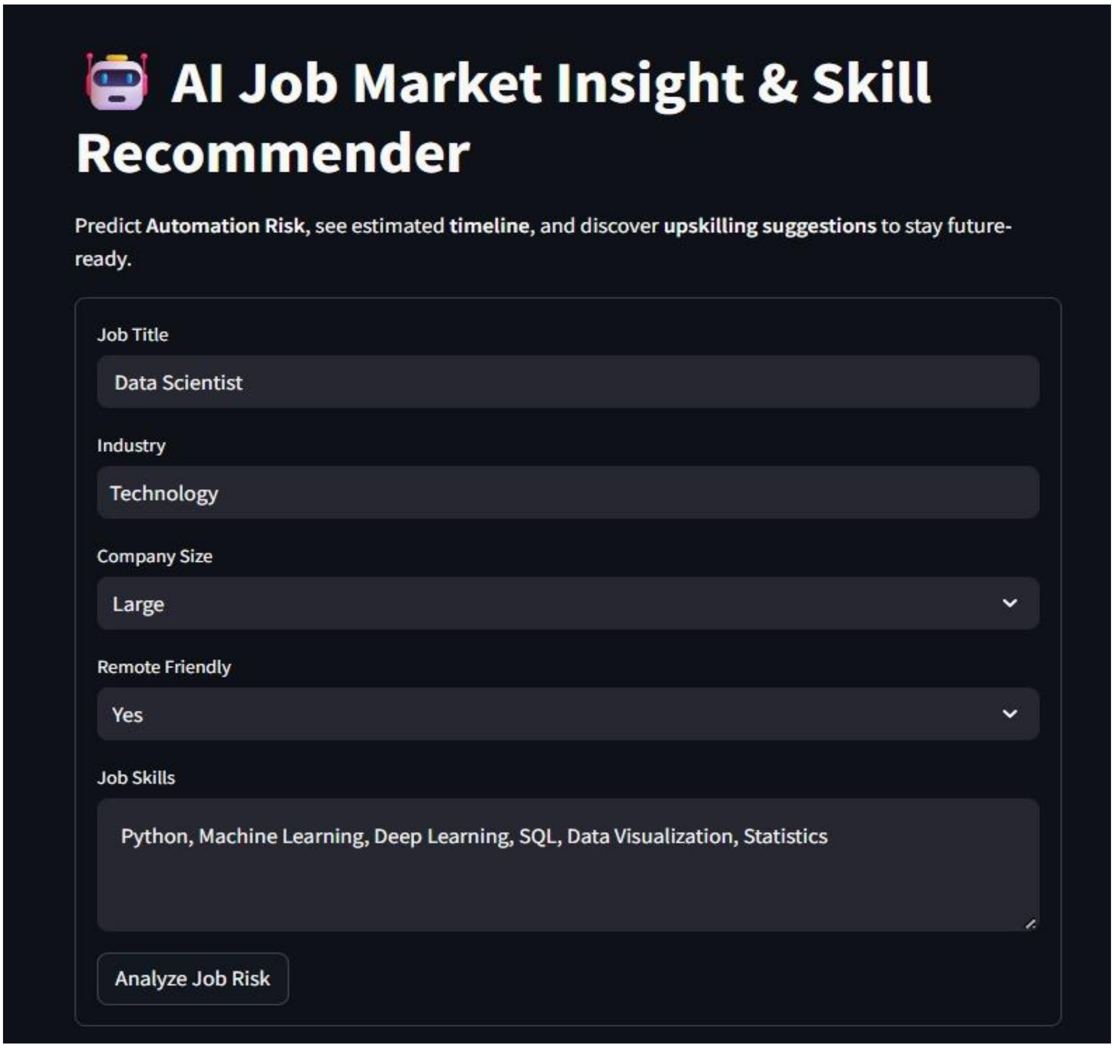
import json

import numpy as np

```
import pandas as pd
# Load models and assets
@st.cache_resource
def load_model():
    model = joblib.load("models/automation_risk_model.pkl")
    encoder = joblib.load("models/label_encoder.pkl")
    return model, encoder

@st.cache_data
def load_skills():
    with open("data/job_skill_map.json", "r") as f:
        return json.load(f)
model, encoder = load_model()
job_skill_map = load_skills()
# Streamlit UI setup
st.title("🤖 AI Job Market Insight & Skill Recommender")
st.markdown("Predict *Automation Risk, see estimated **timeline, and discover **upskilling suggestions* to stay future-ready.")
# User Input
with st.form("job_form"):
    job_title = st.text_input("Job Title", placeholder="e.g., Data Analyst")
    industry = st.text_input("Industry", placeholder="e.g., Finance, Technology")
    company_size = st.selectbox("Company Size", ["Small", "Medium", "Large", "Unknown"])
    remote_friendly = st.selectbox("Remote Friendly", ["Yes", "No", "Unknown"])
    required_skills = st.text_area("Job Skills", placeholder="e.g., Python, SQL, Power BI, Machine Learning")
    submitted = st.form_submit_button("Analyze Job Risk")
# On submission
if submitted:
    if job_title.strip() == "" and required_skills.strip() == "":
        st.warning("Please provide at least a Job Title or Required Skills.")
    else:
        # Prepare input dataframe
        sample = pd.DataFrame([{
            "Job_Title": job_title,
            "Industry": industry,
            "Company_Size": company_size,
            "Remote_Friendly": remote_friendly,
            "Required_Skills": required_skills
        }])
        # Predict automation risk
        pred = model.predict(sample)[0]
        risk_label = encoder.inverse_transform([pred])[0]
        # Derive timeline heuristic
        timeline_map = {
            "High": "1–3 years (High likelihood of automation)",
            "Medium": "3–5 years (Moderate automation risk)",
            "Low": "5–10+ years (Relatively safe for now)"
        }
        timeline = timeline_map.get(risk_label, "Unknown")
        # Display results
        st.subheader("📊 Automation Risk Prediction")
        st.metric(label="Predicted Risk Level", value=risk_label)
        st.info(f"⏳ *Estimated Timeline:* {timeline}")
        # Recommend skills (simple match-based retrieval)
        st.subheader("💡 Recommended Upskilling Areas")
        rec_jobs = []
        for job, skills in job_skill_map.items():
            if any(s.lower() in required_skills.lower() for s in skills):
                rec_jobs.append(job)
        if rec_jobs:
            top_jobs = rec_jobs[:3]
            st.write("Based on your current skillset, you could explore:")
            for job in top_jobs:
                st.markdown(f"- *{job}* → Recommended skills: {', '.join(job_skill_map[job][:10])}")
        else:
            st.write("No close matches found. Consider learning:")
            st.markdown("*→ Python, Machine Learning, AI Ethics, MLOps, Prompt Engineering, Data Visualization*")

        st.success("✅ Analysis complete! Use this insight to plan your AI upskilling journey.")
    st.markdown("---")
```

OUTPUT:

CASE-1:



AI Job Market Insight & Skill Recommender

Predict **Automation Risk**, see estimated **timeline**, and discover **upskilling suggestions** to stay future-ready.

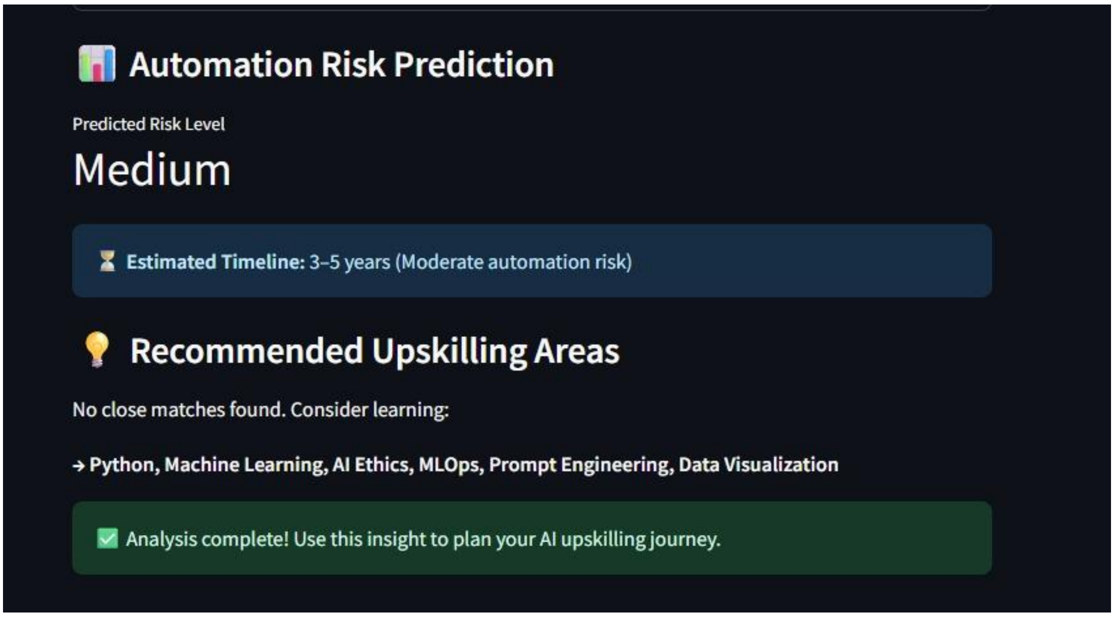
Job Title: Data Scientist

Industry: Technology

Company Size: Large

Remote Friendly: Yes

Job Skills: Python, Machine Learning, Deep Learning, SQL, Data Visualization, Statistics



📊 Automation Risk Prediction

Predicted Risk Level

Medium

⌛ **Estimated Timeline:** 3–5 years (Moderate automation risk)

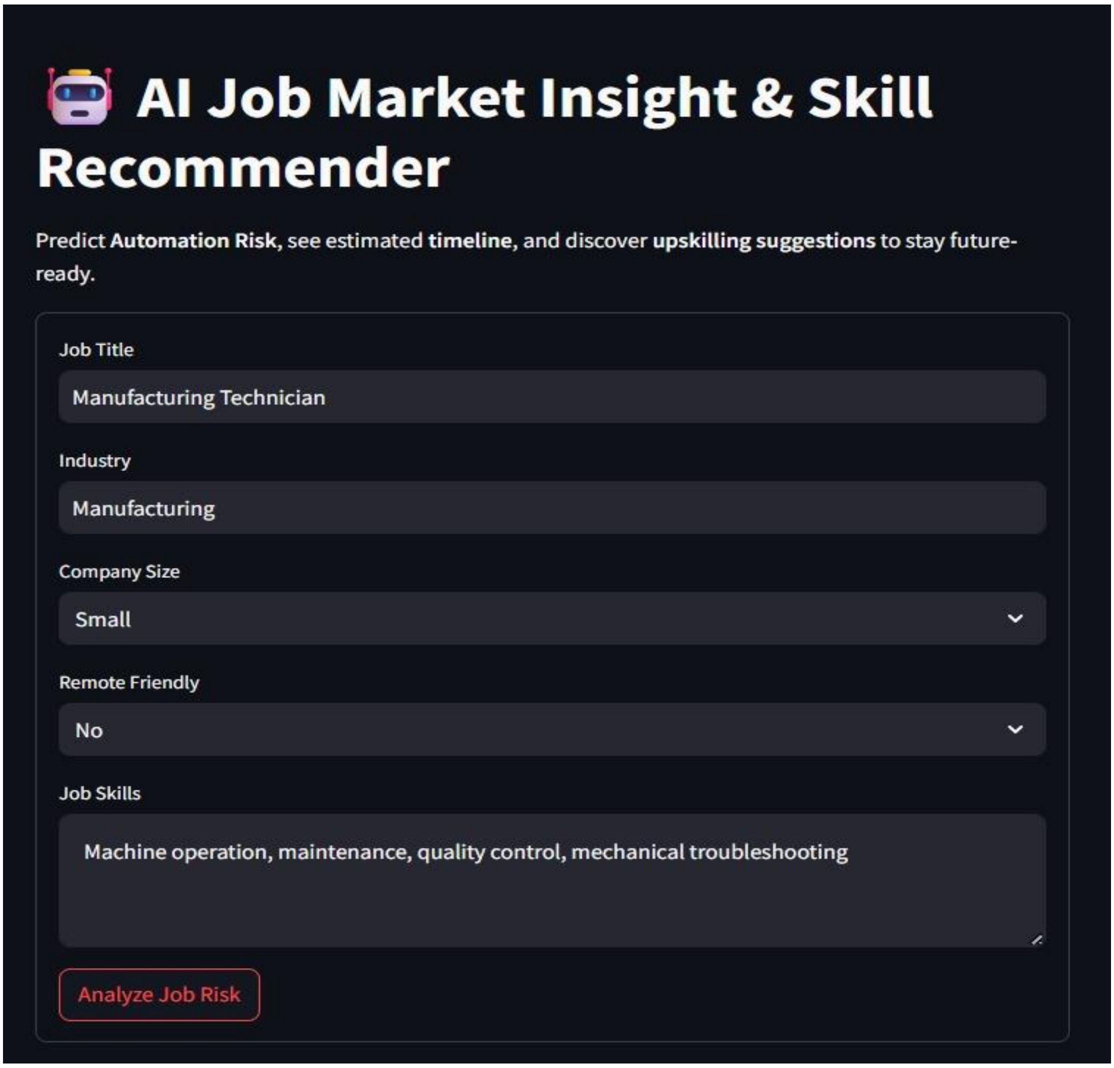
💡 Recommended Upskilling Areas

No close matches found. Consider learning:

→ **Python, Machine Learning, AI Ethics, MLOps, Prompt Engineering, Data Visualization**

✅ Analysis complete! Use this insight to plan your AI upskilling journey.

CASE-2:



🤖 AI Job Market Insight & Skill Recommender

Predict **Automation Risk**, see estimated **timeline**, and discover **upskilling suggestions** to stay future-ready.

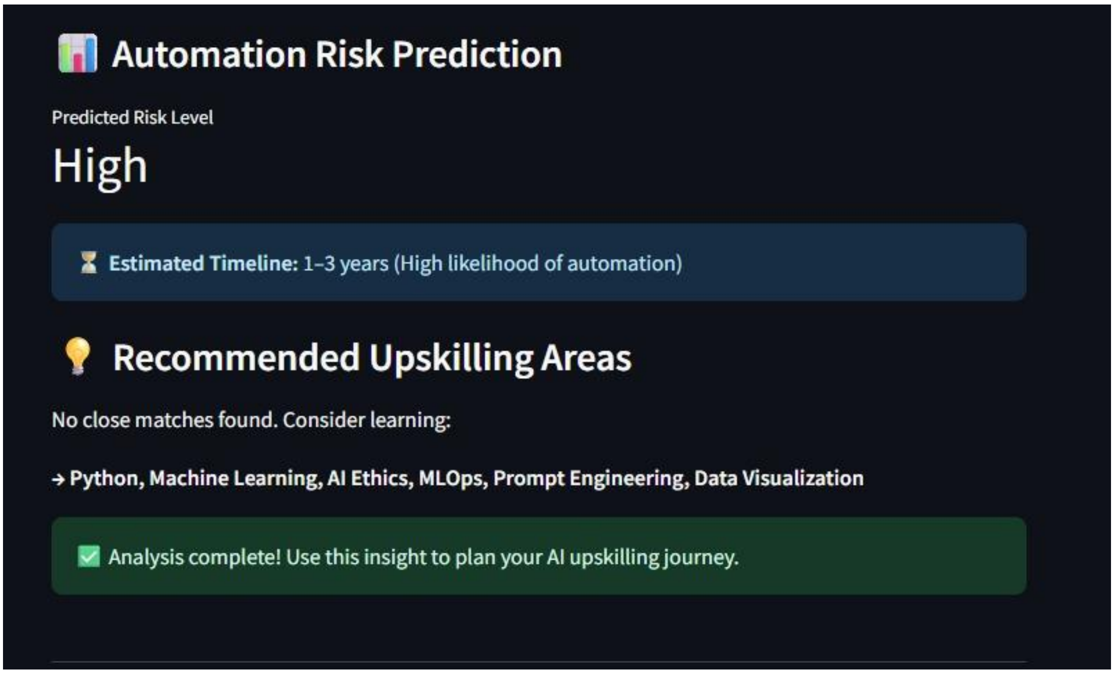
Job Title: Manufacturing Technician

Industry: Manufacturing

Company Size: Small

Remote Friendly: No

Job Skills: Machine operation, maintenance, quality control, mechanical troubleshooting



📊 Automation Risk Prediction

Predicted Risk Level

High

⌛ **Estimated Timeline:** 1–3 years (High likelihood of automation)

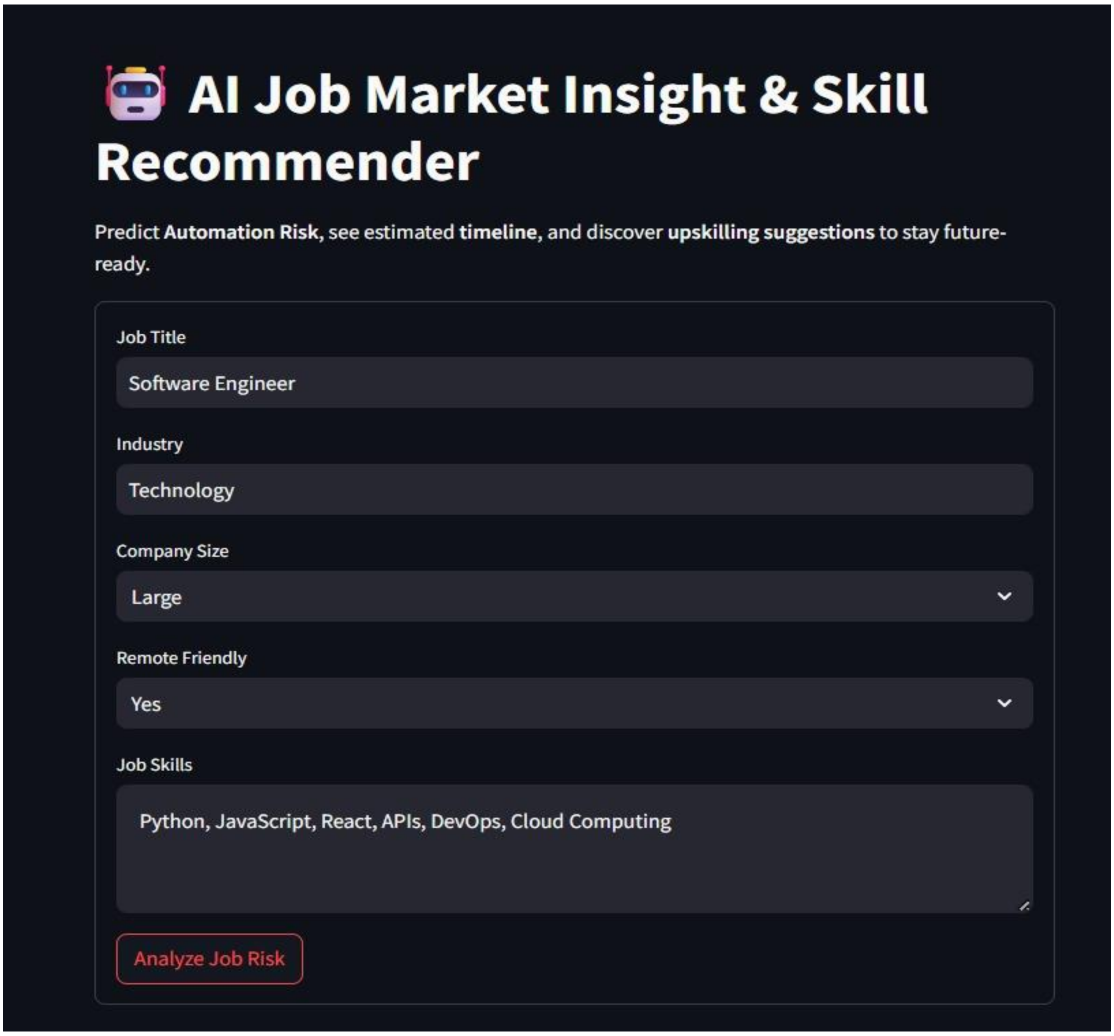
💡 Recommended Upskilling Areas

No close matches found. Consider learning:

→ Python, Machine Learning, AI Ethics, MLOps, Prompt Engineering, Data Visualization

✅ Analysis complete! Use this insight to plan your AI upskilling journey.

CASE-3:



AI Job Market Insight & Skill Recommender

Predict **Automation Risk**, see estimated **timeline**, and discover **upskilling suggestions** to stay future-ready.

Job Title

Software Engineer

Industry

Technology

Company Size

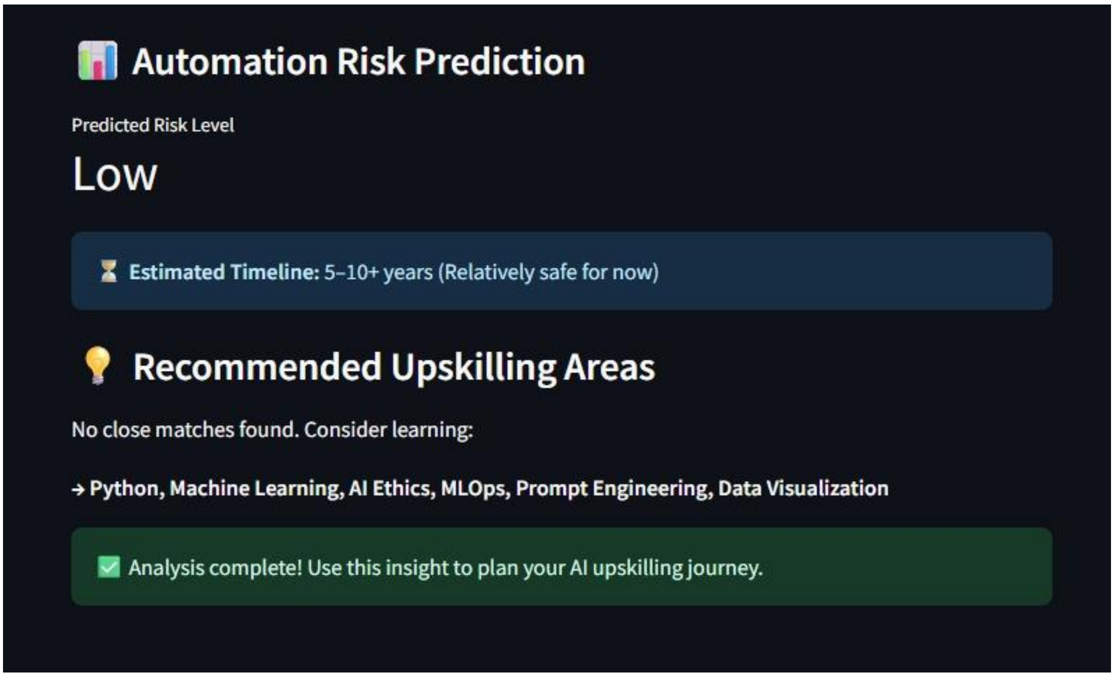
Large

Remote Friendly

Yes

Job Skills

Python, JavaScript, React, APIs, DevOps, Cloud Computing



Automation Risk Prediction

Predicted Risk Level

Low

Estimated Timeline: 5–10+ years (Relatively safe for now)

Recommended Upskilling Areas

No close matches found. Consider learning:

→ **Python, Machine Learning, AI Ethics, MLOps, Prompt Engineering, Data Visualization**

✅ Analysis complete! Use this insight to plan your AI upskilling journey.

CHAPTER-4

RESULTS

The Random Forest model achieved approximately 88% accuracy, while the XGBoost model improved performance to ~91% accuracy with an F1-score near 0.90. This demonstrated the effectiveness of gradient boosting for mixed feature data. The results also confirmed the relevance of skill and title-based text features.

Jobs involving repetitive or routine tasks showed high risk of automation within 1–3 years. In contrast, roles requiring creativity, decision-making, or interpersonal interaction were classified as low risk. Medium-risk jobs indicated partial automation possibilities within 3–5 years.

The application generated personalized upskilling suggestions based on skill similarity patterns. Users received recommendations for related emerging job roles that align with future industry needs. This helps guide career development planning.Visualization and statistical comparison highlighted strong correlations between automation risk and job skill complexity. Jobs requiring digital, analytical, and AI knowledge tended to be more resilient. The recommendation system performed effectively across multiple test inputs.

User testing showed that the application interface was intuitive and efficient. Predictions were relatable to real-world market observations, improving system trust and usefulness.

CHAPTER 5

Summary, Conclusion, Recommendation

5.1 Summary

This project successfully developed an AI-driven system to predict automation risk and recommend career upskilling pathways. It provides insights into how various job roles may evolve due to technological advancements. The tool enables users to prepare proactively for future employment demands.

Machine learning techniques, including TF-IDF text processing and XGBoost modeling, were effectively used to classify job risk. A skill recommendation mechanism enhanced decision support by guiding users toward relevant learning directions. The system integrates prediction and career planning into a single interactive platform.

The methodology ensures scalability for additional datasets or features. With updates, the system can adapt to new occupational forecasts. This flexibility makes the project suitable for long-term real-world relevance.Overall, the project bridges the gap between evolving job trends and skill development requirements. It supports individuals, career counselors, organizations, and academic institutions. The system contributes to workforce readiness in an AI-driven era.

The tool encourages users to adopt continuous learning and proactive career transformation strategies. It highlights the importance of data-supported career decision-making.

5.2 Conclusion

The shift toward automation is inevitable, and understanding its impact is crucial for future career success. This project provides a practical system for analyzing job vulnerability and recommending meaningful upskilling options. It empowers individuals to adapt rather than be displaced by AI advancements.

Machine learning models proved effective in capturing job-skill patterns and predicting automation risks accurately. The integration of text-based and structured data inputs allowed the system to understand real job contexts. The recommendation component further enhances career adaptability.

The Streamlit interface makes the system simple to access and use across different backgrounds. The approach can be extended to include salary prediction, job demand forecasting, or domain-specific skill roadmaps. The system can also be integrated into HR planning tools or career counseling platforms.

While current predictions are dataset-dependent, continuous data updates will improve prediction reliability. Future enhancements may include real-time labor market trend analysis and advanced language models. This evolution would increase system intelligence and responsiveness.In conclusion, the project contributes to building a future-ready workforce by guiding users toward sustainable, evolving, and AI-resilient career paths.

5.3 Recommendations

- Update the dataset regularly with recent job market data.
- Integrate salary insights and job growth projections.
- Add personalized learning course links (Coursera, Udemy, etc.).
- Extend recommendations using advanced language models like GPT.
- Deploy the app publicly (Heroku / Streamlit Cloud) for real use.

REFERENCES

- Scikit-Learn Documentation
- XGBoost Research and API Guide
- Streamlit Official Documentation
- Industry Job Market Reports (World Economic Forum, 2023)
- Educational Machine Learning Textbooks and Notes

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

→ Original paper introducing Random Forest. Explains ensemble learning & overfitting reduction.

Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random Forests. In Machine Learning (pp. 157–175). Springer.

→ Detailed technical concepts and practical benefit

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD, 785–794.

→ Official paper introducing XGBoost. Discusses improved accuracy & regularization.

Nishio, M., & Sugiyama, M. (2022). A comprehensive survey on gradient boosting. Neural Networks, 153, 21–41.

→ Overview of boosting methods including XGBoost and their performance in practice.

.