ml-model-input

October 31, 2024

[2]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, accuracy_score
import warnings
warnings.filterwarnings("ignore")

data = pd.read_csv("C:/Users/Manas/OneDrive/Documents/TYDS Final Year Project/
 ↪TYDS Project Data/Merged_data2.csv")
df = pd.DataFrame(data)

df['skill_count'] = df['combined_skills'].apply(len)

scaler = StandardScaler()
df['salary_normalized'] = scaler.fit_transform(df[['average_salary_value']])

X = df[['combined_skills', 'salary_normalized', 'average_experience',␣
 ↪'skill_count']]
y = df['job_title']

X['combined_skills'] = X['combined_skills'].apply(lambda x: ' '.join(x))
X = pd.concat([X.drop('combined_skills', axis=1), X['combined_skills'].str.
 ↪get_dummies(sep=' ')], axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=42)

model = DecisionTreeClassifier()
model.fit(X_train, y_train)


def get_user_input():
    user_skills = []
    while True:
        skill = input("Enter your skill (Q to end): ").strip()
        if skill.upper() == 'Q':
```

```
            break
        elif skill=='':
            print("Enter a Valid Skill!")
            print("TRY AGAIN")
            while True:
                skill = input("\nEnter your skill (Q to end): ").strip()
                if skill.upper() == 'Q':
                    break
                elif skill=='':
                   print("Enter a Valid Skill!")
                   return None
        else:
            user_skills.append(skill)
    if len(user_skills)==0:
        print("Skills cannot be Empty Enter a Skill\n")
        while True:
            skill = input("Enter your skill (Q to end): ").strip()
            if skill.upper() == 'Q':
                break
            elif skill=='':
                print("Enter a Valid Skill!")
                print("TRY AGAIN")
                while True:
                    skill = input("\nEnter your skill (Q to end): ").strip()
                    if skill.upper() == 'Q':
                        break
                    elif skill=='':
                       print("Enter a Valid Skill!")
                       return None
            else:
                user_skills.append(skill)
    if len(user_skills)==0:
        print("Skills cannot be Empty, Try Again!\n")
        return None

    try:
        user_experience = float(input("Enter your years of experience (0 if␣
↪none): "))
        if user_experience < 0:
            print("Experience cannot be negative. \nEnter a valid number!")
            try:
                user_experience = float(input("\nEnter your years of experience␣
↪(0 if none): "))
                if user_experience < 0:
                    print("\nEnter a valid number!")
                    return None
            except ValueError:
```

```
                print("\nPlease Enter a Number!")
                return None
    except ValueError:
        print("Invalid experience input. Please Enter a Number.")
        try:
            user_experience = float(input("Enter your years of experience (0 if␣
↪none): "))
            if user_experience < 0:
                print("Experience cannot be negative!")
                return None
        except ValueError:
            print("Invalid experience input!")
            return None

    try:
        user_salary = float(input("Enter your expected annual salary: "))
        if user_salary < 200000:
            print("Salary must be at least 200,000. Enter a valid salary!")
            try:
                user_salary = float(input("Enter your expected annual salary:␣
↪"))
                if user_salary < 200000:
                    print("Salary must be at least 200,000. Enter a valid␣
↪salary!")
                    return None
            except ValueError:
                print("Invalid salary input.")
                return None
    except ValueError:
        print("Invalid salary input. Please enter a number.")
        try:
            user_salary = float(input("Enter your expected annual salary: "))
            if user_salary < 200000:
                print("Salary must be at least 200,000. Enter a valid salary!")
                return None
        except ValueError:
            print("Invalid salary input.")
            return None

    skill_count = len(user_skills)

    user_salary_normalized = scaler.transform([[user_salary]])

    user_data = pd.DataFrame([[user_skills,user_salary_normalized,␣
↪user_experience, skill_count]],
                             columns=['skills','salary_normalized',␣
↪'average_experience', 'skill_count'])
```

```
    user_data['skills'] = user_data['skills'].apply(lambda x: ' '.join(x))
    user_data = pd.concat([user_data.drop('skills', axis=1),␣
 ↪user_data['skills'].str.get_dummies(sep=' ')], axis=1)

    user_data= user_data.reindex(columns=X.columns, fill_value=0)

    return user_data

user_data = get_user_input()

if user_data is not None:
    predicted_job = model.predict(user_data)
    print(f"Predicted Job Role: {predicted_job}")
```

```
Enter your skill (Q to end): python
Enter your skill (Q to end): sql
Enter your skill (Q to end): Q
Enter your years of experience (0 if none): 2
Enter your expected annual salary: 400000
Predicted Job Role: ['Data Scientist']
```