First phase Project

1. Baseball project

```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Load the dataset
baseball_data = pd.read_csv('baseball.csv')

# Exploratory Data Analysis (EDA)
# Perform EDA to understand the data, handle missing values, and identify outliers

# Data Preprocessing
# Encode categorical variables if present
# Scale numerical features if necessary

# Split the data into features (X) and target (y)
X = baseball_data.drop('W', axis=1)
y = baseball_data['W']

# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Build and evaluate multiple models
models = [
    LinearRegression(),
    # Add other regression models here
]
```

```
for model in models:
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    print(f"Model: {model.__class__.__name__}")
    print(f"Mean Squared Error: {mse}")
    print(f"R-squared: {r2}")
    print("-" * 30)

# Select the best model based on performance metrics
# Perform hyperparameter tuning if necessary
# Save the best model for production
```

2. Avocado project

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import mean_squared_error, accuracy_score

# Load the dataset
avocado_data = pd.read_csv('avocado.csv')

# Exploratory Data Analysis (EDA)
# Perform EDA to understand the data, handle missing values, and identify outliers

# Data Preprocessing
# Encode categorical variables if present
```

```
# Scale numerical features if necessary


# Classification Task

# Split the data into features (X) and target (y) for classification

X_class = avocado_data.drop('Region', axis=1)

y_class = avocado_data['Region']


# Split the data into train and test sets for classification

X_train_class, X_test_class, y_train_class, y_test_class = train_test_split(X_class, y_class,
test_size=0.2, random_state=42)


# Build and evaluate classification model

clf = DecisionTreeClassifier()

clf.fit(X_train_class, y_train_class)

y_pred_class = clf.predict(X_test_class)

accuracy = accuracy_score(y_test_class, y_pred_class)

print(f"Classification Accuracy: {accuracy}")


# Regression Task

# Split the data into features (X) and target (y) for regression

X_reg = avocado_data.drop('AveragePrice', axis=1)

y_reg = avocado_data['AveragePrice']


# Split the data into train and test sets for regression

X_train_reg, X_test_reg, y_train_reg, y_test_reg = train_test_split(X_reg, y_reg, test_size=0.2,
random_state=42)


# Build and evaluate regression model

reg = LinearRegression()

reg.fit(X_train_reg, y_train_reg)

y_pred_reg = reg.predict(X_test_reg)

mse = mean_squared_error(y_test_reg, y_pred_reg)
```

```
print(f"Regression Mean Squared Error: {mse}")

# Select the best models based on performance metrics
# Perform hyperparameter tuning if necessary
# Save the best models for production
```

3. HR Analytics project

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Load the dataset
hr_data = pd.read_csv('hr_analytics.csv')

# Exploratory Data Analysis (EDA)
# Perform EDA to understand the data, handle missing values, and identify outliers

# Data Preprocessing
# Encode categorical variables
# Scale numerical features if necessary

# Split the data into features (X) and target (y)
X = hr_data.drop('Attrition', axis=1)
y = hr_data['Attrition']

# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Build and evaluate classification model
```

```
clf = RandomForestClassifier()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
print(classification_report(y_test, y_pred))

# Analyze feature importances to understand factors contributing to attrition
feature_importances = pd.Series(clf.feature_importances_, index=X.columns)
print(feature_importances.sort_values(ascending=False))

# Perform hyperparameter tuning if necessary
# Save the best model for production
```