Task 1

Problem Description: A portfolio is the collection of financial assets, such as stocks, bonds, and other securities, held by an investor or an entity. Portfolio returns represent the overall financial performance of this collection of assets over a specific period. Design a system to predict the returns over a period.

Type of task: Regression.

Information from Data Preparation Team: We have finalized 6 features that the DNN model will utilize for learning.

Number of features in input: 6.

Task: Design a DNN model using Keras.

Reference Model:

```
model = Sequential()
model.add(Dense(5, input_shape= (16,))
model.add(Dense(5, activation='sigmoid'))
model.add(Dense(5, activation='sigmoid'))
model.add(Dense(5, activation='sigmoid'))
model.add(Dense(1))
adam = keras.optimizers.Adam(lr=0.01)
model.compile(optimizer = adam, loss = 'binary_crossentropy')
```

Instructions:

1. Navigate to 'test_model' folder and select 'design_model.py' file.
2. Design the model in 'design_model.py' file.
3. Run the 'design_model.py' file to check for compilation errors.
4. Test the designed model using automatically generated mock data. Run 'test_design_model.py'. If all the test cases pass, run 'test_improved_designed_model.py'.

Task 2

Problem Description: A seed trading company is looking for software that can distinguish the grain into one of the 7 types depending on the different features of the seed.

Type of task: Multiclass classification.

Information from Data Preparation Team: We have finalized 16 features that the DNN model will utilize for learning.

Number of features in input: 16.

Task: Design a DNN model using Keras.

Reference Model:

```
model = Sequential()
model.add(Dense(64, activation = 'relu', input_shape = (7,)))
model.add(Dropout(0.5))
model.add(Dense(64, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(3))
model.add(Activation('softmax'))
opt = RMSprop(learning_rate=0.01)
model.compile(loss='categorical_crossentropy', optimizer=opt, metrics = ['mae'])
```

Instructions:

1. Navigate to ‘test_model’ folder and select ‘design_model.py’ file.
2. Design the model in ‘design_model.py’ file.
3. Run the ‘design_model.py’ file to check for compilation errors.
4. Test the designed model using automatically generated mock data. Run ‘test_design_model.py’. If all the test cases pass, run ‘test_improved_designed_model.py’.

Task 3

Problem Description: Tracto is a leading automotive marketplace where a large collection of new and used trucks for sale is available. The selling price of these trucks depends on several factors such as year of built, number of owners etc. The sales manager at Tracto is interested in software that predicts the selling price of these trucks given several factors.

Type of task: Regression.

Information from Data Preparation Team: We have finalized 7 features that the DNN model will utilize for learning.

Number of features in input: 7.

Task: Design a DNN model using Keras.

Reference Model:

```
model = Sequential()
model.add(Dense(20, input_shape = (10,)))
model.add(Dense(20))
model.add(Dense(units = 1, activation='sigmoid'))
opti = Adam(learning_rate= 0.0001)
model.compile(optimizer = opti, loss='mean_absolute_error')
```

Instructions:

1. Navigate to ‘test_model’ folder and select ‘design_model.py’ file.
2. Design the model in ‘design_model.py’ file.
3. Run the ‘design_model.py’ file to check for compilation errors.
4. Test the designed model using automatically generated mock data. Run ‘test_design_model.py’. If all the test cases pass, run ‘test_improved_designed_model.py’.

Task 4

Problem Description: Fast Financing is a loan lending organization that offers funds for house remodeling. A person's eligibility for a loan is determined by several factors, including age, employment etc. Fast Financing is looking for software to determine an individual's eligibility for a loan.

Type of task: Binary classification.

Information from Data Preparation Team: We have finalized 14 features that the DNN model will utilize for learning.

Number of features in input: 14.

Task: Design a DNN model using Keras.

Reference Model:

```
model = Sequential()
```

```
model.add(Dense(32, input_shape = (20,)))
model.add (Dense(16, activation='relu'))
model.add (Dropout(0.2))
model.add (Dense(8, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1, activation='linear'))
opt = Adam(learning_rate=0.005)
model.compile(optimizer =opt, loss='binary_crossentropy')
```

Instructions:

1. Navigate to ‘test_model’ folder and select ‘design_model.py’ file.
2. Design the model in ‘design_model.py’ file.
3. Run the ‘design_model.py’ file to check for compilation errors.
4. Test the designed model using automatically generated mock data. Run ‘test_design_model.py’. If all the test cases pass, run ‘test_improved_designed_model.py’.

Task 5

Problem Description: Rail Authority reported derailment of a train between XYZ and ABC. Unfortunately, some people are injured badly while the others are reported to be safe. Design a system which can predict what sort of passengers are more likely to be injured.

Type of task: Binary Classification.

Information from Data Preparation Team: We have finalized 7 features that the DNN model will utilize for learning.

Number of features in input: 10.

Task: Design a CNN model using Keras.

Reference Model:

```
model = Sequential()
model.add(Conv1D(filters=20, kernel_size=4, activation='relu', padding='same', input_shape= (10, 1)))
model.add(MaxPooling1D(pool_size=2))
model.add(Flatten())
model.add(Dense(50, activation='relu'))
model.add(Dense(1, activation='softmax'))
opt = Adam(learning_rate=0.1)
model.compile(loss='binary_crossentropy', optimizer=opt, metrics=['accuracy'])
```

Instructions:

1. Navigate to ‘test_model’ folder and select ‘design_model.py’ file.
2. Design the model in ‘design_model.py’ file.
3. Run the ‘design_model.py’ file to check for compilation errors.
4. Test the designed model using automatically generated mock data. Run ‘test_design_model.py’. If all the test cases pass, run ‘test_improved_designed_model.py’.