Consider the `insurance.csv` data file. This file contains basic demographic information on customers. The goal is to build a regression model to predict insurance premium. **In Python**, answer the following:

1. (3 points) Using the pandas library, read the csv data files and create three a data-frame called `insurance`.
2. (5 points) Change `sex`, `smoker` and `region` from labels to dummy variables.
3. (5 points) Engineer the interactions/features from Chapter 4 lecture notes (the ones from the decision tree).
4. (5 points) Based on the feature selection analysis shown in Chapter 4, it seems that `age`, `bmi`, `children`, `smoker`, and `interaction_4` are the top 5 important variables. Using the top variables as input variables and `charges` as the target variable, split the data into three datasets: `train` (80%) and `test` (20%).
5. (10 points) Using `train` data-frame and the top 5 features, perform a hyper-tuning job on the random forest model. Using the GridSearchCV function and the following dictionary:

   ```
   RF_param_grid = {'n_estimators': [100, 300, 500],
                    'min_samples_split': [10, 15],
                    'min_samples_leaf': [5, 7],
                    'max_depth' : [3, 5, 7]}
   ```

   perform the hyper-parameter job with 3 folds. Identify the hyper-parameter combination that produces the minimum mean squared error. Then, use that model to predict the `charges` on the `validation` and `test` data-frames. Finally, compute the mean squared error of the predictions on the `test` data-frame.
6. (10 points) Using `train` data-frame and the top 5 features, perform a hyper-tuning job on the XGBoost model. Using the GridSearchCV function and the following dictionary:

   ```
   XGBoost_param_grid = {'n_estimators': [500],
                         'max_depth': [3, 5, 7],
                         'min_child_weight': [5, 7],
                         'learning_rate': [0.01],
                         'gamma': [0.3, 0.1],
                         'subsample': [1],
                         'colsample_bytree': [1]}
   ```

   perform the hyper-parameter job with 3 folds. Identify the hyper-parameter combination that produces the minimum mean squared error. Then, use that model to predict the `charges` on the `validation` and `test` data-frames. Finally, compute the mean squared error of the predictions on the `test` data-frame.
7. (3 points) Based on your results from parts 5, and 6, what model would you use to predict `charges`? Be specific.