DATA-101
Exam II Take-Home

Exam Instructions

1. **Show all your work** and write complete and coherent answers.
2. Show all of the steps that you used to get your final answer. If you do not show your work I can not give partial credit in the case of incorrect answers.
3. **Please strive for clarity and organization.**
4. **You are not allowed to discuss any of the exercises in Exam II take-home with others. Identical submissions will receive a 0 as a grade in Exam II take-home.**
5. **Late submission will not be accepted, regardless of the circumstances.**

Take the time to carefully read all the questions on the exam. GOOD LUCK!

1. Consider the `churn-bigml-80.csv` and `churn-bigml-20.csv` datafile. The Orange Telecom's churn dataset, which consists of cleaned customer activity data (features), along with a churn label specifying whether a customer canceled the subscription, will be used to develop predictive models. Each row represents a customer; each column contains customer's attributes. The datasets have the following attributes or features:

   - `State`: state where the customer live.
   - `Account_length`: number of months the account is active.
   - `Area_code`
   - `International_plan`: whether or not the customer has an international plan.
   - `Voice_mail_plan`: whether or not the customer has a voice mail plan.
   - `Number_vmail_messages`: number of voice mails.
   - `Total_day_minutes`
   - `Total_day_calls`
   - `Total_day_charge`
   - `Total_eve_minutes`
   - `Total_eve_calls`
   - `Total_eve_charge`
   - `Total_night_minutes`
   - `Total_night_calls`
   - `Total_night_charge`
   - `Total_intl_minutes`
   - `Total_intl_calls`
   - `Total_intl_charge`
   - `Customer_service_calls`
   - `Churn`: whether or not the customer churn.

   In Python, answer the following:

   (a) (3 points) Using the `pandas` library, read the `churn-bigml-80.csv` datafile and create a data-frame called `telecom_train`.

   (b) (3 points) Using the `numpy` library, create a variable in `telecom_train` called `Churn_numb` that takes the value of 1 when `Churn = True` and 0 when `Churn = False`. *Hint:* notice that `Churn` is stored as a logical variable, so use the following code:

   ```
   telecom_train['Churn_numb'] = telecom_train['Churn'].map({False: 0, True: 1})
   ```

   (c) (6 points) Using the `numpy` library, create two variables in `telecom_train`: one called `int_plan` that takes the value of 1 when `International_plan = Yes` and 0 when `International_plan = No`, and another one called `voice_plan` that takes the value of 1 when `Voice_mail_plan = Yes` and 0 when `Voice_mail_plan = No`.

(d) (6 points) Using the `sklearn.ensemble` library, build a `RandomForestClassifier` model, in which `Churn_numb` is the target variable, and `Account_length`, `int_plan`, `Total_intl_minutes`, `voice_plan`, `Number_vmail_messages`, `Total_day_minutes`, and `Customer_service_calls` are the input variables. Make sure you use `n_estimators` = 500 and `max_depth` = 4 in the `RandomForestClassifier` model.

(e) (6 points) Using the `sklearn.ensemble` library, build a `GradientBoostingClassifier` model, in which `Churn_numb` is the target variable, and `Account_length`, `int_plan`, `Total_intl_minutes`, `voice_plan`, `Number_vmail_messages`, `Total_day_minutes`, and `Customer_service_calls` are the input variables. Make sure you use `n_estimators` = 500 and `max_depth` = 4 in the `GradientBoostingClassifier` model.

(f) (3 points) Using the `pandas` library, read the `churn-bigml-20.csv` datafile and create a data-frame called `telecom_test`.

(g) (3 points) Using the `numpy` library, create a variable in `telecom_test` called `Churn_numb` that takes the value of 1 when `Churn = True` and 0 when `Churn = False`. *Hint:* notice that `Churn` is stored as a logical variable, so use the following code:

```
telecom_test['Churn_numb'] = telecom_test['Churn'].map({False: 0, True: 1})
```

(h) (6 points) Using the `numpy` library, create two variables in `telecom_test`: one called `int_plan` that takes the value of 1 when `International_plan = Yes` and 0 when `International_plan = No`, and another one called `voice_plan` that takes the value of 1 when `Voice_mail_plan = Yes` and 0 when `Voice_mail_plan = No`.

(i) (8 points) Using the models from part (d) and (e), predict the likelihood of churn in `telecom_test`. Then, using a cutoff value of 0.3, flag out customers who are likely to churn. That is, customers, with likelihood of churn greater than or equal than 0.3, will get a churn value of 1. On the other hand, customers, with likelihood of churn less than 0.3, will get a churn value of 0.

(j) (6 points) Compare the prediction of the two considered models with the actual values on the `telecom_test` dataset. Compute the accuracy and recall of each of the model. What is the best model? Be specific.

2. Consider the `insurance.csv` datafile. This file contains some basic customers' information and their corresponding insurance charges:

- `age`: age of primary beneficiary
- `sex`: insurance contractor gender, female, male
- `bmi`: body mass index
- `children`: number of children covered by health insurance/number of dependents
- `smoker`: smoking status
- `region`: the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.
- `charges`: individual medical costs billed by health insurance

In Python, answer the following:

(a) (3 points) Using the `pandas` library, read the csv file and create a data-frame called `insurance`.

(b) (6 points) Using the `numpy` library, create the following variables in the `insurance` data-frame:

- `sex_0_1` that takes the value of 1 when `sex = male` and 0 when `sex = female`.
- `smoker_0_1` that takes the value of 1 when `smoker = 1` and 0 when `smoker = 0`.

(c) (5 points) Using the `age`, `bmi`, `sex_0_1`, and `smoker_0_1` as input variables, and `charges` as the target variables, split the data into `train` (80%) and `test` (20%).

(d) (7 points) Using the `train` dataset, build a `RandomForestRegressor` model with `n_estimators = 500` and `max_depth = 3`. After that, use the model to predict the insurance charges in the `test` dataset. Compute the RMSE and MAE of the model.

(e) (7 points) Using the `train` dataset, build a `RandomForestRegressor` model with `n_estimators = 500` and `max_depth = 5`. After that, use the model to predict the insurance charges in the `test` dataset. Compute the RMSE and MAE of the model.

(f) (7 points) Using the `train` dataset, build a `GradientBoostingRegressor` model with `n_estimators = 500` and `max_depth = 3`. After that, use the model to predict the insurance charges in the `test` dataset. Compute the RMSE and MAE of the model.

(g) (7 points) Using the `train` dataset, build a `GradientBoostingRegressor` model with `n_estimators = 500` and `max_depth = 5`. After that, use the model to predict the insurance charges in the `test` dataset. Compute the RMSE and MAE of the model.

(h) (4 points) Using the results from part (d) to (g), what model would you select to make predictions of insurance charges? Be specific.

3. Consider the `income_age_days_annual.csv` datafile. This file contains information related to customer's age, their income, the number of days since last purchase, and annual spend (how much money a customer spend on your business). You have been asked to group the customers into groups based on their income, age, interactions and money spend. **In Python**, answer the following:

(a) (3 points) Using the `pandas` library, read the csv file and create a data-frame called `customers`.

(b) (5 points) Do customers with higher income spend more on your business? Create the scatter-plot of `income` and `annual_spend`. Comment on the plot.

(c) (6 points) Compute the 0-1 transformation (Min-Max transformation) of all the variables and store in the data-frame `customers`. Called these variables `income_0_1`, `age_0_1`, `days_since_purchase_0_1` and `annual_spend_0_1`, respectively.

(d) (8 points) Cluster the data into 3 clusters (using the 0-1 transformed variables from the `customers` data-frameand append the cluster membership to the `customers` data-frame). Make sure you use `n_init` = 20 in the `KMeans` function.

(e) (6 points) Describe each of the cluster. Is there any obvious difference between the clusters? Be specific.