APML Assignment 1

Tan Siew Ling

Titanic - Data Exploration

```
1 # List out all variables with nulls/missing values
2 titanic.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

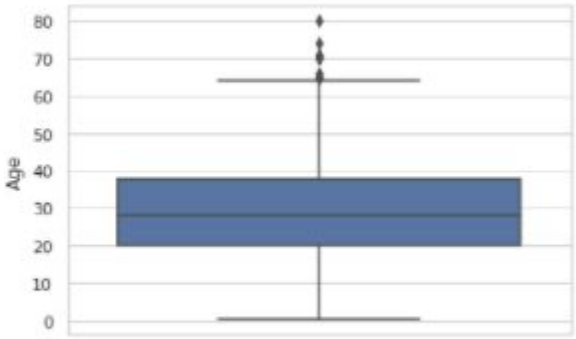
```
1 titanic.info()
```

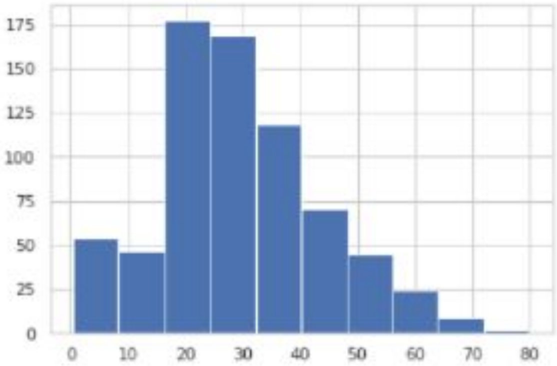
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

1. There are are 177 passengers with no age
2. There are 687 passengers without Cabin number (Likely as the Cheapest class passengers did not have a cabin)
3. There are 2 Passengers with no Embarked information

Titanic - Data Exploration

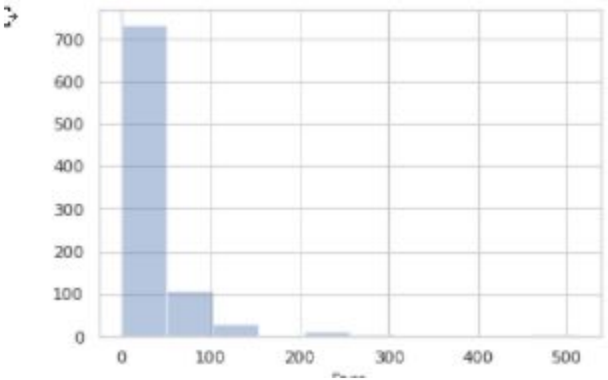
- Most passengers are aged between 20 to 40 years old





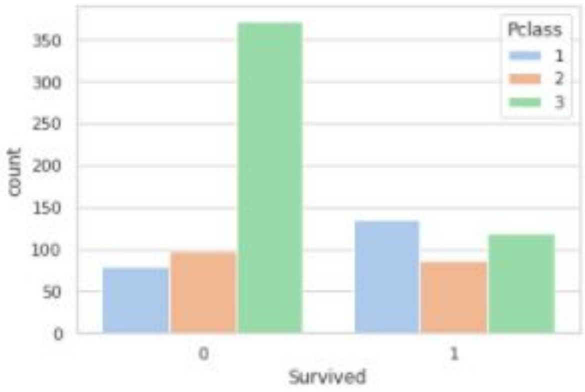
Titanic - Data Exploration

- Most passengers paid the lowest fare (PClass3).



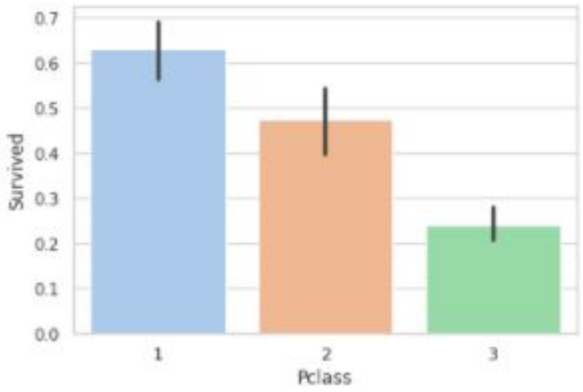
Titanic - Data Exploration

- Passengers of the lowest Class Fare (PClass3) are more likely not to survive. Possibly because they are located at the bottom of the ship. The tragedy striked during daytime so passengers were most likely not in their cabin but for the PClass3 passengers, they were not allowed up on the deck. So when the tragedy strike, they are mostly stuck at the bottom of the ship. As the ship sank, it was difficult for them to make their way up to the deck and get on the lifeboats.

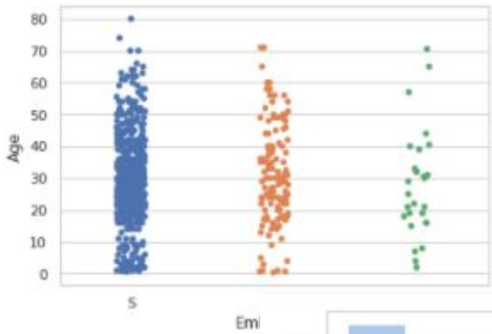


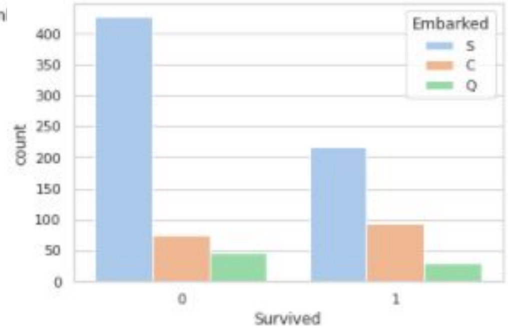
Titanic - Data Exploration

- PClass1 passengers were more likely to survive as they were mostly on the deck when the tragedy strike during the daytime. They had paid a higher fare for PClass1 and were of a higher social class and hence are allowed to go to the deck. When the tragedy striked, they had faster access to the lifeboats.



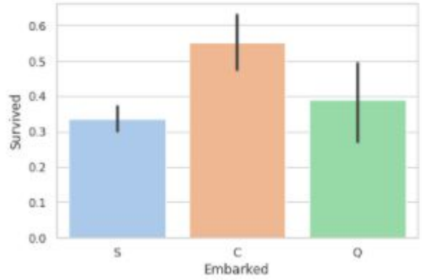
Titanic - Data Exploration





- Most passengers embarked at Southampton (Titanic sailed out point), hence passengers who survived or died were from Southampton

- Among those who survived, most of them embarked at Cherbourg, France (second port of call)



The outward **route** was to be Southampton, England – Cherbourg, France – Queenstown, Ireland – New York, USA.

Titanic - Data Exploration

```
Survived       1.000000
Sex_female     0.543351
Cabin_Yes      0.316912
Fare           0.257307
Embarked_C     0.168240
Parch          0.081629
Embarked_Q     0.003650
SibSp         -0.035322
Age           -0.069809
Embarked_S    -0.149683
Cabin_No      -0.316912
Pclass        -0.338481
Sex_male      -0.543351
Name: Survived, dtype: float64
```

Correlation of Features to Survived:

- Top 3 highest correlation feature to Survived are : **Sex_female, Cabin_Yes, Fare**

  (1) **Female** passengers were more likely to survive possibly due to social norms to let Females and the young to escape first so they are more likely to get on the lifeboats

  (2) **Passengers With Cabin** are more likely to survive as they were of higher social class so they are allowed to go to the deck. At the time of tragedy, it is daytime, so these Passengers with Cabin are likely to be on the deck, making it easier for them to reach the lifeboats first. The passengers with no cabin are not allowed to go to the deck because they were of a lower social class. So they were stuck in the bottom of the ship most of the time. So when the tragedy strike, it is difficult for them to make their way from the bottom of ship to the lifeboats

  (3) **Passengers who paid more fare** are likely to have a cabin and were of higher social class, Hence, o they are allowed to go to the deck which gave them faster access to the lifeboats when the tragedy striked. The low fare paying passengers had no cabin and cannot go to the deck because they are of a lower social class. Hence, they are at the bottom of the ship at most of the time. As the ship sinks, it was difficult for them to get to the lifeboats quickly from the bottom of the ship.

Titanic - Data Cleaning

- Handle Missing Values for features - Age, Cabin and Embarked
- Encode all the categorical features - Sex, Cabin, Embarked to tranform non-numeric feature to numeric so that they can be processed by the Learning Algorithms
- Drop columns not selected as features - PassengerId, Name as these have no Correlation to Survival of Passengers
- Data Transformation : Use StandardScaler to transform data to a relatively normal distribution on the feature columns as most Learning Algorithms assumed a normal distributed dataset

Titanic - Feature Engineering

- Imputation:
  - Impude the Missing Values for features :
    - Age : with mean age
    - Cabin: convert alphanumeric value and missing value to categorical Yes (with Cabin number) , No (No Cabin)
    - Embarked: with mode
- One-Hot Encoding:
  - Encode all the categorical features - Sex, Cabin, Embarked to tranform non-numeric feature to numeric so that they can be processed by the Learning Algorithms
- Scaling:
  - Data Transformation : Use StandardScaler to transform data to a relatively normal distribution on the feature columns as most Learning Algorithms assumed a normal distributed dataset

Titanic - Model Building

How did you select which learning algorithms to use?

- **Round 1:** Use most of the classification Machine Learning Algorithms (Logistics Regression, KNN, Naive Bayes, Decision Tree, Random Forest, SVC) to fit the data to see which gives the best Kaggle Score
- **Round 2:** Do hyperparameter tuning to get the best parameters for the best 3 Machine Learning Algorithms - Logistics Regression,Gaussian Naive Bayes and Random Forest
- **Round 3:** Retrained the models with the best parameters and select the best one - Random Forest

Titanic - Evaluation

Learning Algorithm	Test Score	Kaggle Score
Logistics Regression	77.09%	0.75358
k-Nearest Neighbor	78.21%	0.60765
Gaussian Naive Bayes	75.42%	0.73205
Decision Tree Classifier	76.54%	0.72727
SVC	80.45%	0.62200
Random Forest Classifier	78.77%	0.73923

Titanic - Hyperparameters tuning

- Did Hyperparameter tuning with cross validation of 3 folds using **GridSearchCV** to find best parameters for
  (a) Logistic Regression
  (b) Gaussian Naive Bayes

- Did Hyperparameter tuning using **RandomSearchCV** with cross validation of 3 folds (to Narrow the parameters list to tune) followed by **GridSearchCV** (with cross validation of 3 folds) to get the best parameters for
  (a) Random Forest

Titanic - Evaluation (after Hyperparameters tuning)

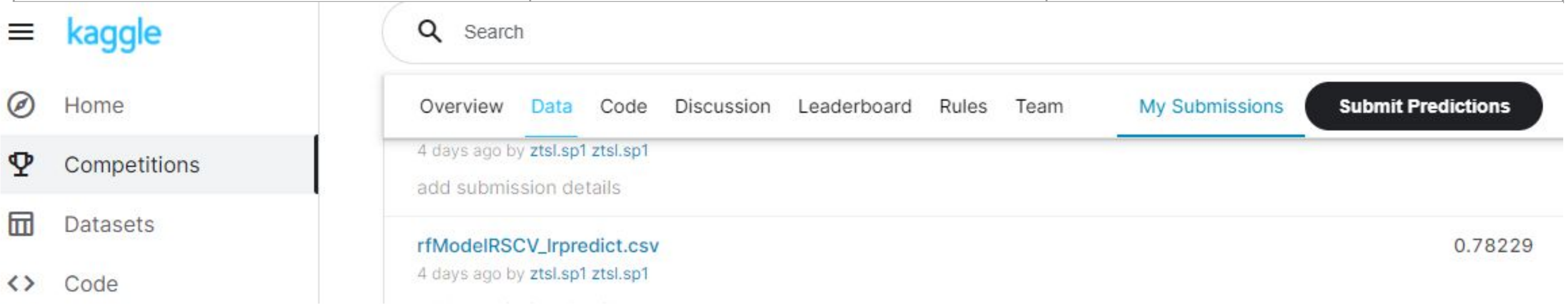
- After using **GridSearchCV** to tune for best parameters
  - Logistics Regression model has a lower Kaggle Test Score!

Learning Algorithm	Test Score	Kaggle Score
Logistics Regression	0.7765	0.66985
Gaussian Naive Bayes	0.754	0.66985

Titanic - Evaluation (after Hyperparameters tuning)

- As there are many parameters for **RandomForest**, it takes a long time for **GridSearchCV** to find the best parameters (until Colabs session crashed)
  - First use **RandomSearchCV** to narrow down the parameter list first
  - Then use **GridSearchCV** to tune the parameters list from RanndomSearchCV
- After **RandomSearchCV**:

earning Algorithm	Test Score	Kaggle Test Score
RandomForest	0.8156	0.78229



Titanic - Evaluation (after Hyperparameters tuning)

- After **GridSearchCV**:

Learning Algorithm	**Test Score**	**Kaggle Test Score**
RandomForest	0.8156	0.62200

Titanic - Conclusions

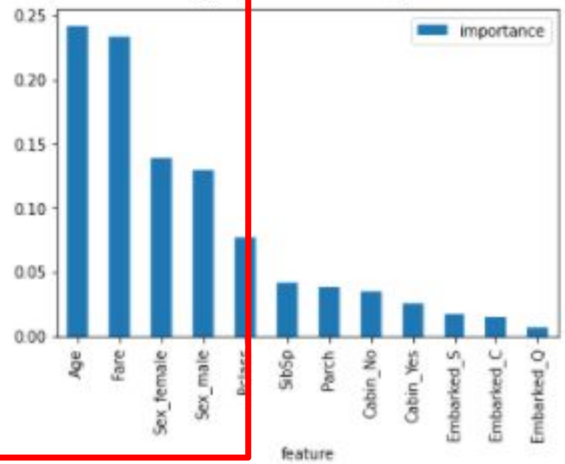
- Random Forest learning model gives the best prediction (best Kaggle score 78.229%) using the following parameters

```
rfModelRSCV=RandomForestClassifier(bootstrap=False, ccp_alpha=0.0, class_weight=None,
                                   criterion='gini', max_depth=None, max_features='auto',
                                   max_leaf_nodes=5, max_samples=None,
                                   min_impurity_decrease=0.0, min_impurity_split=None,
                                   min_samples_leaf=2, min_samples_split=10,
                                   min_weight_fraction_leaf=0.0, n_estimators=911,
                                   n_jobs=None, oob_score=False, random_state=None,
                                   verbose=0, warm_start=False)
```

Titanic - Conclusions

Which features does the model considers important?

<matplotlib.axes._subplots.AxesSubplot at 0x7f9e41eed9d0>




```
importances = pd.DataFrame({'feature':titanicdf.drop(["Survived"],axis=1).columns,'importance':np.round(rfModel.feature_importances_,3)})
importances = importances.sort_values('importance',ascending=False).set_index('feature')
importances.head(15)
importances.plot.bar()
```

House Sales - Data Exploration

- No missing values
- All values numeric except for the date
- Some columns are related to each other :
    a. sqft_living15, sqft_above related to sqft_living
    b. sqft_lot15 related to sqft_lot

House Sales - Data Exploration

Correlation of Features to Price

- Top 3 highest correlation feature to prices are : sqft_living,grade,sqft_above
    - (1) sqft_living - interior living space
    - (2) grade - 1 -13, 1-3=fall short of building constructoion,7=average, 11-13=high quality
    - (3) sqft_above - interior housing space above ground level

```
price            1.000000
sqft_living      0.702035
grade            0.667434
bathrooms        0.525138
view             0.397293
sqft_basement    0.323816
bedrooms         0.308350
lat              0.307003
waterfront       0.266369
floors           0.256794
yr_renovated     0.126434
sqft_lot         0.089661
yr_built         0.054012
condition        0.036362
long             0.021626
zipcode         -0.053203
Name: price, dtype: float64
```

House Sales - Data Cleaning

- No missing values to fill
- Data transformation: Use StandardScaler to transform data to a relatively normal distribution on the feature columns as most machine learning algorithms assume a normal distribution of data. Hence, this will allow more machine learning algorithms to be used.

House Sales - Feature Engineering

- Scaling:
    - Data Transformation : Use StandardScaler to transform data to a relatively normal distribution on the feature columns as most Learning Algorithms assumed a normal distributed dataset

House Sales - Model Building

How did you select which learning algorithms to use?

- **Round 1:** Use most of the Regression Machine Learning Algorithms to fit the data to see which gives the best Test Score

- **Round 2:** Do hyperparameter tuning to get the best parameters for the best 3 Machine Learning Algorithms

  - Random Forest Regressor, Decision Tree Regressor, Gradient Boosting Tree

- **Round 3:** Retrained the model with the best parameters and select the best one - Random Forest

House Sales - Evaluation

Learning Algorithm	Test R2 Score	Train Score
Linear Regression	0.6855275267633103	0.7019143152578121
Ridge	0.6855262867517637	0.7019143121983769
Lasso	0.6855273524872325	0.7019143150914512
Elastic Net	0.6855273213167796	0.7019143151677597
Random Forest Regressor	0.8857538901915738	0.9814522135185255
Decision Tree Regressor	0.7382126740613755	0.9992762599889737
Gradient Boosting Tree	0.867325854012937	0.8972579108154493
SVR	-0.05660755462982903	-0.05767185725860768 4
Neural Network	1.1780604800758914	-1.1208136693248374

House Sales - Evaluation (after Hyperparameters tuning)

- Using RandomSearchCV:
  - Did not improve

Learning Algorithm	Test R2 Score	Train Score
Random Forest Regressor	0.8843704812230988	0.9829409345248749
Decision Tree Regressor	0.2717010035300643	0.27172339733986695
Gradient Boosting Tree	0.8821225464868516	0.9223112918248872

House Sales - Evaluation (after Hyperparameters tuning)

- Using GridSearchCV:
  - Test R2 score drop

Learning Algorithm	Test R2 Score	Train Score
Random Forest Regressor	0.8824443790289633	0.9822848180581116

House Sales - Conclusions

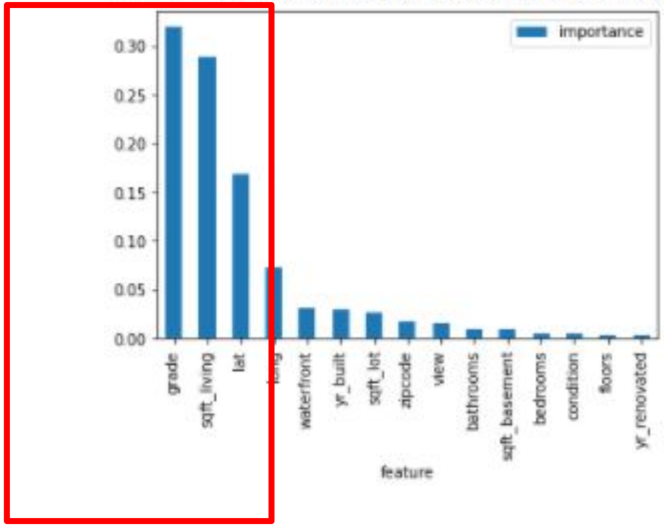
- Random Forest learning model gives the best prediction using the following parameters:

```
# fit model to train data with best params
rfRSCVModel = RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=30, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=377, n_jobs=None, oob_score=False,
                      random_state=None, verbose=0, warm_start=False)
rfRSCVModel.fit(X_train, y_train)
```

House Sales - Conclusions

Which features does the model considers important?

<matplotlib.axes._subplots.AxesSubplot at 0x7f823fdf74d0>




```
importances = pd.DataFrame({'feature':housedf.drop(["price"],axis=1).columns,'importance':np.round(rfModel.feature_importances_,3)})
importances = importances.sort_values('importance',ascending=False).set_index('feature')
importances.head(15)
importances.plot.bar()
```

References

1. https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/
2. https://towardsdatascience.com/optimizing-hyperparameters-in-random-forest-classification-ec7741f9d3f6
3. https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

Used these references to help to understand and determine which parameters to tune and how to tune